U·M·I

University Microfilms International
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313/761-4700 800/521-0600

Kinetic study on pyrolysis of C_6H_5Cl and m-$C_6H_4Cl_2$ diluted in H_2; detailed reaction kinetics and thermodynamic property estimation techniques

Ritter, Edward Robert, D.Eng.Sc.

New Jersey Institute of Technology, 1989

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

KINETIC STUDY on PYROLYSIS of C_6H_5Cl and m-$C_6H_4Cl_2$ DILUTED IN H_2; DETAILED REACTION KINETICS and THERMODYNAMIC PROPERTY ESTIMATION TECHNIQUES

by

Edward R. Ritter

Dissertation submitted to the Faculty of the Graduate
School of the New Jersey Institute of Technology in partial
fullfilment of the requirements for the degree of
Doctor of Engineering Science
1989

APPROVAL SHEET

Title of Thesis: Kinetic Study on Pyrolysis of C6H5Cl and m-C6H4Cl2 diluted in H2; Development of Thermodynamic Property Estimation Techniques for use with Detailed Reaction Kinetic Mechanisms

Name of Candidate: Edward R. Ritter
Doctor of Engineering Science, 1989.

Thesis and Abstract Approved: ______________________ ________

Professor Joseph W. Bozzelli
Full Professor
Chemical Engineering, Chemistry,
and Environmental Science Dept.

Signature of other members
of the thesis committee:

______________________ ________
B. Baltzis date

______________________ ________
A. M. Dean date

______________________ ________
D. E. Knox date

______________________ ________
G. Lewandowski date

ABSTRACT

Title of Thesis: Kinetic Study on Pyrolysis of C_6H_5Cl and m-$C_6H_4Cl_2$ Diluted in H_2; Detailed Reaction Kinetics and Thermodynamic Property Estimation Techniques

Edward R. Ritter, Doctor of Engineering Science, 1989.

Thesis Directed by: Professor Joseph W. Bozzelli

Thermal reaction studies of dilute mixtures (0.37%) of C_6H_5Cl in hydrogen and m-$C_6H_4Cl_2$ (0.4%) in hydrogen have been performed in tubular flow reactors at various surface to volume ratios, and 1 atm total pressure. Residence times range from 0.02 to 2.5 seconds,with temperatures between 1050 to 1275 K. HCl, C_6H_5Cl, C_6H_6, and carbon solids (C(s)) are observed as the major products; minor products include methane, cyclopentadiene, toluene, naphthalene and biphenyls. Pyrolysis in helium yields significantly less conversion, but more C(s) for similar residence times.

A detailed chemical mechanism is developed to describe this reaction system. Our modeling calculations incorporate Energized Complex QRRK analysis for accurate inclusion of temperature and pressure effects in radical addition reactions. This is a straightforward method to estimate rate constants and branching ratios as a function of both T and P. The detailed mechanism, based upon fundamental thermodynamic and kinetic principles, describes the overall reaction remarkably well. We also propose a plausible

kinetic scheme describing formation of minor products.

The observed reagent loss can be explained in terms of energized adduct formation, followed by unimolecular dissociation to low energy exit channels. Phenyl, chlorophenyl, and Cl radicals produced by the initial unimolecular decay of C_6H_5Cl or m-$C_6H_4Cl_2$ react with H_2 to form C_6H_6, C_6H_5Cl, and HCl + H respectively. Hydrogen atom addition to C_6H_5Cl and m-$C_6H_4Cl_2$ via ipso attack, with subsequent loss of Cl from the energized complex, is required to explain the faster reaction in H_2 than He.

QRRK Rate constant analysis is presented for the addition of H atom, phenyl and chlorophenyl radicals to C_6H_5Cl and m-$C_6H_4Cl_2$, temperature and pressure ranges of 300 to 1900 K and 10^{-3} to 1.0 atm, respectively. Calculations were performed for both nitrogen and hydrogen dilution gases. Computer simulations show that initial chlorophenyl and phenyl addition reactions cannot, by themselves, explain the rapid formation of solid carbon in these systems.

Computer simulation and QRRK analyses on these reaction systems require accurate thermodynamic property data for all radicals and stable molecules considered. A computer code called THERM was developed to estimate these properties using the group additivity method of Benson. Accurate heat capacity estimates for radicals and molecules are obtained only to 1000 K using Benson's method. Two methods for the accurate extrapolation of this low temperature heat capacity data (T < 1000 K) to higher temperatures were developed in the present study.

The first method uses an exponential function to interpolate and extrapolate heat capacity data, with a typical maximum error of 4%. The second fitting method is based upon the harmonic oscillator model for heat capacity of an ideal gas described by statistical mechanics. A 5 parameter harmonic oscillator equation (HOE) was developed and used to estimate heat capacity data from 300 - 5000 K, with maximum error less than 0.1% for simple polyatomic molecules having no hindered internal rotations. Less than 4% maximum error is obtained for molecules which have hindered internal rotations. The harmonic oscillator equation parameters were determined by a least square regression of heat capacity data using the method of Marquardt. These parameters were used to calculate various molecular properties such as geometric and arithmetic mean frequencies, vibrational partition functions, or the sum and density of vibrational quantum states. Estimates are found to compare favorably to those calculated from actual vibrational frequency data.

A polynomial fitting method was developed to create thermodynamic property functions in the NASA polynomial format which is required by the CHEMKIN reaction simulation computer code. Two polynomials were generated (one for low temperature and one for high temperature) and a point of tangency was determined using the Newton-Raphson method.

This polynomial fitting method differs from those developed at NASA and Sandia National Laboratory, where the point of tangency is fixed at 1000 K.

The heat capacity fitting/extrapolation method, polynomial fitting, and property change for chemical reaction procedures were coupled with the group additivity method of Benson to form the THERM computer package. This code may be used to estimate the required ideal gas thermodynamic properties for radicals and molecules considered in a detailed reaction mechanism used to model kinetic data.

VITA

Name: Edward R. Ritter

Degree and date to be conferred: D. Eng. Sc., May 25, 1989

Secondary education: Essex Catholic H. S., Newark, NJ. 1977

Collegiate institutions:	Date	Degree	Date of degree
New Jersey Institute of Technology, Newark, NJ.	1977-1983	B.S.Ch.E.	1983
New Jersey Institute of Technology, Newark, NJ.	1984-1986	M.S.Ch.E.	1986
New Jersey Institute of Technology, Newark, NJ.	1986-1989	D.Eng.Sc.	1989

Major: Chemical Engineering.

Publications and Presentations:

"Kinetic Study on Thermal Decomposition of Chlorobenzene Diluted in Hydrogen", manuscript submitted to J. Phys. Chem, with J. W. Bozzelli and A. M. Dean.

"Therm: Thermodynamic Property Estimation Computer Code for Use with Detailed Reaction Kinetic Mechanisms", 1989 Spring Technical Meeting, Central States Section, The Combustion Institue, Dearborn, Michigan, with J.W. Bozzelli.

"Experimental Observation and Mechanism for Ring Growth from Chloro and Dichlorobenzene Pyrolysis in H2 (1000 - 1300 K)", 1989 Spring Technical Meeting, Central States Section, The Combustion Institue, Dearborn, Michigan, with J.W. Bozzelli. and A.M. Dean.

"Thermodynamic Property Estimation for Chlorinated Dioxins, and likely Candidate Radical Precursors to Chlorinated Dioxin Formation", 1989 Spring Technical Meeting, Central States Section, The Combustion Institue, Dearborn, Michigan, with S.N. Patel and J.W. Bozzelli.

"Therm: Thermodynamic Property Estimation Computer Code for Use with Detailed Reaction Kinetic Mechanisms", 1988 Fall Technical Meeting, Eastern States Section, The Combustion Institue, Clearwater Beach, Fl., with J.W. Bozzelli.

"Pyrolysis Reactions for Chloro and Dichlorobenzene in H_2" 1988 Fall Technical Meeting, Eastern States Section, The Combustion Institue, Clearwater Beach, Fl., with J.W. Bozzelli and M. Hung.

"Pyrolysis Reactions for Chloro and Dichlorobenzene in H_2 and He Bath Gas", 1988 Spring Technical Meeting, Central States Section, The Combustion Institue, Indianapolis, In., with J.W. Bozzelli and M. Hung.

"Mechanism of CH_4, Cyclopentadiene, Biphenyl, Chlorobiphenyl, Toluene, and Naphthalene Formation from Pyrolysis of Chlorobenzene in H_2", 20th Fall Technical Meeting, Eastern States Section, The Combustion Institute, NBS, Gaithersburg, Md. 1987, with J.W. Bozzelli and A.M. Dean.

"Thermal Decomposition of Chlorobenzene in H_2 and He Atmospheres", 1987 Spring Technical Meeting, Central States Section, The Combustion Institute, Argonne National Laboratory, Chicago, with J. W. Bozzelli.

Positions held: Research Associate, Dept. of Chemical Engineering, Chemistry, and Environmental Science, 1987-1989.

Research Assistant Technician, Dept. of Chemical Engineering, Chemistry, and Environmental Science, 1986-1987.

Engineer-In-Training, E.I. Dupont, Photo Products Department, Parlin, NJ. 1981-1982.

CONTENTS

List of Tables iii
List of Figures v

Section I : Kinetic Study on Thermal Decomposition of Chlorobenzene Diluted in Hydrogen 1
1. Introduction 2
2. Experimental 6
3. Experimental Results 8
4. Kinetic Modeling Results and Discussion 13
5. References for Section I 32
6. Appendix I - A 34
7. Appendix I - B 50
8. Appendix I - C 69

Section II : Kinetic Study on Thermal Decomposition of m-Dichlorobenzene Diluted in Hydrogen..... 71
1. Experimental Results 72
2. Kinetic Modeling Results and Discussion 76
3. References for Section II 86
4. Appendix II - A 87
5. Appendix II - B 114
6. Appendix II - C 135

Section III : THERM: Thermodynamic Property Estimation Computer Code for Use With Detailed Reaction Kinetic Mechanisms.............. 140
1. Introduction 142
2. Other Computer Implementations of Benson's Method 143
3. Source of Group Contribution Data 145
4. Some Examples of The Use of This Program 145
5. References for Section III 151
6. Appendix III - A 153
7. Appendix III - B 156
8. Appendix III - C 160

Section IV : Fitting / Extrapolation Methods for The Heat Capacity of an Ideal Gas 218
1. Extrapolation/Interpolation of Cp(T) for an Ideal Gas Using an Exponential Function 219
2. Generation of Thermodynamic Property Data in Polynomial Form 223
3. Heat Capacity Data From Statistical Mechanics 234
4. Harmonic Oscillator Equation for Cp(T) 236
5. Applications of Harmonic Oscillator Parameters to Calculations Requiring Vibrational Frequencies ... 242
   Vibrational Partition Function 242
   Sum and Density of Vibrational Quantum States 244
   Estimation of Geometric and Arithmetic Mean Frequencies 245
6. References for Section IV 250
7. Appendix IV - A 252
8. Appendix IV - B 261

9. Appendix IV - C 342
10. Appendix IV - D 358
11. Appendix IV - E 378
12. Appendix IV - F 386

Appendix V : Source Code Listing 406

LIST OF TABLES

Section I:

I.i Chemact Input: C6H5Cl + H 35
I.ii Chemact Input: C6H6 + H 36
I.iii Chemact Output: C6H6 + H 37
I.iv Chemact Input: CH3 + C6H6 38
I.v Chemact Output: CH3 + C6H6 39
I.vi Chemact Input: CH3 + C6H5Cl 40
I.vii Chemact Output: CH3 + C6H5Cl 41
I.viii Chemact Input: C6H5 + C6H5Cl (ipso) 42
I.ix Chemact Output: C6H5 + C6H5Cl (ipso) 43
I.x Chemact Input: C6H5 + C6H5Cl (non-ipso) 44
I.xi Chemact Output: C6H5 + C6H5Cl (non-ipso) 45
I.xii Chemact Input: C6H5 + C6H6 46
I.xiii Chemact Output: C6H5 + C6H6 47
I.xiv Detailed Reaction Mechanism 48

Section II:

II.i Geometric Mean Vibrational Frequencies and L-J parameters for Radical Adducts 88
II.ii Chemact Input/Output: H + m-C6H4Cl2 92
II.iii Chemact Input/Output: C6H4Cl + C6H5Cl (ipso) 93
II.iv Chemact Input/Output: C6H4Cl+C6H5Cl (non-ipso) 94
II.v Chemact Input/Output: C6H5+C6H4Cl2 (ipso) ... 95
II.vi Chemact Input/Output: C6H5+C6H4Cl2 (non-ipso) 96
II.vii Chemact Input/Output: C6H4Cl+C6H4Cl2 (ipso) . 97
II.viii Chemact Input/Output: C6H5+C6H5C6H5 (meta) .. 98
II.ix Chemact Input/Output: C6H5+C6H5C6H5 (ortho) . 99
II.x Chemact Input/Output: C6H5+PhPhCl (ortho) ... 100
II.xi Chemact Input/Output: C6H5+PhPhCl (meta) 101
II.xii Chemact Input/Output: C6H5+PhPhCl (meta/ipso) 102
II.xiii Chemact Input/Output: C6H5+PhClPhCl(meta/ipso)103
II.xiv Chemact Input/Output: C6H5+PhClPhCl (meta) .. 104
II.xv Chemact Input/Output: C6H4Cl+C6H5C6H5 (meta) 105
II.xvi Chemact Input/Output: C6H4Cl+C6H5C6H5 (ortho) 106
II.xvii Chemact Input/Output: C6H4Cl+PhPhCl (ortho) . 107
II.xviii Chemact Input/Output: C6H4Cl+PhPhCl(meta/ipso)108
II.xix Chemact Input/Output: C6H4Cl+PhClPhCl (meta) 109
II.xx Chemact Input/Output: o-Terphenyl radical ... 110
II.xxi Chemact Input/Output: o-Ph3Cl2 radical 111
II.xxii Chemact Input/Output: o-Ph3Cl radical 112
II.xxiii Mechanism listing 113

Section III:

III.i Comparison of Hf(298 K) estimates w/ Lit 154
III.ii Examples: # of Rotors and Symmetry # 155

Section IV:

IV.i Polynomial Representation of Thermodynamic Property Data 253
IV.ii NASA Format Explained 254
IV.iii Estimated and Literature Mean Vibrational Frequencies for Selected Molecules 255

IV.iv Input Data for SF6 dissociation (DISSOC) 258
IV.v Input Data for C6H5 dissociation (DISSOC) ... 259
IV.vi Input Data for C6H6 dissociation (DISSOC) ... 260

LIST OF FIGURES

Section I:

1.1 Experimental Apparatus 51
1.2 Reactor Temperature Profile 52
1.3 1st Order Decomposition for C6H5Cl 53
1.4 Effect of S/V Ratio on 1st Order Rate Const.. 54
1.5 Product Distribution vs. Time (1223 K) 55
1.6 Product Distribution vs. T(K) (1 second) .. 56
1.7 Effect of Bath Gas on Decomposition 57
1.8 Energy Diagram: H + C6H5Cl 58
1.9 Energy Diagram: H + C6H6 59
1.10 Isomerization Sequence for Figure 1.9 60
1.11 Apparent Rate Constant for H + C6H6 61
1.12 Energy Diagram: CH3 + C6H6
CH3 + C6H5Cl 62
1.13 Energy Diagram: C6H5 + C6H5Cl (ipso attack) . 63
1.14 Energy Diagram: C6H5 + C6H5Cl (non-ipso) 64
1.15 Model Predictions and Experimental Observation at 1126 K 65
1.16 Model Predictions and experimental Observation at 1177 K 66
1.17 Model Predictions for Decomposition vs. T(K) 67
1.18 Model Prediction assuming equilibrium Between C6H6 and PAH's 68

Section II:

2.1 1st Order decomposition of m-C6H4Cl2 115
2.2 Effect of Reactor Diameter on Decomposition . 116
2.3 Effect of S/V on 1st Order Rate Const 117
2.4 Product Distribution vs. Time (1123 K) 118
2.5 Product Distribution vs. T(K) (1 second) 119
2.6 GC/Mass Chromatogram of Soot Extract 120
2.7 Energy Diagram: H + M-C6H4Cl2 121
2.8 Energy Diagram: C6H4Cl + C6H5Cl (ipso) 122
2.9 Fall Off Behavior for system in Figure 2.8 .. 123
2.10 Energy Diagram: C6H4Cl + C6H5Cl (non-ipso) .. 124
2.11 Non-Arrhenius Behavior for Systems in Figures 2.8 and 2.10 125
2.12 Energy Diagram: C6H4Cl + C6H4Cl2 126
2.13 Energy Diagram: C6H4Cl + PhClPhCl (meta/ipso) 127
2.14 Fall Off curve for System in Figure 2.13 128
2.15 Energy Diagram: C6H4Cl + PhPhCl (ortho) 129
2.16 Energy Diagram: o-Terphenyl radical 130
2.17 Fall Off Curve for Systme in Figure 2.16 131
2.18 Model Prediction and Experimental Observation at 1096 K 132
2.19 Model Prediction and Experimental Observation at 1200 K 133
2.20 Model Prediction for Decomposition vs. T(K) . 134

Section III:

3.1 Example Document File Format (*.DOC) 157
3.2 Example NASA Polynomial Format (*.DAT) 158
3.3 Example Thermo List File (*.LST) 158
3.4 Comparison of Cp(T) predictions w/ Literature 159

Section IV:

4.1 Cyclopentadiene: Cp(T) w/ Exponential model . 262
4.2a Ethylene Oxide: Cp(T) w/ Exponential Model .. 263
4.2b " " " " HOE model 264
4.2c " " Cp(T) residuals 265
4.2d " " H(T)-H(298) w/ Exponential F'n . 266
4.2e " " " " HOE model 267
4.2f " " H(T)-H(298) residuals 268
4.2g " " S(T)-S(298) w/ Exponential F'n . 269
4.2h " " " " HOE model 270
4.2i " " S(T)-S(298) residuals 271
4.3a " " Exponential F'n Extrapolation .. 272
4.3b " " " " residuals 273
4.3c " " " " Projection 274
4.4 NASA Cp(T) Polynomials: Cycloheptatriene 275
4.5a Biphenyl Radical: Cp(T) 276
4.5b " " Cp(T) residuals 277
4.5c " " H(T)-H(298) 278
4.5d " " H(T)-H(298) residuals 279
4.5e " " S(T)-S(298) 280
4.5f " " S(T)-S(298) 281
4.6a Phenoxy Radical: Cp(T) 282
4.6b " " Cp(T) residuals 283
4.6c " " H(T)-H(298) 284
4.6d " " H(T)-H(298) residuals 285
4.6e " " S(T)-S(298) 286
4.6f " " S(T)-S(298) 287
4.7a Furan Cp(T) 288
4.7b " Cp(T) residuals 289
4.8 Cyclopentadiene: Cp(T) w/ geometric mean vibrational frequency and harmonic oscillator model 290
4.9 Cyclopentadiene: Cp(T) w/ arithmetic mean vibrational frequency and harmonic oscillator model 291
4.10 Cyclopentadiene: Cp(T) w/ HOE 291
4.11a Benzene: Cp(T) fit w/ HOE 292
4.11b " Cp(T) residuals 293
4.12a C2Cl6: HOE Cp(T) 295
4.12b " " Cp(T) residuals 296
4.12c " " H(T)-H(298) 297
4.12d " " H(T)-H(298) residuals 298
4.12e " " S(T)-S(298) 299
4.12f " " S(T)-S(298) 300
4.13a C2H2 HOE Cp(T) 301
4.13b " " Cp(T) residuals 302
4.13c " " H(T)-H(298) 303
4.13d " " H(T)-H(298) residuals 304

4.13e C_2H_2 " S(T)-S(298) 305
4.13f " " S(T)-S(298) 306
4.14a NH_3 HOE Cp(T) 307
4.14b " " Cp(T) residuals 308
4.14c " " H(T)-H(298) 309
4.14d " " H(T)-H(298) residuals 310
4.14e " " S(T)-S(298) 311
4.14f " " S(T)-S(298) 312
4.15a C_2F_6: HOE Cp(T) 313
4.15b " " Cp(T) residuals 314
4.15c " " H(T)-H(298) 315
4.15d " " H(T)-H(298) residuals 316
4.15e " " S(T)-S(298) 317
4.15f " " S(T)-S(298) 318
4.16a Phenyl HOE Cp(T) 319
4.16b " " Cp(T) residuals 320
4.16c " " H(T)-H(298) 321
4.16d " " H(T)-H(298) residuals 322
4.16e " " S(T)-S(298) 323
4.16f " " S(T)-S(298) 324
4.17a Phenol HOE Cp(T) 325
4.17b " " Cp(T) residuals 326
4.17c " " H(T)-H(298) 327
4.17d " " H(T)-H(298) residuals 328
4.17e " " S(T)-S(298) 329
4.17f " " S(T)-S(298) 330
4.18a Biphenyl HOE Cp(T) 331
4.18b " " Cp(T) residuals 332
4.18c " " H(T)-H(298) 333
4.18d " " H(T)-H(298) residuals 334
4.18e " " S(T)-S(298) 335
4.18f " " S(T)-S(298) 336
4.19 Vibrational Partition Function vs. T(K) for Cyclohexa-1,3-diene 337
4.20 Vibrational Partition Function vs. T(K) for C_2F_6 338
4.21 Sum of Vibrational Quantum States for Cyclohexa-1,3-diene 339
4.22 Sum of Vibrational Quantum States for Biphenyl 339
4.23 Fall Off Curve for SF_6 dissociation 340
4.24 Fall Off Curve for C_6H_5 dissociation 341
4.25 Fall Off Curve for C_6H_6 dissociation 341

SECTION I

KINETIC STUDY ON THERMAL DECOMPOSITION OF C_6H_5Cl DILUTED IN H_2

Introduction

The chemistry and high temperature reactions of chlorinated organic species are currently receiving significant attention due to the consistent observation of known or suspected toxic / carcinogenic chlorocarbons or chloro-oxy carbon species in effluent from waste and resource recovery incinerators[1,2]

Chuang and Bozzelli [3] indicate that thermal reactions of chlorinated hydrocarbons in a hydrogen atmosphere may lead to complete conversion of alkyl chlorocarbon species to HCl and the hydrocarbon products acetylene and ethylene. Louw et.al. [4] in separate studies have further indicated that thermal reactions of chlorinated biphenyls with hydrogen may lead to detoxification of these species, producing non-chlorinated aromatics and HCl. Louw and his coworkers[5-8] have performed thermal studies on a number of chloro-aromatic reactions in the presence of hydrogen, postulating mechanistic pathways and some rate constants. One plausible low activation energy reaction path is the addition of a radical species, e.g. H atoms, to the aromatic ring (adduct formation) with subsequent rapid reaction of the adduct to new products, which serve to propagate a chain reaction e.g.

$$H + C_6H_5Cl \text{ <----> } [ClCyC_6H_6]^* \text{ ---> } C_6H_6 + Cl$$

These reaction paths may be considered displacements, but do not occur on a single bond system where the outer shell of the central atom is expanded. Instead they follow a scheme suggested in by Benson and Shaw[9] in a review of hydrogen atom reactions with propylene, benzene, toluene and xylene where the data showed rapid reactions that could be explained within a factor of ten by one rate constant. This rate constant, furthermore, remains nearly constant over order of magnitude variations in pressure and temperature. Benson and Shaw explained this observation in part by indicating that reactions such as:

H + Toluene ---> CH_3 + Benzene.
H + Propylene ---> CH_3 + Ethylene

were rapid and pointed out that the reactions could not be explained by abstraction reactions due to the high rate constants observed at lower temperatures. They further indicated, however, that stabilization should occur for any adducts that might be formed. An alternate explanation, which we advocate, when a low energy exit channel such as displacement of methyl by H on an unsaturated hydrocarbon is present, is that unimolecular dissociation of the energized adduct is rapid and often occurs prior to any stabilization. The H atoms in the reviewed data[9] were initially generated from unimolecular decomposition of the resonantly stabilized parent and the subsequent displacement reaction is exothermic. The ipso attack of H on the toluene carbon bonding the methyl group forms a C - H bond, which is stronger than this C_b - CH_3 bond and allows facile cleavage of the C - C single bond. Louw[5] extended this reaction concept to the carbon chlorine bond system.

W. Tsang[10] has recently reported a rate constant on the displacement reaction of H with chlorobenzene from shock tube studies between 1000 and 1100 K of $2.2x10^{13}exp(-9000$ cal/mole/RT). He further indicates that displacement is the only significant pathway at temperatures included in his study.

Frenklach et. al[11] monitored soot formation from chlorobenzene reactions in shock tubes and showed that soot was produced at lower temperatures than occurred with benzene in pyrolysis type reactions strongly suggesting that the Cl on benzene accelerated the reaction to soot. Rao and Skinner[12], on the other hand, studied reactions of chlorobenzene and benzene at low concentrations in shock tubes (1600 to 1900 K) by following atomic hydrogen concentration profiles and observed little difference in overall reaction for the two aromatic species. They concluded that little difference existed in the reaction rates of the two species.

One added complexity in reaction studies of Cl-aromatics and a possible reason for chlorobenzene's acceleration of carbon (soot) formation is the role of Cl, a product in our reaction system. Here the rapid chain carrying or catalytic transfer process can take place:

Cl + H_2 ---> HCl + H

H + ClBz ---> Cl + Bz

H_2 + ClBz = HCl + Bz   overall reaction

where abstraction by Cl has lower energies of activation than abstraction by carbon containing radicals.

Earlier studies by Cullis and Manton[13] on pyrolysis of chlorobenzene between 1043 and 1123 K indicated that aromatic ring rupture did not occur significantly in experiments below 1073 K where principal products were chlorobiphenyls, H_2 and HCl. Loss of aromatic ring structure and production of vinyl chloride was, however, reported as the the major pathway between 1103 and 1123 K while major products of chlorobiphenyls and HCl were again observed above 1123 K. Low concentrations of methane were also observed along with solid carbon or soot formation. Cullis and Manton also added hydrogen to the reaction system and observed significant acceleration of the reaction.

We have elected to study and model the reactions of dilute mixtures of chlorobenzene in hydrogen and helium, at temperatures up to 1275 K in an attempt to understand and characterize the detailed reaction chemistry in these more straightforward 3 element (hydrogen / carbon / chlorine) systems.

Our modeling calculations incorporate the Energized Complex Quantum RRK analysis of Dean[14] for accurate inclusion of temperature and pressure effects on radical addition reactions at our temperatures. Dean demonstrated the importance of chemical activation in high temperature pyrolysis and oxidation systems and further studies[15,16] have shown that this approach provides a reliable and straightforward method to estimate rate constants and branching ratios as a function of both T and P.

Experimental

Reaction of chlorobenzene with hydrogen was carried out using the apparatus described in Fig. 1.1. Reactions were studied at 1 atmosphere total pressure with temperatures ranging from 1073-1283 K. Separate experiments were performed using quartz tubular reactors of 4mm, 10.5mm, and 16mm id, residence times 0.02 to 2.5 sec., and with flow conditions determined as in Ref. 17. Varying reactor surface to volume ratio (S/V) allows one to decouple apparent wall and bulk phase decomposition rates using a method based upon the work of Kaufman[18].

Chlorobenzene was reagent grade supplied by Aldrich Co. Hydrogen and helium gases were commercial grade supplied by Airdyne Inc., and were filtered of O_2, H_2O, and hydrocarbon impurities before entering the reactor system.

Reactor tubes were housed within a three zone electric tube furnace. Temperature profiles were obtained using a type K thermocouple probe moved axially within the 0.5 meter length reactor. Thermocouple error caused by furnace wall radiation was minimized by providing a shield for sensor junctions; measurements were made with He (1-2 liter/min) flowing through the reactor. Tight temperature control resulted in temperature profiles isothermal to within ±5 K over 85-90% of the furnace length throughout the temperature range studied. Steep temperature gradients of 500 K in 5 cm occurred at the inlet and outlet of the reactor (see Fig. 1.2). Uncertainty in absolute temperature measurements is estimated to be ±1% (± 8-12 K) but relative temperatures are believed to be known to within ±5 K. Hydrogen (or helium) was bubbled through 273 K liquid chlorobenzene at rates ranging from 50-4000 ml/min (stp). Sufficient contact time was allowed to insure saturation of the gas (initial C6H5Cl mole fraction of 0.0037). The flow of the saturated vapor/gas mixture was manipulated to control the residence time within the reactor by adjusting H2/He flow.

Reactants and stable products were quantified using on-line gas chromatography: Varian 3700, 25m x 0.32mm i.d. fused silica capillary column, methyl silicone stationary phase, automated cryofocussing (concentrator), flame ionization detection, and integration by an SP 4270 integrator / plotter. Initial C6H5Cl peak area was obtained using a reactor bypass (473 K) and was verified for each flow rate. All transfer lines were heated to 473 K to prevent condensation and suppress adsorption.

Quantitative analysis of HCl was performed for 1 sec residence time. Reactor effluent was diverted through a bubbler train containing 0.01M NaOH before being exhausted to a fume hood. The HCl produced was then calculated based upon titration of bubbler solution with 0.01M HCl to its phenolphthalein end point.

Products were identified by GC/Mass Spectrometry (Kratos MS25, 50m x 0.22mm i.d. methyl silicon stationary phase column) on batch samples of reactor gas drawn from the reactor exit into evacuated 25ml stainless steel sample cylinders for later analysis.

EXPERIMENTAL RESULTS: C_6H_5Cl

The reaction of chlorobenzene with hydrogen in excess was observed to follow pseudo-first order kinetics throughout the temperature range studied (1025 - 1283 K). Figure 1.3 illustrates the first order behavior obtained for several temperatures studied in a 10.5mm reactor. Major products were benzene, HCl, and solid carbon.

Decomposition was most rapid within the 4mm i.d. and slowest within the 16mm id reactor. This trend is expected since observed chlorobenzene loss may be the result of two parallel reaction paths, both contributing under our conditions. Clearly the relative importance of the wall reaction is greater when the surface to volume (S/V) ratio or relative extent of the wall surface is greater. The importance of wall reactions was found to drop as S/V was reduced, with 16mm and 10.5mm reactors giving approximately the same results, but 4mm reactors consistently showed higher conversion. Figure 1.4 demonstrates the effect of reactor diameter upon observed first order rate constants for chlorobenzene loss at several temperatures. Data plotted here have been smoothed by fitting first order rate data to an Arrhenius expression. Wall effects are characterized by the slope of these lines with the intercept estimating the bulk reaction rate. It is apparent that wall effects become very important at temperatures above about 1200 K for 4mm id reactors. These results are in agreement with the trends observed in previous studies at this laboratory.[3,17]

Decoupling wall and bulk reaction rate constants can be achieved by using a linear relationship which assumes ideal plug flow behavior and is based upon the work of Kaufman,[18] where observed experimental rate constants increase linearly as S/V is increased. Results of decoupling with the plug flow model are global rate constants for reaction in the homogeneous phase (kb) and the heterogeneous wall reaction (kw). The Arrhenius expressions for kb and kw are as follows:

$$\ln(kb\ \{1/\text{sec}\}) = 23.15 \pm 0.96 - 54 \pm 2.26\{\text{Kcal/mol}\}\ /RT(K)$$

$$\ln(kw\ \{\text{cm/sec}\}) = 25.7 \pm 0.78 - 62 \pm 1.9\{\text{Kcal/mol}\}\ /RT(K)$$

with error bars determined from the scatter in the plot of k_{exp} vs. S/V (2/R).

Figure 1.5 presents the normalized concentration of major products (HCl not included) as a function of residence time, 1.6cm i.d. reactor at 1123 K. The benzene yield rises consistently with time. It reaches an apparent steady state value near 65% of the initial C_6H_5Cl after approximately 0.4 seconds. At this point, the C_6H_5Cl concentration has fallen to 10% its initial value. The toluene concentration (not shown) rises from zero to an apparent steady state value of 0.5% the initial C_6H_5Cl concentration.

Nearly 30% the C_6H_5Cl initially present is lost to solid carbon (soot) formation. Solid carbon was estimated by difference after taking into account the relative response factors for the stable products which were measured by the GC (FID), and was obvious upon examination of the reactor tube and gas sampling filter. The propensity of chlorobenzene to form soot is well documented[11]. The C-Cl bond dissociation energy (97 kcal/mol, 298 K)is lower than that of C-H (113.5 kcal/mol), leading to more rapid phenyl radical formation from C_6H_5Cl than C_6H_6. In our system, C(s) was found in two distinct forms. Graphitic-like deposits covered reactor walls in the high temperature region; and particulate carbon (entrained by gas flow) deposited in post reactor zones and sampling line filters. No carbon deposits were observed at the boundary of the flow tube high temperature (reaction) and low temperature (quenching) zones. This suggests that either particulate growth is occurring during quenching (radical loss and particle agglomeration) or particle settling occurs due to reduced flow velocities outside the reactor.

Minor products ($\leq$0.5%) observed in samples of the reactor gas include cyclopentadiene, toluene, dichlorobenzene, ethylene, methane, naphthalene, 1,1-biphenyl, and chlorobiphenyl. Observations of products are in good agreement with previous studies.[5-8]

The distribution of major products formed at 1 second average residence time are presented as functions of temperature in Figure 1.6 for the 16mm reactor. Benzene yield increases to approximately 65% the C_6H_5Cl initially present at 1223 K. Nearly 35% of C_6H_5Cl is converted to carbon. HCl yield per mole C_6H_5Cl initially present is also shown. HCl yield rises steadily as temperature is increased. The maximum yield is about 82% and occurs at 1103 K. We do not feel that this incomplete Cl mass balance is a result of systematic error in the HCl measurement. The titration and its calibration are accurate to better than 1.0%. The reactor effluent was bubbled through a two stage sampling train with standard NaOH solution in each stage, until an end point was reached in the 1st stage. The two solutions were then combined and back titrated with standard HCl to phenolphthalein endpoint. Complete mass balance for Cl is obtained, in addition, for almost all other lower temperature pyrolysis and oxidation studies on chlorohydrocarbons completed and ongoing in this lab.

Mass spectral analyses of solids (soot) from the reactor exit show trace levels of high molecular weight chlorinated compounds (chloro and dichlorobiphenyl) are present. It is felt that these accumulated in the carbon black by adsorption after a period of time. In addition, other compounds in the MW range 228-380 were present. These are believed to be several PAH and chlorinated PAH (such as triphenylene and chlorotriphenylene) which result from phenyl polymerization reactions and are not more highly chlorinated PCB's.

A limited number of experiments were performed to examine C_6H_5Cl decomposition in a helium atmosphere under conditions similar to the hydrodechlorination studies. A comparison of these results is made in Figure 1.7, although we did not collect sufficient data to perform a rigorous kinetic analysis on pyrolysis in He. As shown in Fig. 1.7, the presence of molecular hydrogen was found to greatly accelerate C_6H_5Cl decomposition. This was accompanied by accelerated formation of benzene and, in contrast, polyphenyl and soot formation was significantly reduced, although polymerization is not eliminated. If present, hydrogen plays a key role in C_6H_5Cl decomposition.

KINETIC MODELLING RESULTS AND DISCUSSION

Energized Complex/QRRK Theory as presented by Dean[14] was used for modeling of reactions where a chemically activated adduct is formed. Calculations are based upon a geometric mean frequency <v>. The algorithm presented by Dean was modified, however. Gamma function evaluations have replaced factorials for evaluation of the energy dependent terms. Barriers (Ea/hv) are not rounded to the nearest integer before summations with respect to energy, so these summations are not forced to stop at integer values of energy (E_i/hv). This is believed to give better estimates, especially at low temperatures where small changes in barrier height can have a large effect.

The mechanistic considerations in this paper make use of chemical activation principles for reactions involving atom or radical addition to an unsaturated bond and for atom + radical or radical + radical combination reactions. Consider the addition of two species, A and B to form a chemically activated complex $AB^{\#}$, which has energy equal to the sum of the thermal energies of A and B plus the bond energy from adduct formation. This chemically activated adduct $AB^{\#}$ can now undergo the following reactions:

i Reaction to new products.
ii Stabilizing collisions to form AB^o.
iii Back reaction to initial reagents A + B.

$$\begin{tikzcd} A + B \arrow[r, leftrightarrow, "k_1", "k_{-1}"'] & AB^{\#} \arrow[r, "k_2"] \arrow[d, "k_{stab}"] & C + D \\ & AB^{o} & \end{tikzcd}$$

The Quantum RRK chemical activation treatment provides apparent reaction rate constants k_{rxn} to products C + D and k_{stab} to AB^o. The unimolecular reaction of AB^o to C + D is incorporated into the mechanism separately with consideration for falloff[14]. (The channel $AB^o \longrightarrow A + B$ is included as the reverse of k_{stab}).

This approach can be compared to a more conventional thermal equilibrium analysis of the above system, which would consider the reaction as:

$$A + B \underset{k_{-1}}{\overset{k_1}{\longleftrightarrow}} AB \xrightarrow{k_2} C + D$$

For conditions of steady state:

$$(AB)_{ss} = k_1 \, [A][B]/(k_{-1} + k_2)$$

$$\frac{d([C]+[D])}{dt} = \frac{k_2 k_1 \, [A][B]}{(k_{-1} + k_2)} = k_2 K_{eq}[A][B]$$

in the limit that $k_{-1} \gg k_2$ and where K_{eq} is the equilibrium constant.

$$K_{eq} = [AB]/([A][B])$$

We will show that there can be substantial differences in the computed rate of formation of C + D, depending upon whether one uses a thermal or chemical activation approach. The two approaches only become equivalent at the high pressure limit, where all of the $AB^{\#}$ is stabilized prior to reaction. An important finding is that the pressure required for this complete stabilization varies dramatically with temperature[14]. As a result, we find that chemical activation is important in this chlorobenzene / hydrogen system at temperatures 1100 to 1300 K and pressures of 1 atmosphere. Furthermore, the chemical activation approach becomes significantly more important as the temperature is increased.

We feel that the concept of energized complex formation and consideration of its stabilization, forward and reverse reaction channels is a more accurate way to both view and treat the reaction process. We include microreversibility for all reactions in our mechanism so equilibrium can be established for any reaction should the necessary and sufficient conditions exist.

The initial reaction occurring is chlorobenzene dissociation to phenyl radical + Cl. These initial products will both react rapidly with H2 (if present) to generate benzene and HCl in addition to reactive H atoms. Hydrogen atom addition to the Cl-Carbon in chlorobenzene (ipso position) forms an energized complex and results in rapid dissociation of the adduct to benzene plus Cl, the low energy exit channel (see Figure 1.8). This addition reaction is believed to be a major pathway for the hydrodechlorination of chlorobenzene to form benzene and HCl under our conditions (H2 in excess). Virtually every energized complex formed by this addition decomposes to C_6H_6 + Cl, because the exit barrier is so small. This is an example where the rate constant has virtually no pressure dependence even though the process is not an elementary reaction. Parameters used in these calculations appear in Table I.i. The apparent rate constant for reaction to benzene + Cl is therefore, just the rate constant for formation of the adduct, or $1.5x10^{13}$exp(-7,500 Kcal/mole/RT).

Atomic H will add to benzene to form cyclohexadienyl radicals (CHD.)* as shown in Figures 1.9 and 1.10. Also shown are some of the subsequent reactions of this radical. The CHD. complex is initially "hot" since, in addition to the thermal energy, it contains energy resulting from formation of the new chemical bond. Prior to stabilization it may unimolecularly isomerize through a bicyclo intermediate to a cyclopentadiene methyl radical (CpdMe.), which can abstract from H_2, undergo a hydrogen shift to a resonantly stabilized methyl cyclopentadienyl radical (MeCpd.) or beta scission to either fulvene + H or back to CHD radical. The energetics for these isomerizations are significantly less than ring opening reaction to linear C6H7 as shown in Fig. 1.9. Calculations for ring opening to linear C6H7 show that the energy requirement limits this path to less than 0.01% of collisions which form cyclohexadienyl radical complex at our conditions. This channel has been omitted from our reaction mechanism since it was not observed to play a significant role. However at temperatures near 2000 K, this channel may become more significant.

Methane probably arises from further reactions of methyl cyclopentadiene and MeCpd· radical. Both unimolecular dissociation of MeCpd and H addition to MeCpd·, yield methyl + cyclopentadiene radicals, with the methyl reacting with H_2 to form the CH_4. A similar scheme has been proposed by Louw, et. al. to account for the methane observed in their study[7].

A QRRK analysis of the chemically activated system, using generic or literature values for high pressure rate constants[12] and species thermodynamic properties for the reaction sequence, yields the apparent rate contants shown in Fig. 1.11. Rate parameters used in these calculations are summarized in Table I.ii, with results in Table I.iii. Dissociation of the complex back to reactants (no reaction) clearly dominates and is consistent with work of Skinner.[12,19] The isomerization channel, however, may result in MeCpd. for slightly more than 0.1% of collisions which create the CHD· complex at 1200 K and 1 atm pressure. Analysis predicts that direct production of MeCpd. will be favored at lower pressures. At higher pressures (above 760 torr) the stabilized CHD· species are predominant and require additional collisions to make MeCpd. or revert to initial reagents. Although our calculations predict that isomerizations will occur to form stabilized MeCypd., the reverse reaction is also rapid due to a low energy barrier so that only a small steady state concentration will exist in equilibrium with the CHD radical and C_6H_6 + H.

We can use this reaction to compare the results of the chemical activation QRRK calculations with those where the products are calculated from thermal activation. Consider:

$$H + C_6H_6 \text{ <---> } CyC_6H_7 \qquad K_{eq}$$

$$CyC_6H_7 \text{ ----> } LinC_6H_7 \qquad k_d$$

where the $linC_6H_7$ would unimolecularly decompose to acetylene and butadienyl radical. The butadienyl radical would further decompose to acetylene plus vinyl.

For the thermal calculation we assume the equilibrium

$$H + C_6H_6 \text{ <---> } CyC_6H_7$$

or $k_{-1} >> k_2$ from the earlier discussion.

$$K_{eq} = [CyC_6H_7]/[H][C_6H_6]$$

then $[CyC_6H_7] = K_{eq}[H][C_6H_6]$ and with the rate constant for production of $LinC_6H_7$ from above:

$$d[LinC_6H_7]/dt = k_d\ K_{eq}\ [H]\ [C_6H_7] = k_{therm}\ [H]\ [C_6H_7]$$

with k_{therm} the thermal bimolecular rate constant.

We calculate k_d at 1200 K and 1 atm with the energetics and rate constants illustrated in Figure 1.9 and Table I.ii as 4.5 x 10^3, 19 percent of the high pressure limit 2.4 x 10^4. Other dissociation channels, forward to products and back to reactants, were also considered in the calculation of k_d. Combining this with an equilibrium constant of 2.18 x 10^4 yields a value for k_{therm} of 9.8 x 10^7 cc/mole-sec for production of linC_6H_7. The QRRK analysis at these same conditions yields a chemically activated rate constant of 6.4 x 10^8, 6.5 times faster than the thermal calculation result. This demonstrates the importance of these reactions involving energized adducts at high temperatures, prior to their collisional stabilization, even at pressures as high as 1 atm.

Toluene may be formed by methyl addition at the ipso position in chlorobenzene followed by rapid loss of Cl from the complex (figure 1.12b). Methyl addition to benzene would not be as important in this system since the C-H bond in MeCHD. is nearly 10 Kcal/mole stronger than $C_{benzene}$-CH3 (Fig 1.12a). Methyl cyclohexadienyl radical will dissociate back to CH_3 + C_6H_6 reactants at our conditions. We estimate the reaction of methyl radical with chlorobenzene forms toluene 10 times faster than addition of methyl to benzene. The high pressure A factors and barriers are listed in Tables I.iv and I.vi, with results tabulated in Tables I.v and I.vii. While it is clear from Figures 1.12a and 1.12b that enthalpy favors reaction of CH_3 + C_6H_5Cl to produce Cl + toluene, close analysis shows that entropy counters this effect to some extent.

The reaction of methyl radicals with benzene to form toluene plus H atoms through the $MeCyC_6H_6$ (Methylcyclohexadienyl radical) intermediate provides another comparison between chemical and thermal activation.

CH_3 + C_6H_6 <---> $MeCyC_6H_6$ ---> H + $MeCyC_6H_5$ (toluene)

The equliibrium constant for methyl plus benzene to $MeCyC_6H_6$ is calculated as 2.5 at 1200 K. The rate constant for dissociation of $MeCyC_6H_6$ to toluene + H is calculated as 3.2×10^7, 7 % of the high pressure limit $4.9x10^8$. Initial input data for the unimolecular dissociation of $MeCyC_6H_6$ are illustrated in Figure 1.12 and Table I.v. These data yield a value of k_{therm} to H + toluene of $7.9x10^7$ cc/mole-sec. The chemical activation apparent rate constant from the energized complex calculation at 1200 K and 1 Atm pressure is 1.04×10^9 cc/mole-sec, thirteen times the result obtained assuming equilibrium with the thermally stabilized adduct.

Phenyl radical addition to either chlorobenzene or benzene will yield some biphenyl product, again going through a substituted cyclcohexadienyl intermediate, shown in Figures 1.13 and 1.14. Calculations show that more than 90% of the collisions result in biphenyl + Cl for phenyl addition to the chlorobenzene ipso position at 1 atm pressure with H2 bath gas. This appears to be a primary route to biphenyl production in this system. Phenyl addition at

C-H positions in benzene or chlorobenzene results in nearly equal amounts of stabilized intermediate and the biphenyl or chlorobiphenyl + H products. While dissociation of the complex is again important, the QRRK calculations show 20% of the phenyl + benzene collisions dissociate to biphenyl + H products at 1200 ^{o}K and 1 atm. A similar result was obtained for phenyl + chlorobenzene (for all positions except ipso addition). Rate parameters used in the calculations in addition to results are summarized in Tables I.viii - I.xiii. It is noteworthy that our calculation for the reaction of phenyl radical + benzene to give biphenyl + H, 7.7×10^{10}, is in excellent agreement with results recently published by Stein[21] $6.0 \pm 2 \times 10^{10}$ cc/mole-sec.

Lowering the heat of formation for the phenyl radical to 78.6 Kcal/mole, the value recommended by McMillen and Golden[22], versus 81.4 Kcal/mol determined by Tsang[23] (used throughout this study) will lower the entrance barrier relative to the exit barrier for phenyl attack on chlorobenzene (non ipso), cf Figure 1.14. The effect of this change is to lower both the stabilization and the chlorobiphenyl + H product channel formation rates by 30 and 47% respectively at 1200 K and 1 Atm. This effect is much less significant where there is a larger difference in the two exit channels for the energized complex, for example, addition of H or methyl radical to chlorobenzene, Figures 1.8 and 1.12 respectively.

We feel it important to note that improvements in accuracy of the thermodynamic properties for these molecules are critical to improvements in our thermochemical based reaction mechanisms. A listing of thermochemical data used in this study is included in appendix I-C.

We have developed a detailed reaction mechanism for chlorobenzene pyrolysis in hydrogen which appears in Table I.xiv. Elementary reaction rate parameters for abstraction reactions are based upon literature comparison, thermodynamic estimations and Transition State Theory methods of Benson.[20] QRRK calculations, as described above, were used to estimate apparent rate parameters for addition and dissociation reactions (1 atm).

Experimental pyrolysis data are compared with model predictions in Figures 1.15 and 1.16 for chlorobenzene decomposition at 1126 and 1177 K in H2. Predictions for chlorobenzene loss match experiment well. Prediction of benzene formation, on the other hand, is not quite as good except at low conversions (<50%) where significant soot formation has not yet occurred. Figure 1.17 compares predicted and experimental C_6H_5Cl decay curves as functions of residence time for several temperatures. Our model (Table I.xiv) is a first-principles description of the C_6H_5Cl chain decomposition, and we feel predictions shown are quite acceptable.

Rate parameters for several reactions appearing in Table I.xiv were varied over small intervals to give a better fit of our experimental data on both chlorobenzene and dichlorobenzene reactions in hydrogen. A discussion of pyrolysis of dichlorobenzene is presented in Section II of this thesis. Optimization was performed primarily on three reactions. The A factor for chlorobenzene dissociation to phenyl radical and chlorine atom (A_1=3.0E+15) was estimated to be greater than 7.5E+14 and less than 3.0E+15 (A factor for reverse reaction, combination, less than 5E13) with activation energy (E_1 = 97.5 Kcal/mol at 298 K). The high pressure A factor for hydrogen atom addition to chlorobenzene's ipso position (A_{16}1.5E+13) was taken as being between 6.0E+12 and 2.2E+13, with E_a between 4.5 and 9.0 Kcal/mol) upper limit values are those of W Tsang[10]. Abstraction of hydrogen from H_2 by phenyl radical was assumed to have an activation energy E_a between 6.0 and 9.5 kcal/mol[29]. The A factor for this abstraction was estimated as 2.0E+13. These reactions along with the H atom combination H + H + M = H_2 + M were found to have the greatest influence (sensitivity) on predictions by the model.

Sensitivity analysis indicates that this H atom combination is the second most important reaction affecting the concentrations of chlorobenzene and benzene in the system. Manion et al.[8] (and references cited therein) have indicated that the recombination of hydrogen on the walls is efficient and that H recombination above is probably in equilibrium. Changing this recombination rate in our mechanism clearly indicated that we did not have the H in thermal equilibium with H_2. To achieve equilibrium in the system, we needed to increase the H atom recombination rate to greater than 5.0E19. This slowed the chlorobenzene decomposition dramatically, to a point where we could not match our data with realistic rate constants. Rate constants in the range of 10^{17} - 10^{18}, which are ca. one order of magnitude greater than typical reported homogeneous k's, did permit fitting of our data for both chlorobenzene and dichlorobenzene experiments (rate constants consistent in both mechanisms). We can not fit our results with Ea_{-29} larger than 6 kcal/mol for phenyl + H2 abstraction, if 2H<-->H_2 is at equilibrium. This is even more dramatically observed in our dichlorobenzene results.

The rate constant with highest sensitivity in the mechanism to chlorobenzene and benzene concentrations is the unimolecular decomposition of chlorobenzene, a chain initiation reaction. We report a best fit of the data with an A factor of 1.0E15 and an E_a of 95.5 in the temperature range of the experiment. The reaction with third highest sensitivity to the above species is H atom displacement of chlorine atoms. We report a best fit to ur data with a rate constant slightly faster than 2.2E13 exp(-9Kcal/mole/RT) reported by W. Tsang[10] with a value of 1.5E13exp(-7.5Kcal/mole/RT). While we could not fit the experimental data on chlorobenzene and dichlorobenzene with the reported rate constants of W. Tsang, we note that these numbers are very similar in the same temperature range, we have a slightly lower A and E_a, changes which are nearly offsetting.

A fourth reaction which is important in this mechanism to the primary stable reactants and products is the abstraction of H from H_2 by phenyl radicals. We use an A factor of 2.0 E13 and an E_a of 9.0 Kcal/mole. The only references to this abstraction reaction are those of Fielding and Prichard[24] and Duncan [25] who both report an Ea of 6 Kcal/mole, but Stein and Fahr[26] indicate that there is some inconsistency in the two reported rate constants, i.e. a factor of 4 difference when referenced to the one common reagent - methane, and indicate that the Ea is probably not well known. Lowering the E_a for this reaction speeds up the reaction slightly.

It is our objective to present a detailed mechanism, which is thermochemically realistic and which models chlorobenzene decay. We do not wish to convey that the rate constants are absolutely correct, but that each k is reasonable in the realm of current thermochemical kinetics.

At the higher conversions, where we overpredict benzene, it should be noted that solid carbon is a significant fraction of product (ca 30%) and that the present mechanism makes no attempt to account for this carbon formation. Since we intuitively expect benzene to be instrumental in the production of carbon in this system, this over-prediction is not surprising.

We have tried several qualitative tests to check this hypothesis. We have evaluated the addition of several irreversible reactions to our mechanism, attempting to improve agreement between predicted and experimental C_6H_6 production. Irreversible reactions such as:

$$C_6H_5^{\cdot} + C_6H_6 \text{ ---> SOOT } \{K = 10^{11}\text{exp}(-3000.0/RT)\} \quad \text{(S1)}$$

were found to slow the predicted chlorobenzene conversion, and benzene was still over predicted. The decrease in conversion is expected, however, since this is a chain termination reaction and the decomposition occurs via a chain mechanism. It is not unusual in high temperature addition reactions (complex formation) to observe C - H bond cleavage in the complex, so that one might expect an irreversible reaction of the type:

$$C_6H_5 + C_6H_6 \text{ ---> Soot + H } \{K = 10^{11}\text{exp}(-3000.0/RT)\} \quad \text{(S2)}$$

to be a viable path. Inclusion of this reaction did inhibit benzene formation somewhat without significantly decreasing the rate of chlorobenzene disappearance, but was not sufficient to explain the benzene profile.

Use of equilibria[26] such as:

$$\text{3 benzenes <--> triphenylene} + 3H_2,$$

does permit accurate modeling of the observed benzene product concentration profile (see figure 1.18). We shall report on this and on a model allowing a fraction of the benzene collisions with soot particles, concentrations estimated from mass balance and size fraction determined from Harris et. al[27].; to irreversibly stick to soot particles (as Harris did for acetylene) in a more appropriate future publication. These calculations are not, however, based on radical kinetics and do not constitute an elementary mechanism to soot. Our attempts to model soot formation from the gas phase chemistry described here, were unsuccessfull. The use of a homogeneous reaction mechanism may not adequately describe the processes which occur when a significant quantity of soot particles have been formed, since heterogeneous effects may become important.

Phenyl dissociation reactions suggested by Frenklach et. al.[28] and Kiefer[29] have not helped to account for these observations. Their reactions to form soot are not fast enough at our temperatures. Our model does not predict significant phenyl dissociation to linear C_6H_5 at our conditions due to a large excess of H_2 and high energy required for dissociation. Phenyl radical is believed to be reactive toward H_2 and as a result phenyl dissociation would not be an important reaction channel. Predictions shown in Figs. 1.14-1.17 were obtained without the use of irreversible "bleed" reactions.

We have obtained mass spectral evidence on the nonvolatile species contained in the carbon residue which are believed to be triphenylene, chlorotriphenylene, and larger PAH and phenyl adducts which support this reaction pathway. In addition, no significant amounts of acetylene, vinyl acetylene, or other ring fragmentation products were observed. In chlorinated aromatic systems, polymerization may be even more favorable due to the relative ease with which chlorine may be eliminated from the PAH/phenyl adducts to promote ring condensation and growth.

Direct phenyl polymerization was suggested by Gordon, Smith, and McNesby[30] and more recently by Stein[31] as a possible mechanism for soot growth in benzene systems where conditions such as temperature and H2 concentration make ring opening and fragmentation reactions unlikely.

Phenyl radical is believed to add to biphenyl and chlorobiphenyl producing ortho-terphenyl plus hydrogen and chlorine atoms respectively. Ortho-terphenyl is relatively unstable[21] and would either dissociate or condense to triphenylene. Ring growth may then result from successive additions of phenyl (chlorophenyl) followed by condensation to increasingly larger PAH. Such a mechanism would compete with hydrogen abstraction for phenyl radicals resulting in reduced benzene production. Inclusion of these ring addition pathways improves predictions by the mechanism slightly, similar to irreversible "bleed" reactions mentioned above.

The mechanism predicts essentially complete conversion of chlorine to HCl, while experimentally we observe only ca 82% mass balance for the Cl at 100% loss of chlorobenzene. Trace levels of chlorobiphenyls are observed in the solids by GC/Mass Spec analysis. Most other reaction studies in this laboratory, inclusive of chlorocarbon oxidations, show complete mass balance for chlorine, with HCl the major product at high conversions. One difference between this and other studies in our lab is that this experiment operated at 100 to 200 K higher temperatures. We do not know if the Cl is in the carbon solids and not extractable, if it reacts with the quartz reactor, which clearly shows vitrification with use or is lost in the heated transfer lines to the titrator. This is an important issue which needs to be resolved.

The experimental results show production of cyclopentadiene, methane, toluene, biphenyl and chlorobiphenyl as trace products or stable intermediates present at levels less than 0.5% of the initial chlorobenzene.

Model predictions for biphenyl indicate a maxima 0.2% of the initial chlorobenzene at 0.24 seconds reaction time (40% parent conversion), 1177 K, with the decrease in concentrations at longer times producing benzene. This is in fair agreement with the experimental levels of 0.05% in vapor, plus GC/Mass Spec on methylene chloride extracts of solids consistently showing significant amounts of biphenyl and chlorobiphenyl. We feel a fair fraction of the biphenyl species were lost by adsorption on particulate in the reactor exit and sample transfer line to the GC. The model predicts chlorobiphenyl levels approximately 0.02% with a maxima at 0.28 seconds corresponding to 45% conversion of the parent chlorobenzene and the experimental levels are 0.05% of the initial chlorobenzene.

Cyclopentadiene levels predicted by the model are ca. 0.01% of $CLBZ_O$, where the observed concentrations are between 0.1 and 0.5%. The model and the experiment both show cyclopentadiene levels increasing with temperature, reaction time and/or conversion. Experimental observations on toluene show that it is present in concentrations below 0.5% with the model not predicting these levels. Methane is predicted at the 0.01% level, with experimental levels ranging between 0.04 and 0.06%.

While one can describe the reagent and major product concentration profiles with a less detailed reaction mechanism than that used here, we wanted to present a more complete reaction description by trying to model the observed high molecular weight and trace level species. We feel it is worthwhile to the include these reactions; they illustrate plausible pathways and product concentrations. The model calculations also indicate that inclusion of ring addition reactions, which terminate at formation of biphenyls and chlorobiphenyls in this mechanism, do not explain the observed benzene loss at high conversions. Work is ongoing to explain formation of the observed PAH's plus multi - phenyl ring systems and especially the benzene loss. The reactions listed form a complete subset in mechanisms which accurately describe our dichlorobenzene results and our separate attempts to predict the observed PAH's (up to 4 fused rings) plus triphenylene and chloro triphenylene.

REFERENCES SECTION I

[1] Rubey, W. A.; Dellinger, B.; Hall, D. L. and Mazer, S.L., Chemosphere, **1985**, 14, 1483.

[2] Chang, W. D.; Karra, S. B. and Senkan, S. M., Environ. Sci. Technol., **1986**, 20, 1243.

[3] Chuang, S. C. and Bozzelli, J., Environ. Sci. Technol., **1986**, 20, 568.

[4] Manion, J.; Mulder, P. and Louw, R., Environ. Sci. Technol., **1985**, 19, 280.

[5] Louw, R.; Dijks, J. H. and Mulder, P., J. Chemistry and Industry, Oct. 3, **1983**, 759.

[6] Louw, R.; Dijks, J. H. and Mulder, P., J. Chem. Soc. Perkin Trans. II, **1973**, 40, 1635.

[7] Louw, R.; Dijks, J. H. and Mulder, P., Recuell, J. Roy. Neth. Chem. Soc. **1984**, 103, 271.

[8] Manion, J. A.,Dijks,J H M., Mulder, P. and Louw, R., Recl. Trav. Chim. Pays-Bas **1988**,107 434-39.

[9] Benson, S. and Shaw, R. J. Chem. Phys., **1967**, 10, 4052.

[10] Tsang W., Eastern States Combustion Institute Meeting - Proceedings Dec 5-7 1988, Clearwater Beach, Florida. J. Phys. Chem.**93** (1989).

[11] Frenklach, M., Ramachandra, M. and Matula, A., Symposium (Int.) on Combustion, **1984**, 20, 871.

[12] Rao, V. S. and Skinner, G. B. (preprint submitted to) J. Phys. Chem., June, **1987**.

[13] Cullis, C. and Manton., Proc. Roy. Soc. A, **1954**, 224, 308.

[14] Dean, A. M.,J. Phys. Chem. **1985**, 89, 4600.

[15] Westmoreland, P. R.; Howard, J. B.; Longwell, J. P. and Dean, A. M., AICHE Journal, **1986**, 32, 1971.

[16] Dean, A. M. and Westmoreland, P. R. Int. J. Chem. Kin., **1987**, 19, 207.

[17] Chang, S. H. and Bozzelli, J. W. AICHE Journal, **1987**, 33, 1207.

[18] Kaufman, F. Progress in Reaction Kinetics V 1, ed. G. Porter, (New York: Pergamon Press, **1961**).

[19] Rao, V. and Skinner, G.; J. Phys. Chem., **1984**, 88, 5590.

[20] Benson, S., Thermochemical Kinetics (New York: John Wiley & Sons, (**1976**).

[21] Stein, S. and Fahr, A., 22nd Symp. International on Combustion Proc., The Combustion Institute #107 1989.

[22] McMillen, D.F. and Golden, D. M., Ann. Rev. Phys. Chem., **1982** 33, 493 - 532.

[23] Tsang, W. and Robaugh, D., J. Phys. Chem.,**1986**,90, 5363 - 5367.

[24] Fielding, W. and Prichard, H.O., J. Phys. Chem., **1962**, 66, 821.

[25] Duncan, F. J. and Trotman-Dickinson, A. F., J. Chem. Soc., **1962**, p 462.

[26] Stein, S. and Fahr, A., J. Phys. Chem. **1988**, 92, 4951.

[27] Harris, S. J., Weiner, A. and Ashcoft, C. C., Combustion and Flame, **1986** 64, p 65.

[28] Frenklach, M. and Warnatz, J., Combust. Sci. Tecnol. **1987**, 51, 265.

[29] Kiefer, J. H.; Mizerka, L. J.; Patel, M. R. and Wei, H. C., J Phys Chem, **1985**, 89, 2013.

[30] Gordon, A. S.; Smith, S. R. and McNesby, J. R. Symposium (Int.) on Combustion, **1959**, 7, 317.

[31] Stein, S. E. J. Phys. Chem. **1978**, 82, 566.

APPENDIX I - A

TABLES FOR SECTION I

TABLE I.i

$C6H5Cl + H = [ClCHD^{\cdot}]^{\#} = C6H6 + Cl$

(see Figure 1.8 for Potential Energy Diagram)

k	A^{*}	Ea^{**}	source
1	1.5 E+13	7.5	a
-1	3.6 E+12	31.0	a
2	1.16 E+13	8.9	b
<v> = 1064/cm			c
LJ PARAMETERS :			d
sigma = 5.69 A		e/k = 526.9 cal	

[a] This work, Reverse reaction (k-1) from thermodynamics.

[b] based upon (del S) for $Cl+C6H6 = [Cl\text{-}CHD^{\cdot}]$ with A_{-2} = 2.0E+13 cc/mol sec

[c] Geometric mean frequency estimated as follows:
$<v>_{Cl\text{-}CHD.} = <v>_{C6H5Cl} + <v>_{C6H7} - <v>_{C6H6}$

[d] Lennard Jones parameters for radical complexes are estimated by averaging two structurally similar molecules with mixing rules of Prausnitz (Molecular Thermodynamics of Fluid Phase Equilibria, Prentice-Hall, 1969).
For example: $(E/K)_{CHD.} = ((E/K)_{BENZENE}*(E/K)_{CHD})^{1/2}$
and $Sigma_{CHD.} = (\tilde{O}_{BENZENE} + \tilde{O}_{CHD})/2$

Molecular L-J parameters are estimated using the correlation of Tee, Gotoh, and Stewart (I. E. & C. Fundamentals, 1966, 5, 363). Required critical property data tabulated in Reid, Prausnitz, and Sherwood (Properties of Gases and Liquids, 3rd ed.) or from Lydersen's group method for critical property estimation (also in Reid, Prausnitz, and Sherwood).

UNITS:
* A_1 (cc/mol sec)
A_2 (sec^{-1})
** E_a (Kcal/mol)

TABLE I.ii

$C6H6 + H = [CHD^{\cdot}]^{\#} = PRODUCTS$

(see Figures 1.9 and 1.10 for Potential Energy Diagrams)

k	A*	Ea**	source
1	4.0 E+13	4.3	a
-1	1.53 E+13	26.0	a
2	2.0 E+13	48.9	b
3	5.0 E+12	38.1	c
-3	8.0E+11	40.0	c
<v> = 1224/cm			d
LJ PARAMETERS :			e
sigma = 5.40 A	e/k = 464.3 cal		

a
Tsang, W. J. Phys. Chem. **1986**, 90, 1152.
Reverse reaction (k-1) from thermodynamics.

b
E_a = delta H_f : A factor for beta scission taken as that for C=CCC˙C=C = C=C˙ + C=CC=C
Dean, A. M.J. Phys. Chem. **1985**, 89, 4600.

c
E_3 = delta H_f + 7.7 kcal/mol. A_3 is estimated from the reverse reaction: bicyclo-C_6H_7 beta scission to cyclohexadienyl radical (as in Figure 9). $A_{-3}=10^{12.75}$ estimated from beta scission for primary and secondary butyl radicals (Dean, A. M.J. Phys. Chem. **1985**, 89, 4600). E_3 is taken as the energy difference between $[CHD^{\cdot}]^{o}$ and methylene cyclopentadienyl radical plus E_a for 1,2-Hydrogen shift to methyl cyclopentadienyl radical (see Fig. 9).

d
Tsang, W. J. Phys. Chem. **1986**, 90, 1152.

e
See note (d) Table I.i

UNITS:
* A_1 (cc/mol sec)
A_2 (sec^{-1})
A_3 (sec^{-1})
** E_a (Kcal/mol)

TABLE I.iii

APPARENT REACTION RATE CONSTANTS PREDICTED

USING BIMOLECULAR QRRK ANALYSIS

Bath Gas	P (torr)	Reaction	A factor (cc/mol s)	n	Ea (kcal/mole)	range
H2	7.6	H+C6H6 = [ME-CYPD·]^{o}	1.65 E+55	-12.21	41.0	c
	760.0	"	2.39 E+27	-3.92	29.2	"
	7.6	H+C6H6 = [CHD·]^{o}	2.79 E+49	-11.55	13.59	"
	760.0	"	4.87 E+56	-12.73	26.8	"
	7.6	H+C6H6 = Lin-C6H7	7.96 E+16	-0.88	28.8	a
	760.0	"	1.67 E+21	-2.01	34.1	"
N2	7.6	H+C6H6 = [ME-CYPD·]^{o}	1.05 E+60	-13.66	43.2	c
	760.0	"	5.22 E+28	-4.35	28.8	b
	7.6	H+C6H6 = [CHD·]^{o}	4.17 E+46	-10.87	10.87	c
	760.0	"	8.18 E+57	-13.19	26.3	b
	7.6	H+C6H6 = Lin-C6H7	5.04 E+16	-0.82	28.7	a
	760.0	"	9.47 E+19	-1.68	32.5	"

[a] Valid from 300 - 1900 K
[b] " " 600 - 1900 K
[c] " " 900 - 1900 K

TABLE I.iv

$C6H6 + CH3 = [Me\text{-}CHD^{\cdot}]^{\#} = C6H5CH3 + H$

(see Figure 1.12a for Potential Energy Diagram)

k	A*	Ea**	source
1	2.9 E+12	11.1	a
-1	2.1 E+14	22.8	a
2	2.7 E+13	26.1	b

<v> = 1200/cm   c

LJ PARAMETERS :   d

sigma = 5.74 A   e/k = 494 cal

[a] Tsang, W. J. Phys. Chem. **1986**, 90, 4159.
Reverse reaction (k-1) from thermodynamics

[b] based upon (del S) for C6H5CH3+H = [Me-CHD˙]
with A_{-2}=1.2E+13(a)

[c] Geometric mean frequency estimated as follows:
$<v>_{Me\text{-}CHD.} = <v>_{C6H5CH3} + <v>_{C6H7} - <v>_{C6H6}$

[d] See note (d) Table I.i

UNITS:
* A_1 (cc/mol sec)
A_2 (sec^{-1})

** E_a (Kcal/mol)

TABLE I.v

APPARENT REACTION RATE CONSTANTS PREDICTED

USING BIMOLECULAR QRRK ANALYSIS

Bath Gas	P (torr)	Reaction	A factor (cc/mol s)	n	Ea (kcal/mole)	range
H2	7.6	CH3+C6H6 = [ME-CHD·]^{o}	6.64 E+29	-6.93	-4.8	c
	760.0	"	1.49 E+57	-13.84	28.3	b
	7.6	CH3+C6H6 = C6H5CH3+H	6.22 E+15	-1.20	17.1	a
	760.0	"	3.41 E+26	-4.15	26.4	b
N2	7.6	CH3+C6H6 = [ME-CHD·]^{o}	1.59 E+27	-6.29	2.8	c
	760.0	"	5.30 E+48	-11.60	21.2	c
	7.6	CH3+C6H6 = C6H5CH3+H	4.56 E+14	-0.88	16.3	a
	760.0	"	8.65 E+23	-3.44	24.1	b

[a] Valid from 300 - 1900 K
[b] " " 600 - 1900 K
[c] " " 900 - 1900 K

TABLE I.vi

C6H5Cl + CH3 = [Me-Cl-CHD˙]# = C6H5CH3 + Cl

(see Figure 1.12b for Potential Energy Diagram)

k	A*	Ea**	source
1	1.45 E+12	11.1	a
-1	8.54 E+12	26.5	a
2	5.6 E+13	11.1	b

<v> = 1070/cm c

LJ PARAMETERS : d

sigma = 5.99 A

e/k = 548 cal

a
A factor taken as 1/2 that for CH3+C6H6 (Table I.iiia)

Reverse reaction (k-1) from thermodynamics.

b
based upon (del S) for C6H5CH3+Cl = [Me-Cl-CHD˙] with A_{-2}=1.0E+12(a)

c
Geometric mean frequency estimated as follows:
$<v>_{Me-Cl-CHD.} = <v>_{Me-CHD.} + <v>_{C6H5Cl} - <v>_{C6H6}$

d
See note (d) Table I.i

UNITS:
* A_1 (cc/mol sec)
A_2 (sec^{-1})

** E_a (Kcal/mol)

TABLE I.vii

APPARENT REACTION RATE CONSTANTS PREDICTED

USING BIMOLECULAR QRRK ANALYSIS

Bath Gas	P (torr)	Reaction	A factor (cc/mol s)	n	Ea (kcal/mole)	range
H2	7.6	CH3+C6H5Cl=[ME-Cl-CHD˙]^{o}	5.91 E+25	-5.78	13.5	a
	760.0	"	1.98 E+28	-5.93	13.9	"
	7.6	CH3+C6H5CL = C6H5CH3+Cl	5.15 E+13	-0.49	11.6	"
	760.0	"	1.96 E+14	-0.65	12.0	"
N2	7.6	CH3+C6H5Cl=[ME-Cl-CHD˙]^{o}	2.31 E+25	-5.76	13.5	"
	760.0	"	4.17 E+27	-5.83	13.7	"
	7.6	CH3+C6H5Cl = C6H5CH3+Cl	5.11 E+13	-0.49	11.6	"
	760.0	"	9.80 E+13	-0.53	11.8	"

[a] Valid from 300 - 1900 K

TABLE I.viii

$C6H5Cl + C6H5 = [Ph\text{-}Cl\text{-}CHD^{\cdot}]^{\#} = C6H5C6H5 + Cl$

(see Figure 1.13 for Potential Energy Diagram)

k	A^{*}	Ea^{**}	source
1	1.0 E+12	4.3	a
-1	1.6 E+15	33.2	a
2	1.9 E+14	10.0	b

<v> = 975/cm		c
LJ PARAMETERS :		d
sigma = 6.66 A	e/k = 694 cal	

a
Louw, R.; Dijks, J. H.; Mulder, P. J. Chem. Soc. Perkin Trans. II, 1973, 40, 1635.

Reverse reaction (k-1) from thermodynamics.

b
A factor based upon entropy change for reverse.
Cl + Ph-Ph = [Ph-Cl-CHD$^{\cdot}$] (ipso attack)
with A_{-2} = 1.2 E+13 and E_{-2} = 2.0 kcal/mol.

c
Geometric mean frequency estimated as follows:
$<v>_{Ph-Cl-CHD.} = <v>_{Ph-Ph} + <v>_{Cl-CHD.} - <v>_{C6H6}$

d
See note (d) Table I.i

UNITS:
* A_1 (cc/mol sec)
A_2 (sec^{-1})

**E_a (Kcal/mol)

TABLE I.ix

APPARENT REACTION RATE CONSTANTS PREDICTED

USING BIMOLECULAR QRRK ANALYSIS

Bath Gas	P (torr)	Reaction	A factor (cc/mol s)	n	Ea (kcal/mole)	range
H2	7.6	C6H5+C6H5Cl=[Ph-Cl-CHD˙]^{o}	4.45 E+22	-5.06	5.7	a
	760.0	"	7.97 E+24	-5.13	5.87	"
	7.6	C6H5+C6H5CL = C6H5C6H5+Cl	9.85 E+11	0.0	4.29	"
	760.0	"	2.59 E+12	-0.12	4.53	"
N2	7.6	C6H5+C6H5Cl=[Ph-Cl-CHD˙]^{o}	1.68 E+22	-5.03	5.7	"
	760.0	"	2.17 E+24	-5.06	5.79	"
	7.6	C6H5+C6H5CL = C6H5C6H5+Cl	9.85 E+11	0.0	4.29	"
	760.0	"	1.0 E+12	0.0	4.32	"

[a] Valid from 300 - 1900 K
[b] " " 600 - 1900 K
[c] " " 900 - 1900 K

TABLE I.x

C6H5Cl + C6H5 = [Ph-Cl-CHD·]$^{\#}$ = C6H5C6H4Cl + H

(see Figure 1.14 for Potential Energy Diagram)

k	A*	Ea**	source
1	2.0 E+12	4.3	a
-1	2.2 E+15	32.74	a
2	6.7 E+13	24.5	b
<v> = 975/cm			c
LJ PARAMETERS :			d
sigma = 6.66 A		e/k = 694 cal	

[a] Louw, R.; Dijks, J. H.; Mulder, P. J. Chem. Soc. Perkin Trans. II, 1973, 40, 1635.

Reverse reaction (k-1) from thermodynamics.

[b] A factor based upon entropy change for reverse.
H + Cl-Ph-Ph = [Cl-Ph-CHD·]
with A_{-2} = 1.3 E+13 and E_{-2} = 4.0 kcal/mol.

[c] Geometric mean frequency estimated as follows:
$\langle v\rangle_{Ph-Cl-CHD.} = \langle v\rangle_{Ph-Ph} + \langle v\rangle_{Cl-CHD.} - \langle v\rangle_{C6H6}$

[d] See note (d) Table I.i

UNITS:

* A_1 (cc/mol sec)
A_2 (sec^{-1})

** E_a (Kcal/mol)

TABLE I.xi

APPARENT REACTION RATE CONSTANTS PREDICTED

USING BIMOLECULAR QRRK ANALYSIS

Bath Gas	P (torr)	Reaction	A factor (cc/mol s)	n	Ea (kcal/mole)	range
H2	7.6	C6H5+C6H5Cl=[Ph-Cl-CHD·]^{o}	6.82 E+47	-12.12	6.62	c
	760.0	"	2.80 E+69	-17.31	29.2	b
	7.6	C6H5+C6H5CL=C6H5C6H4CL+H	9.52 E+19	-2.51	7.08	c
	760.0	"	4.50 E+49	-10.65	33.3	b
N2	7.6	C6H5+C6H5Cl=[Ph-Cl-CHD·]^{o}	3.34 E+42	-10.76	2.36	c
	760.0	"	5.42 E+72	-18.39	29.94	b
	7.6	C6H5+C6H5CL=C6H5C6H4CL+H	3.43 E+32	-6.04	15.97	"
	760.0	"	4.50 E+49	-10.65	31.95	"

[a] Valid from 300 - 1900 K
[b] " " 600 - 1900 K
[c] " " 900 - 1900 K

TABLE I.xii

C6H6 + C6H5 = [Ph-CHD·]$^{\#}$ = C6H5C6H5 + H

k	A*	Ea**	source
1	2.0 E+12	4.3	a
-1	3.9 E+14	29.4	a
2	1.1 E+13	23.55	b
<v> = 1050/cm			c
LJ PARAMETERS :			d
sigma = 6.45 A		e/k = 662 cal	

[a] Louw, R.; Dijks, J. H.; Mulder, P. J. Chem. Soc. Perkin Trans. II, 1973, 40, 1635.

Reverse reaction (k-1) from thermodynamics.

[b] A factor based upon entropy change for reverse.
H + Ph-Ph = [Ph-CHD·]
with A_{-2} = 2.0 E+13 and E_{-2} = 4.0 kcal/mol.

[c] Geometric mean frequency estimated as follows:
$<v>_{Ph-CHD.} = <v>_{Ph-Ph} + <v>_{CHD.} - <v>_{C6H6}$

[d] See note (d) Table I.i

UNITS:

* A_1 (cc/mol sec)
A_2 (sec^{-1})

$^{**}E_a$ (Kcal/mol)

TABLE I.xiii

APPARENT REACTION RATE CONSTANTS PREDICTED

USING BIMOLECULAR QRRK ANALYSIS

Bath Gas	P (torr)	Reaction	A factor (cc/mol s)	n	Ea (kcal/mole)	range
H2	7.6	C6H5 + C6H6 = [Ph-CHD˙]^{o}	1.92 E+45	-11.24	6.61	c
	760.0	"	3.30 E+60	-15.18	26.43	"
	7.6	C6H5+C6H6 = C6H5C6H6+H	1.21 E+20	-2.56	7.29	b
	760.0	"	1.61 E+45	-9.39	31.0	"
N2	7.6	C6H5 + C6H6 = [Ph-CHD˙]^{o}	1.09 E+40	-9.89	2.38	c
	760.0	"	4.11 E+56	-14.15	16.42	"
	7.6	C6H5+C6H6 = C6H5C6H6+H	1.91 E+30	-5.42	14.64	"
	760.0	"	7.31 E+26	-4.42	13.11	"

[a] Valid from 300 - 1900 K
[b] " " 600 - 1900 K
[c] " " 900 - 1900 K

TABLE I.xiv
REACTION MECHANISM#

	REACTION	A	n	Ea(kcal/mol)	source
1.	C6H5CL=CYC6H5+CL	3.0 E+15	0.0	95.5	a
2.	CYC6H6=CYC6H5+H	1.67E+16	0.0	111.5	b
3.	CYC6H7=MECY24PD1	5.0 E+12	0.0	38.1	c
4.	MECY24PD=CY13PD5+CH3	1.0 E+16	0.0	67.5	d
5.	CY13PD=CY13PD5+H	6.0 E+14	0.0	75.1	e
6.	C6H5C6H5= CYC6H5+CYC6H5	5.0 E+16	0.0	117.0	g
7.	PHPHCL=CYC6H5+C6H4CL	1.9 E+16	0.0	117.0	h
8.	CYC6H5=LINC6H5	3.16E+13	0.0	55.2	i
9.	LINC6H5=C#CC*C.+C2H2	6.09E+14	0.0	47.6	j
10.	LINC6H7=C*CC*C.+C2H2	5.50E+14	0.0	41.0	j
11.	H+CL+M=HCL+M	1.0 E+17	0.0	0.0	k
12.	CL2=CL+CL	7.69E+08	0.0	55.6	k
13.	H+H+M=H2+M	1.0 E+18	0.0	0.0	m
14.	C6H5CL=C6H4CL+H	1.3 E+16	0.0	110.5	f
15.	CYC6H6+H=MECY24PD1	2.39E+27*	-3.92	29.2	l
16.	C6H5CL+H=CYC6H6+CL	1.5 E+13*	0.0	7.5	l
17.	CYC6H6+H=CYC6H7	4.87E+56*	-12.73	26.8	l
18.	CYC6H6+H=LINC67H7	1.22E+22*	-1.87	31.2	l
19.	CH3+C6H5CL=C6H5CH3+CL	1.96E+14*	-0.65	12.0	l
20.	CH3+CYC6H6=C6H5CH3+H	3.4 E+26*	-4.15	26.4	l
21.	CH3+H=CH4	8.09E+36*	-7.19	9.2	l
22.	CYC6H5+C6H5CL=PHPHCL+H	4.5 E+49*	-10.65	33.3	l
23.	CYC6H5+C6H5CL=C6H5C6H5+CL	2.6 E+12*	-0.12	4.5	l
24.	CYC6H5+CYC6H6=C6H5C6H5+H	1.6 E+45*	-9.39	31.0	l
25.	C6H4CL+C6H4CL=PHCLPHCL	1.8 E+13*	0.0	0.0	l
26.	C6H4CL+C6H5CL=PHPHCL+CL	2.6 E+12*	-0.13	4.43	l
27.	C6H4CL+C6H5CL=PHCLPHCL+H	5.43E+15*	-1.39	3.85	l
28.	C6H4CL+CYC6H6=PHPHCL+H	9.92E+37*	-5.48	19.97	l
29.	CYC6H6+H=CYC6H5+H2	2.0 E+13	0.0	18.6	m
30.	CH3+H2=CH4+H	5.0 E+12	0.0	11.0	n
31.	C6H5CL+H=CYC6H5+HCL	1.0 E+13	0.0	11.3	m
32.	H+C6H5CL=H2+C6H4CL	2.0 E+13	0.0	18.6	m
33.	CL+H2=HCL+H	4.8 E+13	0.0	5.0	p
34.	CL+C6H5CL=HCL+C6H4CL	1.0 E+13	0.0	12.5	m
35.	CL+CYC6H6=HCL+CYC6H5	1.1 E+13	0.0	12.5	m
36.	H+CL2=CL+HCL	7.94E+13	0.0	1.2	p
37.	MECY24PD1+H=CY13PD+CH3	8.0 E+13	0.0	0.0	q
38.	CYC6H7=ME.CY24PD	5.5 E+10	0.0	23.5	r
39.	ME.CY24PD+H2=MECY24PD+H	3.98E+12	0.0	14.0	s

* Pressure dependent : rate expression given for 760 torr
Temperature ranges given in Tables I.ii-I.xiv.

# See Figures 1.15 - 1.17 for comparison of experimental data with model results.

References for Mechanism (Table I.xiv)

a A factor from reverse with A_{-1} = 9 E+12: this work.

b A factor based upon reverse with A_{-2} = 1 E+14. H addition to secondary C_4 radicals (Allara, D. L. and Shaw, R. J. Phys. Chem. Ref. Data, 1980, 9, 523).

c see Table II : Qrrk/Energized complex analysis of H+ C6H6.

d A factor from reverse with A_{-4} = 1 E+13. CH_3 addition to C_4 radicals (Allara, D. L. and Shaw, R. J. Phys. Chem. Ref. Data, 1980, 9, 523).

e A factor from reverse with A_{-5} = 1 E+14: see ref. b.

f A factor from reverse with A_{-14} = 7.5E+13: see ref. b.

g A factor from reverse with A_{-6} = 1 E+12 (Louw, R; Dijks, J. H.; Mulder, P. J. Chem. Soc. Perkin Trans. II, 1984, 103, 271).

h see ref. g.

i taken as linear C_6H_7 = C_4H_7 + C_2H_2 (Dean, A. M. J. Phys. Chem. 1985, 89, 4600).

j A factor and Ea from C=CC=C$^{\cdot}$+C_2H_2 (see ref. i).

k Kerr, J. A. and Moss, S. J. eds Handbook of Bimolecular and Termolecular Gas Reactions V I and II CRC Press, Inc. 1981.

l QRRK/ Energized complex analysis presented in this work (Tables I.i - I.xiii) or by analogy to those in tables.

m optimized reaction : this work

n CH_3 + H_2 from Allara and Shaw (see ref. d).

o C_3H_5 + H_2 from Allara and Shaw (see ref. d).

p (Kerr, J. A. and Moss, S. J. eds Handbook of Bimolecular and Termolecular Gas Reactions Vol. I and II CRC Press, Inc. 1981).

q see ref. b.

r Pressure dependence accounted for with QRRK analysis of Dean (see ref. j).

s A factor taken as CCC. + H2 (Allara, D. L. and Shaw, R. J. Phys. Chem. Ref. Data, 1980, 9, 523).

APPENDIX I - B

FIGURES FOR SECTION I

Figure 1.1

SCHEMATIC DIAGRAM OF EXPERIMENTAL APPARATUS

- Tubular Flow Reactors
- Temp Constant to ± 5 Degrees
- Chlorobenzene:H2 270:1
- GC AND GC/Mass Spec Analysis

VARIABLES
- Reaction Time (0.02 to 2.5 sec)
- Temperature (1100 to 1300 K)
- Surface to Volume Ratio (10 to 2.5/cm)

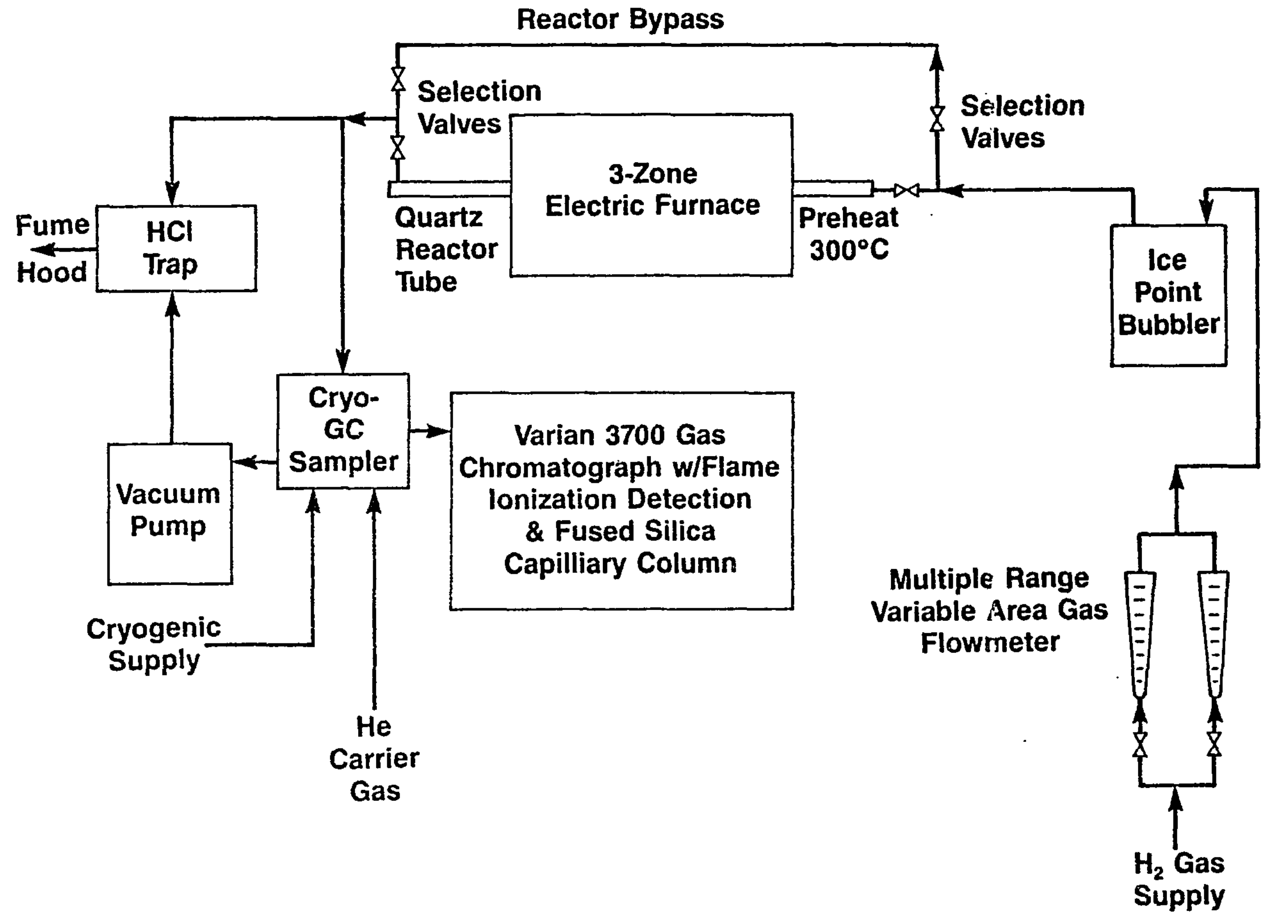


Figure 1.2

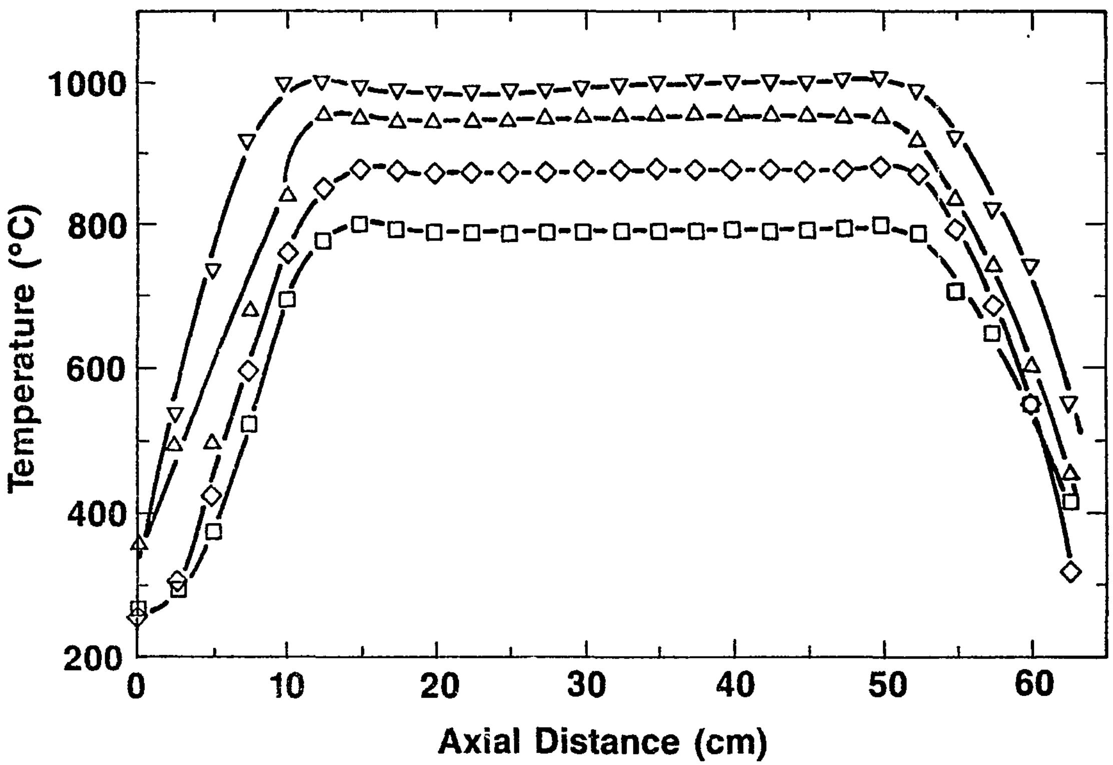
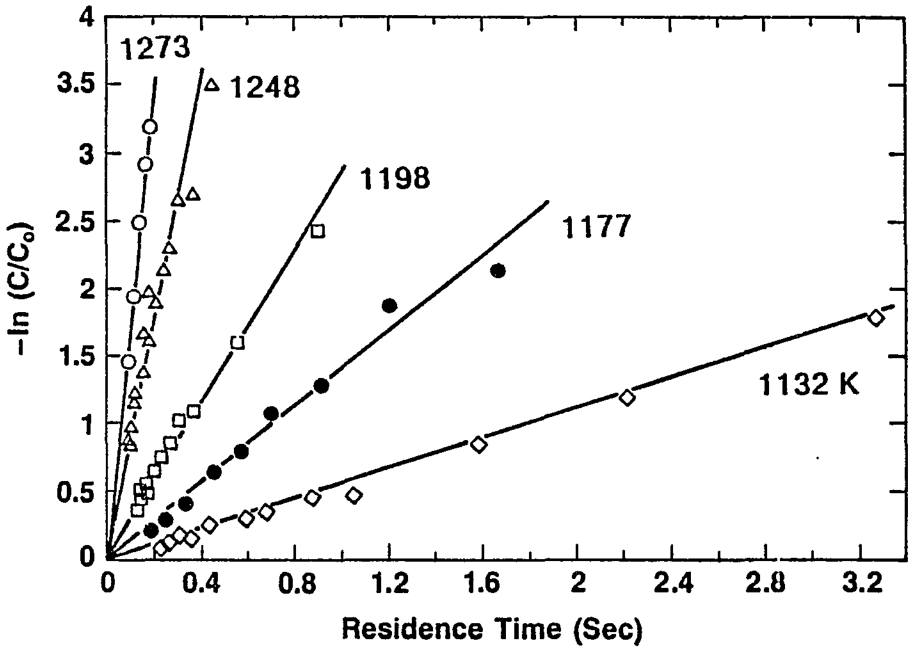

1000
800
600
400
200
0
10
20
30
40
50
60
Temperature (°C)
Axial Distance (cm)

Figure 1.:3

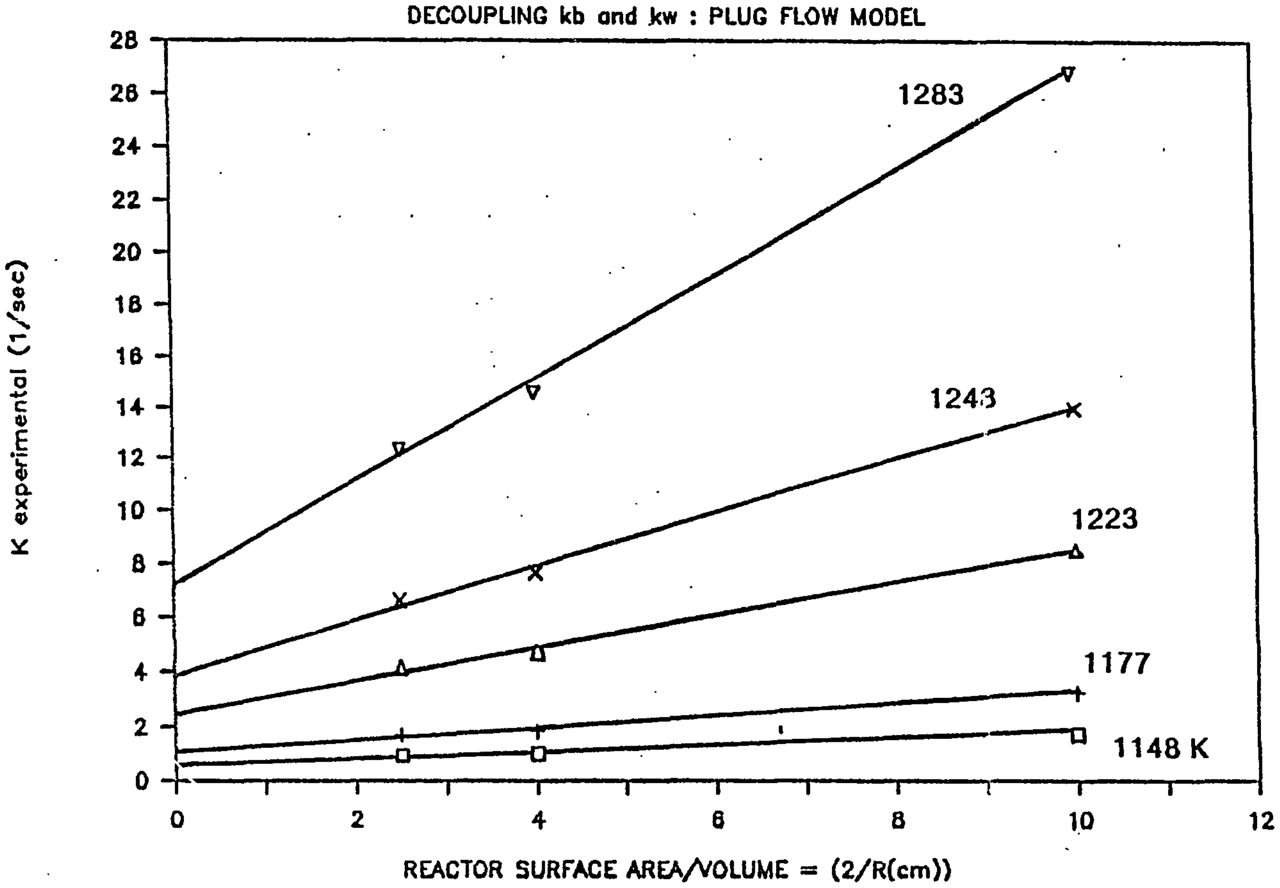
PSEUDO FIRST ORDER REACTION   H2 EXCESS



OVERALL REACTION EXHIBITS FIRST ORDER DECAY
FOR CHLOROBENZENE LOSS

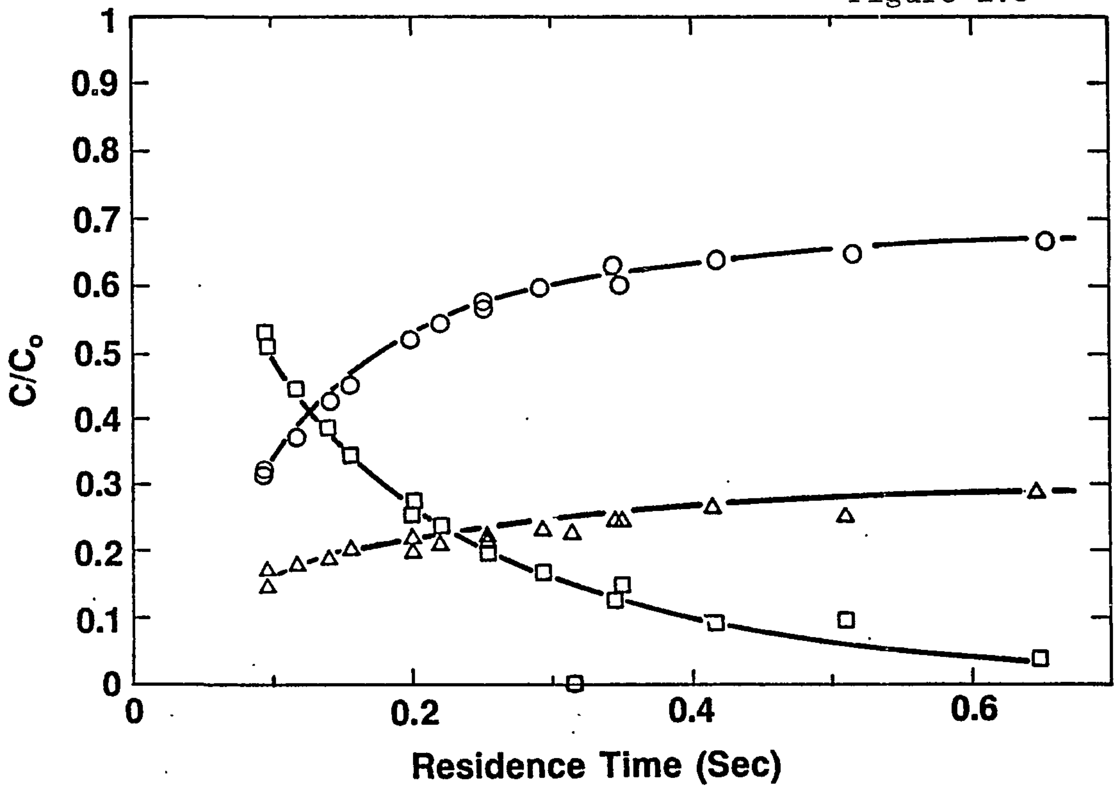
VARIATION IN OBSERVED RATE CONSTANT WITH REACTOR SURFACE TO VOLUME RATIO (2/r)

Figure 1.4



- Global Rate Constant is Function of S/V for Small Diameter Reactors
- Data Allows Separation of BULK Rate Constants from Wall Effects

Figure 1.5



CHLOROBENZENE + H2 ---> PRODUCTS

1 Second Reaction Time

Figure 1.6

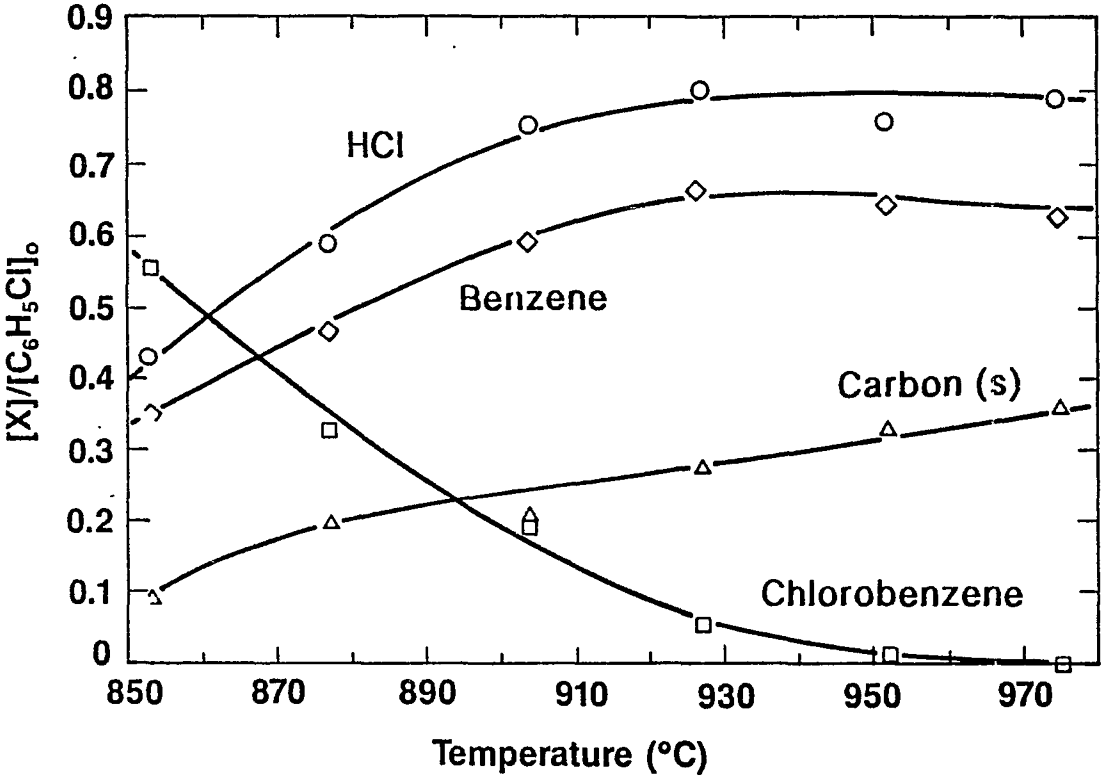


Figure 1.7

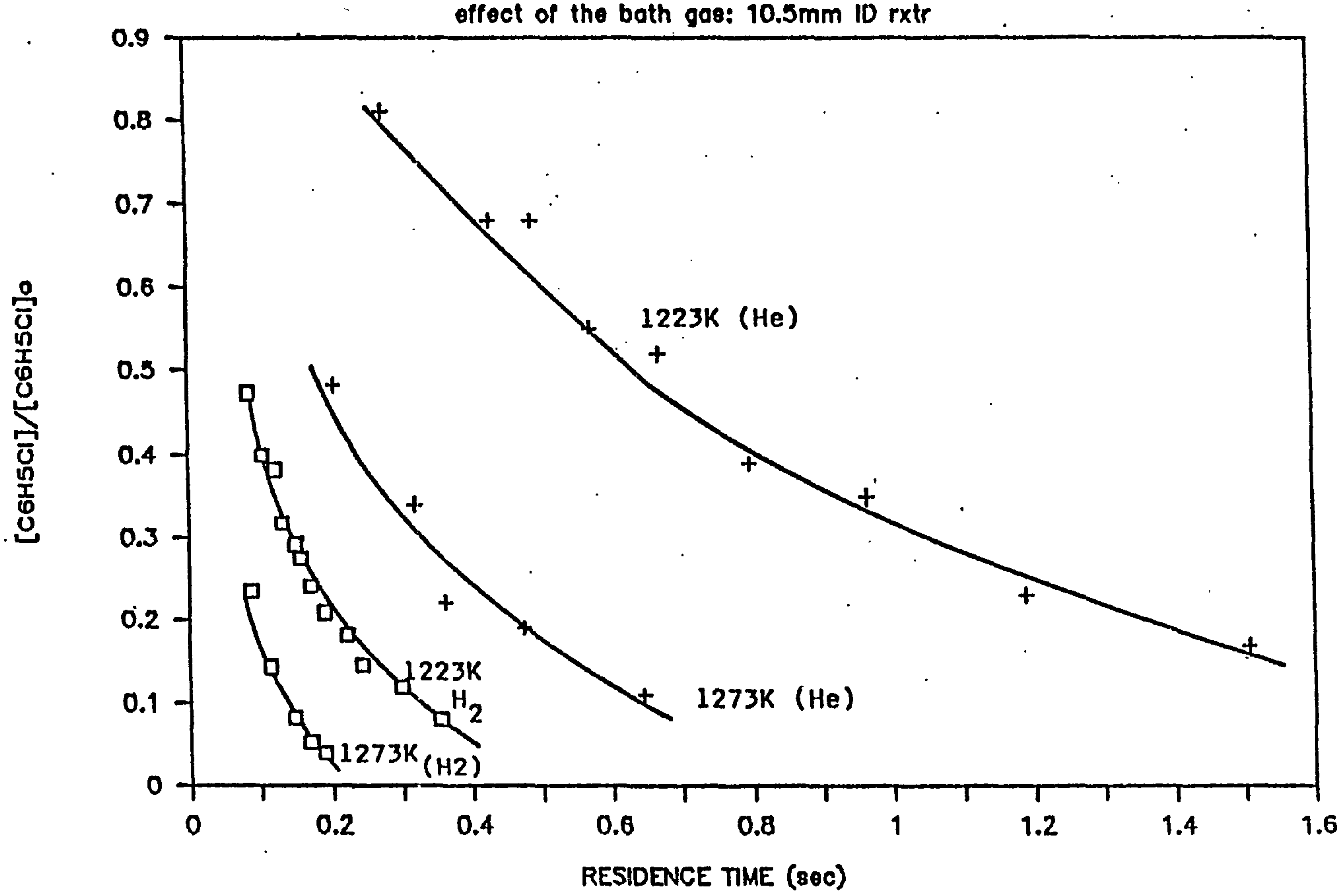

effect of the bath gas: 10.5mm ID rxtr
[C6H5Cl]/[C6H5Cl]o
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
RESIDENCE TIME (sec)
1223K (He)
1273K (He)
1223K
H_2
1273K (H2)

Figure 1.8

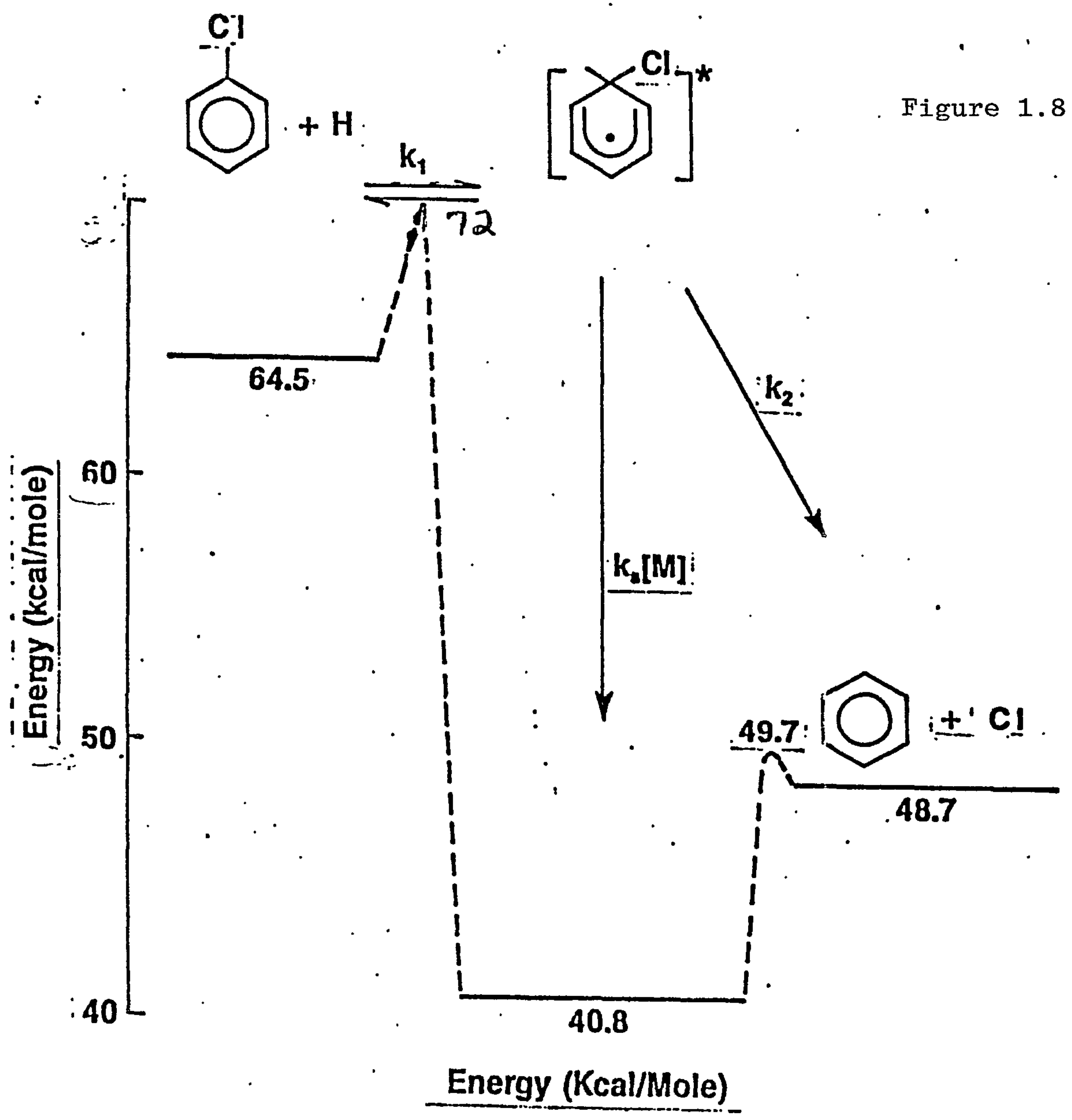

Cl
+ H
k_1
72
Cl
*
k_s[M]
k_2
64.5
60
50
40
Energy (kcal/mole)
49.7
+ Cl
48.7
40.8
Energy (Kcal/Mole)

Figure 1.9

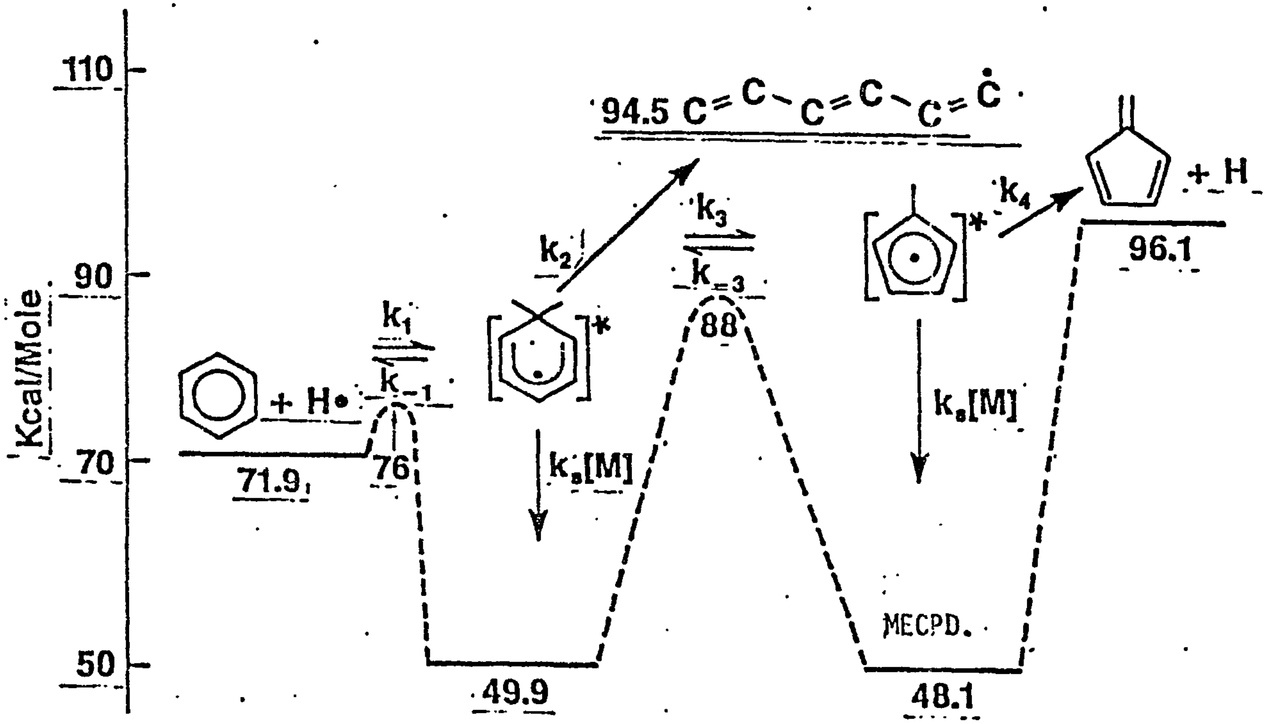
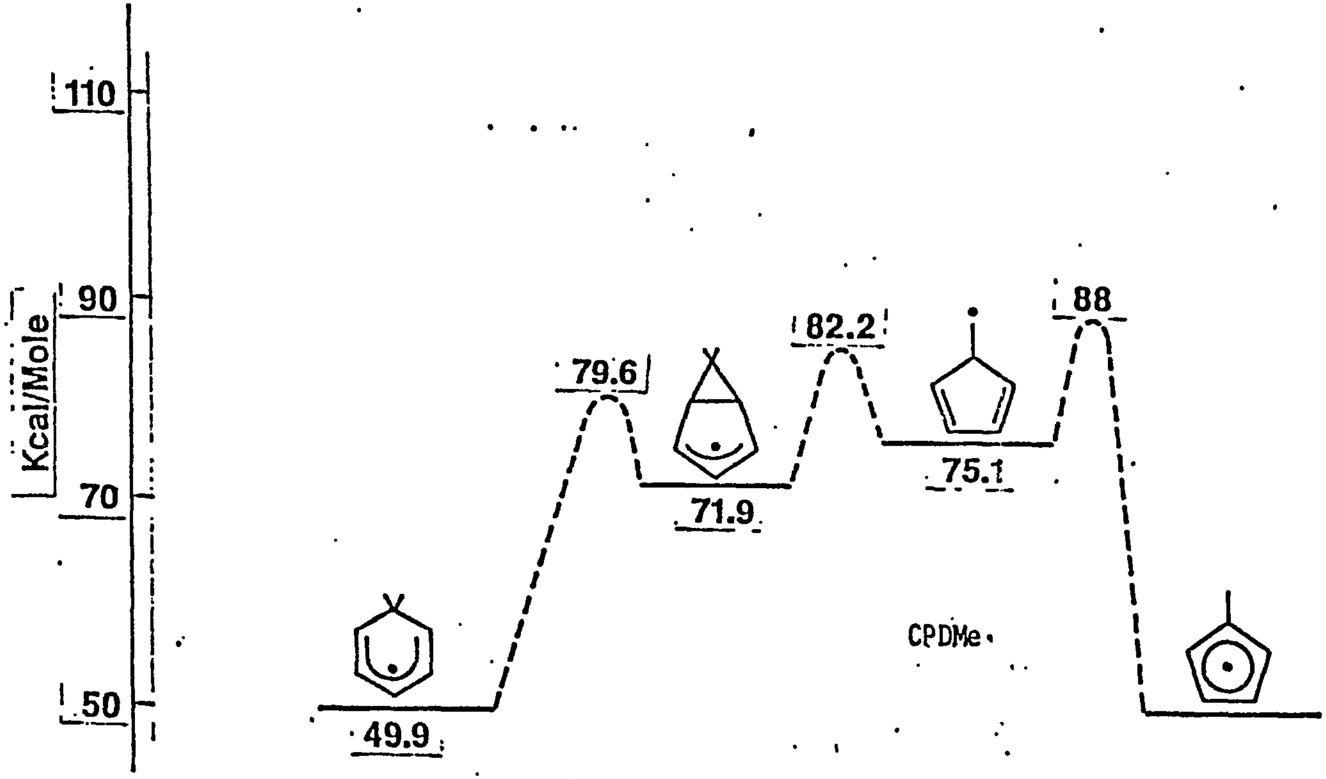

Kcal/Mole
110
90
70
50
+ H•
71.9
k_1
k_{-1}
76
k_2
*
$k_s[M]$
49.9
94.5 C=C–C=C–C=Ċ
k_3
k_{-3}
88
*
k_4
$k_s[M]$
MECPD.
48.1
+ H
96.1

Figure 1.10



H + C6H6 ADDITION PRODUCTS vs PRESSURE APPARENT RATE CONSTANTS at 1200 K

Figure 1.11

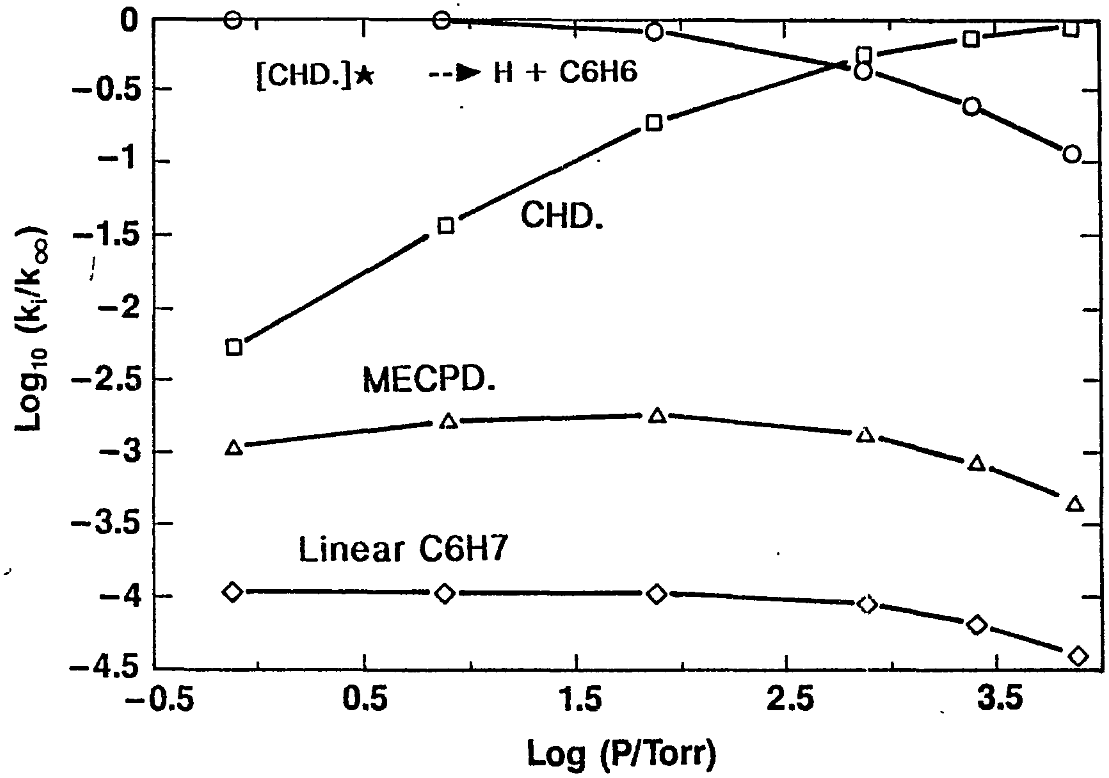


Figure 1.12a

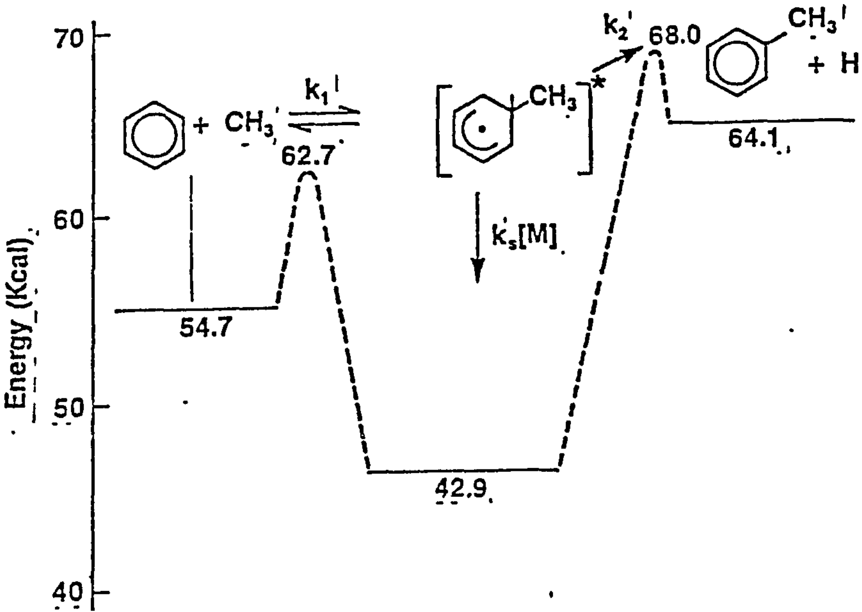


Figure 1.12b

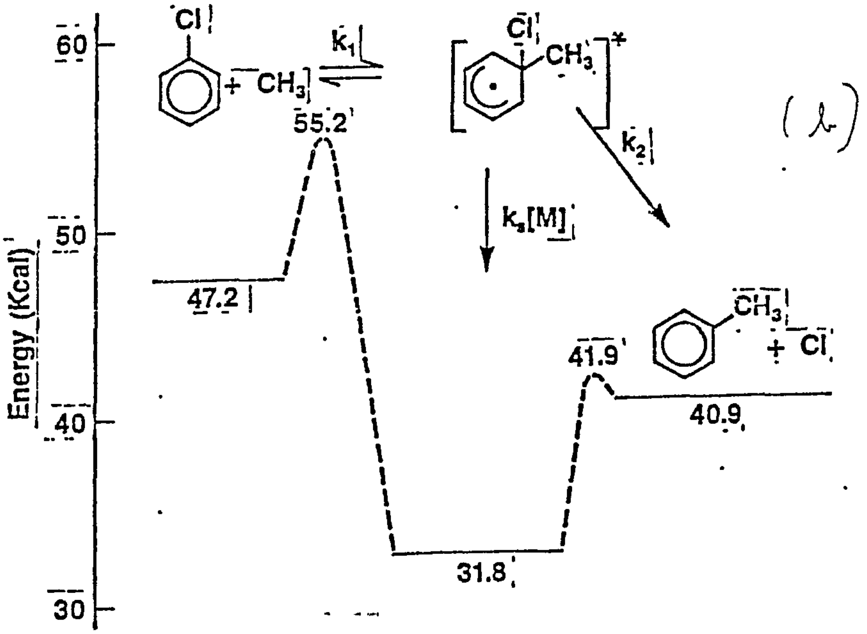


Figure 1.13

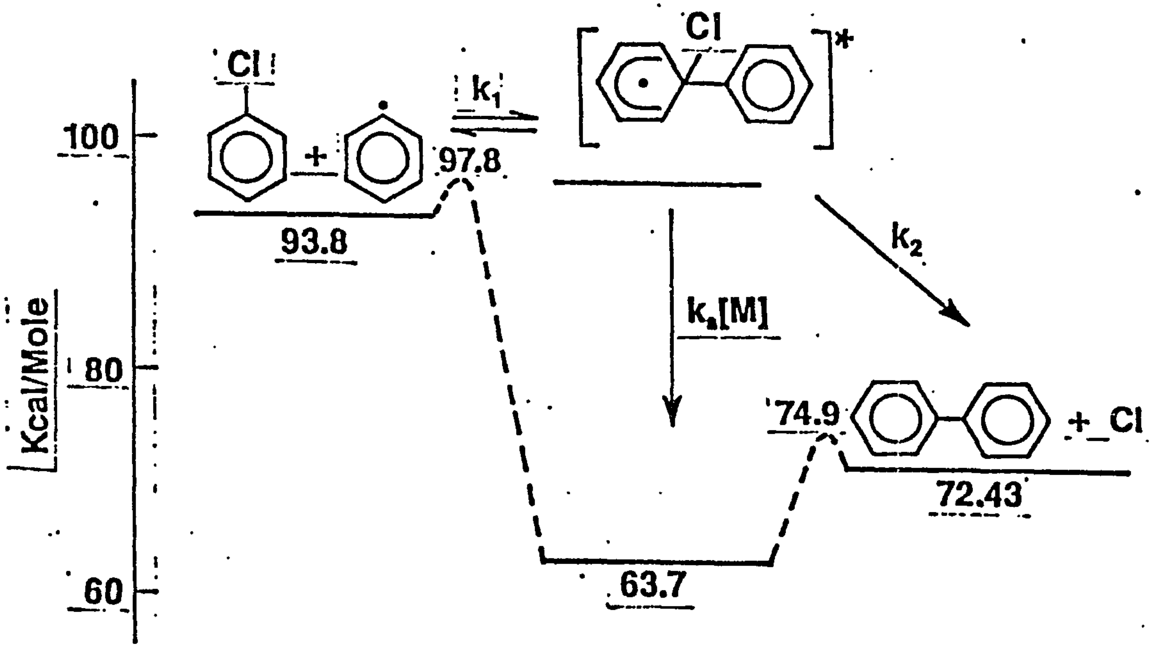
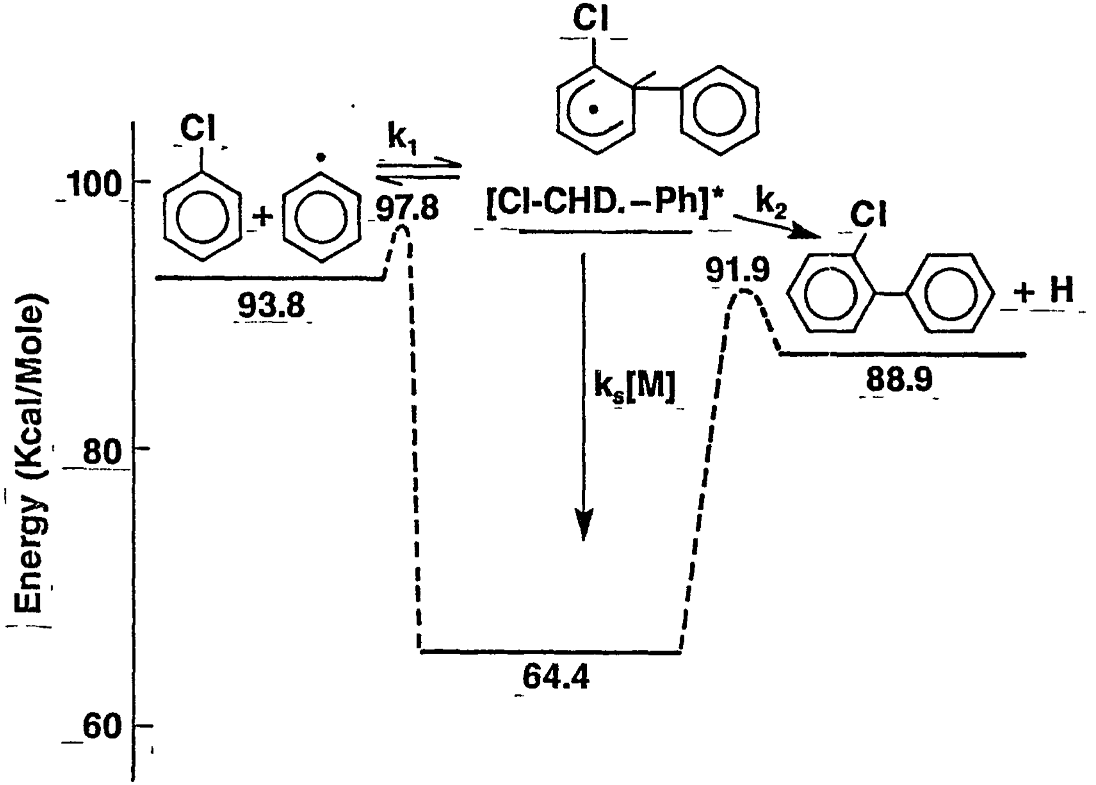


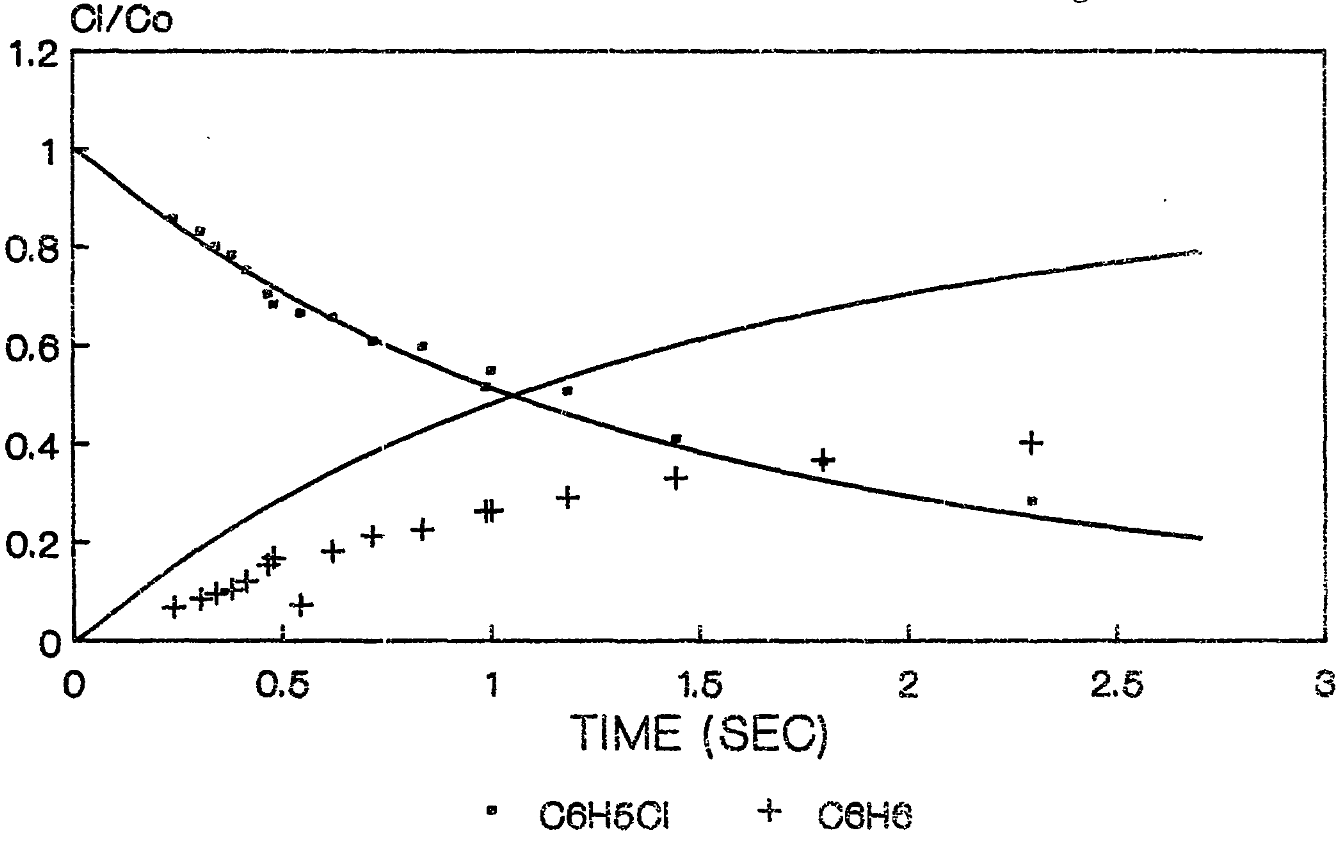
Figure 1.14



ER&E
GRAPHIC DESIGN
0388-75-031

PREDICTED VS. OBSERVED BEHAVIOR 1126 K C6H5Cl DECOMPOSITION IN H2

Figure 1.15



PREDICTED VS. OBSERVED BEHAVIOR 1177 K
C6H5Cl DECOMPOSITION IN H2

Figure 1.16

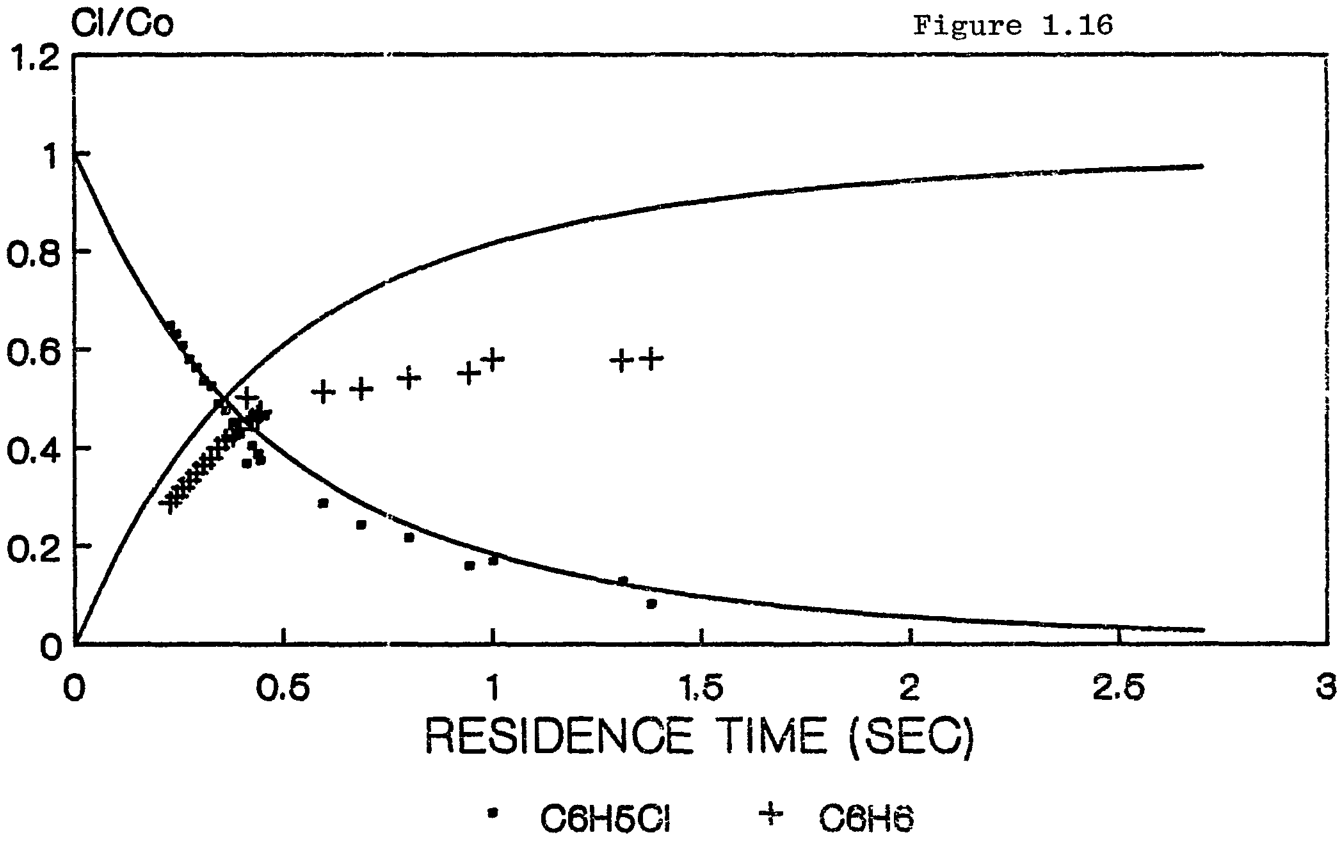


Figure 1.17

DETAILED KINETIC MODEL ACCOUNTS FOR OBSERVED DECAY OF OF CHLOROBENZENE

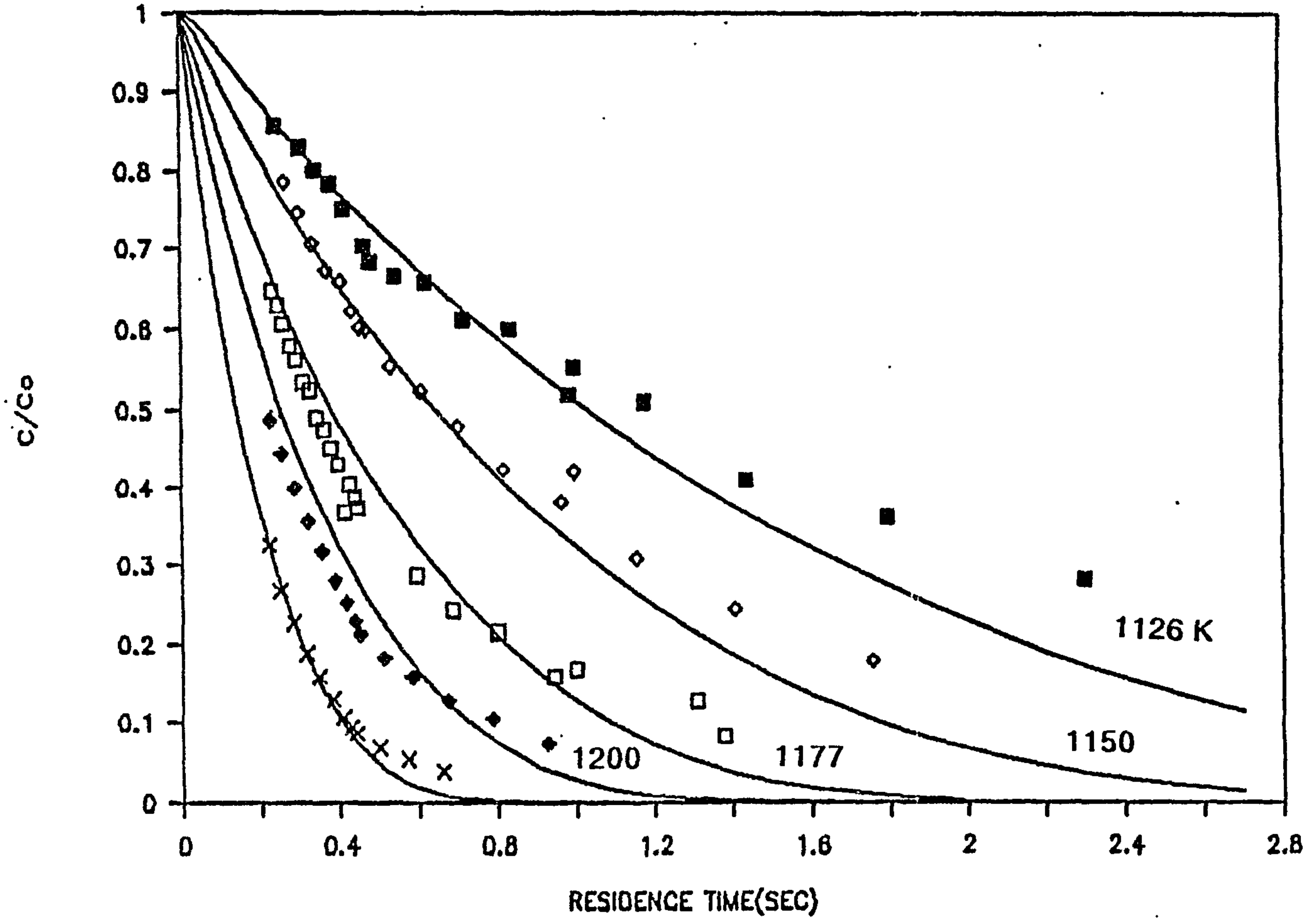
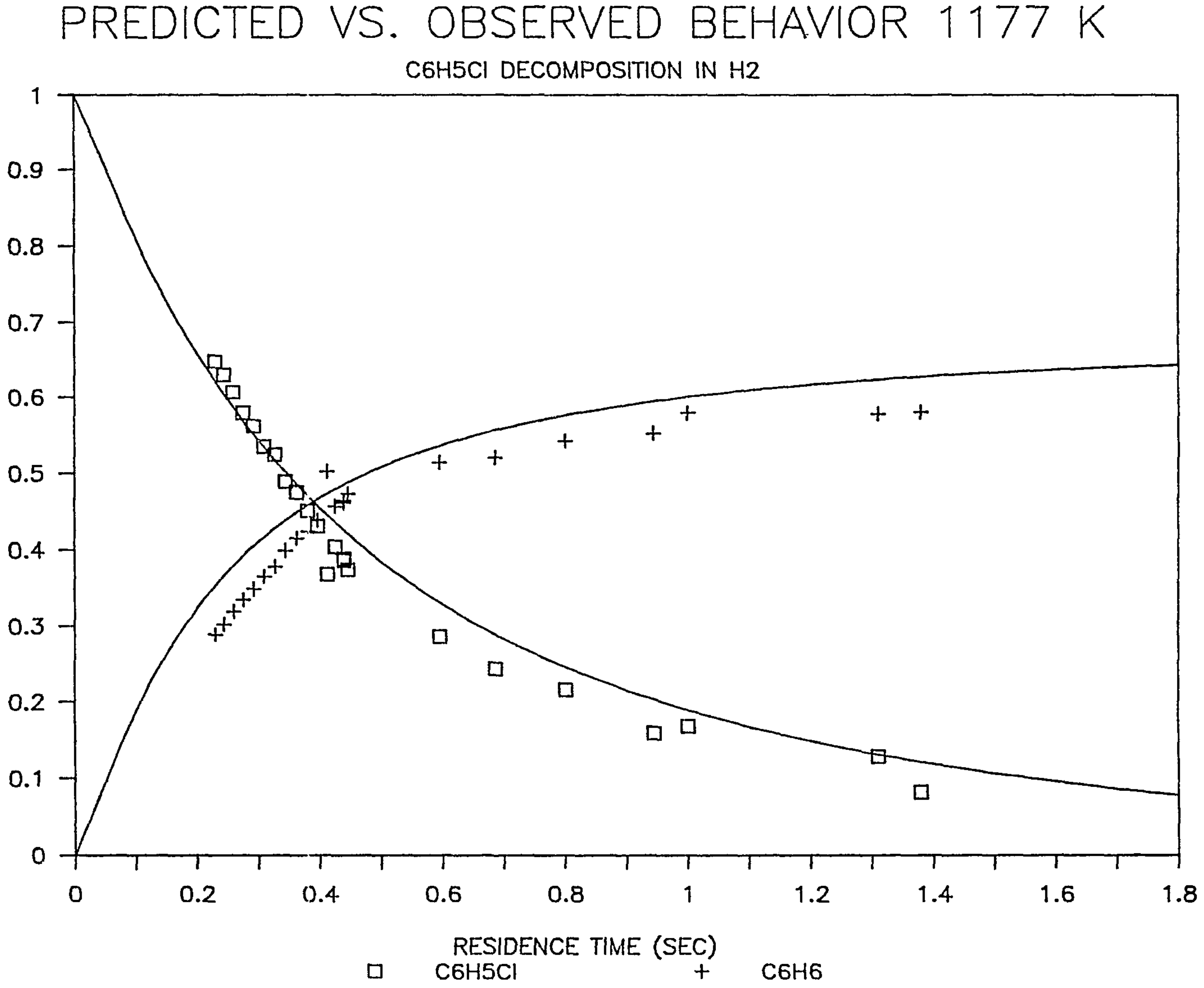


Figure 1.18


PREDICTED VS. OBSERVED BEHAVIOR 1177 K
C6H5Cl DECOMPOSITION IN H2
Ci/Co
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
RESIDENCE TIME (SEC)
□ C6H5Cl
+ C6H6

APPENDIX I - C

THERMO DATA FOR MECHANISM

SPECIES	HF(298)	S(298)	CP300	CP500	CP800	CP1000	CP1500	CP2000	FORMULA
CH3	34.82	46.38	9.26	10.81	12.90	14.09	16.26	17.56	CH3
CH4	-17.90	44.49	8.49	11.15	14.98	17.13	20.57	22.61	CH4
C2H2	54.19	48.00	10.60	13.03	15.29	16.30	18.34	19.57	C2H2
MECY24PD1	48.10	75.50	23.11	36.37	49.85	55.84	64.70	68.39	C6H7
BICYC6H7	71.86	71.52	27.62	34.98	48.20	55.52	63.32	67.20	C6H7
H	52.11	27.39	4.97	4.97	4.97	4.97	4.97	4.97	H
CYC6H5	81.37	69.20	18.79	30.52	41.70	46.37	53.18	56.21	C6H5
CYC6H6	19.82	64.34	19.55	33.07	48.14	54.77	62.27	67.35	C6H6
CYC6H7	49.86	72.01	20.79	35.11	48.74	54.33	62.34	67.34	C6H7
CHD14	26.16	70.80	22.63	37.47	52.04	58.27	67.47	72.45	C6H8
CHD	25.40	72.49	22.67	37.73	52.30	58.45	67.56	72.55	C6H8
CY13PD5	54.30	62.40	17.24	29.40	39.46	43.13	49.10	53.20	C5H5
CY13PD	32.40	64.20	17.64	30.01	41.29	45.81	52.80	57.30	C5H6
MECY14PD3	48.06	75.36	23.11	36.37	49.85	55.84	64.70	68.39	C6H7
MECY24PD	28.30	73.00	23.03	36.74	51.51	58.23	67.83	73.50	C6H8
NAPTH	36.10	80.21	31.89	52.13	70.77	78.43	90.21	94.66	C10H8
C*CC*C	26.33	66.62	19.22	28.36	36.95	40.57	46.34	49.44	C4H6
C#CC*C.	126.27	69.60	17.50	22.98	27.91	29.93	33.27	34.91	C4H3
ME.CY24PD	75.10	76.42	23.11	36.36	49.64	55.53	64.71	69.20	C6H7
C6H5CH3	11.95	76.41	24.88	40.53	55.73	62.22	71.78	75.20	C7H8
H2	0.00	31.21	6.90	6.99	7.10	7.21	7.72	8.17	H2
CL	28.90	39.47	5.20	5.40	5.35	5.30	5.20	3.14	CL
CL2	-0.01	53.29	8.10	8.59	8.91	8.99	9.10	9.16	CL2
C6H5CL	12.35	74.79	23.27	36.47	47.96	52.41	59.17	61.16	C6H5CL
PHCLPHCL	28.72	112.76	44.07	66.94	86.58	93.99	104.92	108.68	C12H8CL2
C6H5C6H5	43.50	93.85	35.74	59.44	81.20	90.05	103.37	108.73	C12H10
PHPHCL	35.82	105.39	39.90	63.17	83.88	92.03	104.14	108.25	C12H9CL
HCL	-22.06	44.64	6.96	6.99	7.29	7.56	8.10	8.40	HCL
C6H4CL	72.45	77.78	22.92	34.33	45.16	49.86	57.75	61.39	C6H4CL
LINC6H5	136.55	80.70	25.50	36.41	45.21	48.39	53.56	55.10	C6H5
PHCYC6H4	104.47	98.02	38.20	62.11	83.59	92.10	104.90	110.93	C12H9
NAPTH	36.00	79.49	31.92	52.07	70.74	78.27	89.14	94.37	C10H8
BICYC6H7.	68.95	76.97	21.04	35.51	48.62	54.14	63.67	67.32	C6H7
CLCHD.	40.83	82.80	23.45	38.43	51.35	56.29	63.90	67.48	C6H6CL
MECHD.	42.92	79.32	25.76	43.62	59.87	66.44	76.69	82.15	C7H9
ICLMECHD.	31.78	85.62	29.42	47.45	63.00	68.98	78.48	83.27	C7H8CL
PHCHD.	76.05	100.40	38.79	65.90	89.97	99.33	113.26	120.62	C12H11
ICLPHCHD.	64.91	106.70	42.45	69.74	93.11	101.87	115.04	121.75	C12H10CL
CLIPHCHD.	65.36	107.43	42.39	69.22	92.56	101.39	114.35	121.31	C12H10CL

SECTION II

KINETIC STUDY ON THERMAL DECOMPOSITION OF m-$C_6H_4Cl_2$ DILUTED IN H_2

EXPERIMENTAL RESULTS: m-$C_6H_4Cl_2$

The decomposition of m-dichlorobenzene (m-$C_6H_4Cl_2$) in hydrogen at 1 atm. total pressure was observed to follow pseudo-first order kinetics throughout the temperature range studied (1095 - 1225 K). The first order behavior obtained for decomposition at several temperatures is shown in figure 2.1 for reaction in a 4 mm id reactor. Major products were chlorobenzene (C_6H_5Cl), HCl, carbon solids, and benzene (C_6H_6).

Decomposition was found to be most rapid within the 16mm id reactor and slowest within the 4mm id reactor. This trend is the opposite to results obtained for reactions of C_6H_5Cl in H_2 studied at this laboratory under similar conditions (see SECTION I : Results for C_6H_5Cl).<1> It is believed that hydrogen atom loss by surface reaction is responsible for this trend. The production rate of H atoms by reaction of Cl atom with H_2 is believed to be more rapid in this system than in the C_6H_5Cl system for two reasons. There is a higher concentration of Cl atom since m-$C_6H_4Cl_2$ dissociates to chlorophenyl radical and Cl atom more rapidly and there are more sites of chlorine substitution for the rapid displacement of Cl atom by addition of H atom to occur. The bond dissociation energy for loss of Cl from m-$C_6H_4Cl_2$ is lower by about 1.5 Kcal/mol <2> than that for loss of Cl from C_6H_5Cl. It is expected that H atom concentration is higher in this system; and therefore loss of H atom at the reactor wall will have a more pronounced effect. This H atom loss mechanism is believed to be the same for m-$C_6H_4Cl_2$ and C_6H_5Cl systems, but H atoms are believed to be much more important in the dichlorinated system. The importance of this recombination increases to the point where it dominates over other wall processes (activation) and reverses the direction of the experimentally observed wall effect. It is felt that we are observing chain termination at the wall instead of initiation. H atom concentration will be discussed further with kinetic modeling results later in this paper.

This inhibition by the wall was found to become more important as the surface to volume (S/V) ratio was increased, even though the opposite trend was observed with C_6H_5Cl studies (large S/V accelerating decomposition). The conclusions one arrives at are, however, completely consistent in both data sets regarding the magnitude of this wall effect in larger id reactors. The importance of wall reactions is found to drop as S/V is decreased, with 16mm and 10.5 mm id reactors giving approximately the same results, but 4mm id reactors consistently giving different conversion.

Figure 2.2 illustrates the effect of reactor diameter upon the observed decomposition of m-$C_6H_4Cl_2$ at 1148 K. Reactors of 10.5mm and 16mm id are shown to give virtually identical conversion while reaction is significantly slower in the 4mm id reactor.

Figure 2.3 demonstrates the influence S/V has on the observed first order decomposition rate constants. It is apparent that wall effects become very important at temperatures above 1200 K for 4mm id reactors. Although large S/V is found to inhibit m-$C_6H_4Cl_2$ decomposition instead of accelerate it, we feel these results are in agreement with trends observed in other studies at this laboratory.<3>

Figure 2.4 presents the normalized concentration of major products (HCl not included) as a function of average residence time, 16 mm id reactor at 1123 K. Reactant concentration decreases steadily reaching 80% conversion by 1.2 sec average residence time. Chlorobenzene concentration increases to a maximum near 35 % after 1 sec and then begins to decrease as it is converted to benzene and solid carbon. Benzene production only rises to 28 % of the m-$C_6H_4Cl_2$ initially present. Nearly 35% of m-$C_6H_4Cl_2$ reagent is converted to solids which accumulated on the reactor tube wall and in exit lines. Carbon solid was estimated from material balance after taking into account the relative response factors of m-$C_6H_4Cl_2$, C_6H_5Cl, and C_6H_6 to detection by flame ionization detector (FID).

Several minor products were detected in samples of reactor gas. Minor products present at less than 0.5 mole percent the initial m-$C_6H_4Cl_2$ consistently include cyclopentadiene, toluene, ethylene, methane, naphthalene, biphenyl, chlorobiphenyl isomers, and isomers of dichlorobiphenyl. These observations are in good agreement with other studies performed at this laboratory <5> and elsewhere<6>.

The distribution of major products formed at 1 second average residence time are presented as functions of temperature in figure 2.5. 98% decomposition of m-$C_6H_4Cl_2$ occurs by about 900°C. Chlorobenzene production rises to a maximum near 30% the initial m-$C_6H_4Cl_2$ near 850°C and then decreases to near 10% by about 900°C. Benzene yield increases steadily as temperature is increased, rising to nearly 45% the initial m-$C_6H_4Cl_2$ near 900°C. Solid carbon accounts for a minimum of 25% and maximum near 40% the initial m-$C_6H_4Cl_2$. Between 850°C and 900°C carbon solid is the major organic product observed for 1 second average residence time.

Methylene chloride extracts were obtained from portions of solid carbon deposits and analyzed by GC/Mass Spectrometry. Spectra were obtained for several compounds in the molecular weight range 154 - 380 (see figure 2.6). These compounds are believed to include biphenyl, isomers of chlorobiphenyl and dichlorobiphenyl, in addition to chlorinated terphenyls and triphenylenes.

KINETIC MODELING RESULTS and DISCUSSION:

Bimolecular activated complex/QRRK analysis of Dean<7> is used to account for temperature and pressure effects on apparent rate constants for radical and atom addition reactions to unsaturated bonds. Addition reactions presented here and elsewhere<1,8,9> focus on likely pathways to biphenyls, chlorinated biphenyls, and other species observed in the soot extracts. Required input parameters come from literature or are estimated as noted in the tables that follow. Required geometric mean vibrational frequencies are estimated using the method described in SECTION IV of this thesis and are tabulated in Table II.i along with Lennard-Jones parameters estimated using Lydersen's group method. Due to the large uncertainties in these estimates, L-J parameters for stable molecules were used for the radical adducts. For example, the L-J parameters for chlorophenyl cyclohexadienyl radical were taken to be those estimated for chlorobiphenyl, which differs by only an H atom. This is not expected to cause serious error since QRRK calculations are not sensitive to small changes in L-J parameters.

The initial reaction occurring when m-$C_6H_4Cl_2$ is pyrolyzed is dissociation to m-chlorophenyl radical (m-C_6H_4Cl) and Cl atom. These radicals produced by initial unimolecular decay react rapidly with H_2 (if present) to generate C_6H_5Cl and HCl in addition to reactive H atoms. H atom addition to the Cl-Carbon bond in m-$C_6H_4Cl_2$ (attack at ipso positions) forms an energized complex and results in rapid decomposition of the adduct to C_6H_5Cl + Cl, the low energy exit channel. An energy diagram for this reaction appears in figure 2.7. This reaction is believed to be a major pathway in the hydrodechlorination of chlorinated benzenes and all chlorinated unsaturates under our conditions (H_2 in excess). Virtually every dichloro-cyclohexadienyl radical complex which is formed dissociates by loss of Cl atom before stabilization of the complex can occur at these temperatures. This occurs because the exit barrier is small compared with that for dissociation back to reactants. QRRK calculations show that the rate constant for displacement of Cl by H atom essentially has no pressure dependence even though it is not an elementary reaction. Parameters used in these calculations appear in Table II.ii. The apparent rate constant for reaction to C_6H_5Cl + Cl is simply the rate constant for formation of the adduct, $2.0\text{x}10^{13}\exp(-7{,}500$ Kcal/mol/RT). H atom addition to C_6H_5Cl will produce C_6H_6 + Cl atom passing through an analogous chloro-cyclohexadienyl intermediate.<1>

$$\begin{array}{ll} Cl + H_2 \text{ ---> } HCl + H & \\ \underline{H + m\text{-}C_6H_4Cl_2 \text{ ---> } Cl + C_6H_5Cl} & \\ H_2 + C_6H_5Cl = HCl + C_6H_5Cl & \quad\text{overall reaction} \end{array}$$

Phenyl addition to C_6H_6 will yield some biphenyl product, while addition to C_6H_5Cl will produce both biphenyl and chlorobiphenyl products, passing through a substituted cyclohexadienyl intermediate. These systems have been discussed elsewhere (see SECTION I).

Phenyl and chlorophenyl radical addition to chlorinated benzenes can be divided into two groups; ipso addition (at a position of chlorine substitution) and addition at a Carbon-H bond. Addition at an ipso position will result in rapid reaction to a biphenyl (chlorobiphenyl) + Cl. Fahr and Stein<11> have shown that phenyl migration is much slower than complex dissociation. Therefore, phenyl must add to the site of chlorine substitution for a displacement reaction to occur.

As an example we shall consider m-C_6H_4Cl radical addition to the ipso position of C_6H_5Cl. Dissociation of the chlorophenyl-chloro-cyclohexadienyl radical complex back to m-C_6H_4Cl + C_6H_5Cl is nearly 25 Kcal/mol more endothermic than dissociation to chlorobiphenyl + Cl products. An energy diagram for this reaction is presented in figure 2.8. Despite the size of this adduct (MW 223), stabilization is of little importance even at 1 atm and low temperature (300 K) see figure 2.9. This reaction is analogous to that of H addition to a position of chlorine substitution in that the reaction to products essentially proceeds at the rate of formation of the complex. Figure 2.9 shows the pressure dependence for apparent rate constants in this system at 300 and 2000 K. Reaction to products is found to dominate at all pressures with dissociation to reactants of little importance even at temperatures near

2000 K. Table II.iii summarizes parameters used in these QRRK calculations in addition to the apparent rate constants calculated as functions of temperature for reaction at 1 atm in H_2 bath gas.

Addition of m-C_6H_4Cl to a site of chlorine substitution results in rapid reaction since the energetics of this system favor the displacement reaction products over dissociation back to reactants. This is not the case, however, for addition to a carbon-H bond. As an example we shall examine m-C_6H_4Cl radical addition to C_6H_5Cl (non ipso). In this case, dissociation to products is only about 6 Kcal/mol less endothermic than that for dissociation to reactants (see figure 2.10). As one might expect, stabilization and dissociation to reactants (no reaction) are significantly more important in this case. Such a system is expected to have unusual pressure and temperature dependence for apparent rate constants. Figure 2.11 shows the non-Arrhenius behavior of apparent rate constants for this system at 1 atm pressure. Reaction to products is shown to drop sharply at low temperatures. This is due in part to the increase in stabilization of the adduct. The rate constant for stabilization is found to dominate at low temperatures, pass through a maximum near 800 K, and then to decrease at higher temperatures as reaction to products and dissociation back to reactants begin to dominate.

The Arrhenius like behavior for the displacement reaction of m-C_6H_4Cl plus C_6H_5Cl (discussed earlier) is also shown for comparison. The relative barrier height for inlet and exit channels is shown to have dramatic effect upon the temperature dependence for these addition reactions to aromatics. In a system where the barrier for dissociation to products is significantly lower than for dissociation back to reactants, such as with ipso attack, Arrhenius like behavior is expected. On the other hand, when the barrier for dissociation to products is similar to that for dissociation back to reactants, such as addition at a carbon-H bond, unusual pressure and temperature (non-Arrhenius) effects are observed. Input parameters used in these QRRK calculations in addition to apparent rate constants calculated for H_2 bath gas are presented in Tables II.iii and II.iv.

Another important route to dichlorobiphenyl formation during dichlorobenzene pyrolysis is ipso m-C_6H_4Cl addition to m-$C_6H_4Cl_2$. An energy diagram is presented for this reaction in figure 2.12. This system is analogous to the cases discussed above where a low energy exit channel is available. Reaction to products is found to dominate with stabilization and dissociation of the adduct back to reactants of little or no importance. Temperature dependence for reaction to dichlorobiphenyl + Cl at 1 atm total pressure follows:

$$k = 4.1 \times 10^{13}\ T^{-0.4} \exp(-4.82\ \text{Kcal/mol/RT}).$$

Phenyl radical (C_6H_5) may also add to m-$C_6H_4Cl_2$ to form either chlorobiphenyl or dichlorobiphenyl depending upon the position of addition. These reaction are analogous to those discussed above. Rate parameters determined for these addition reactions are presented in tables II.v and II.vi along with input data used for these QRRK calculations. The behavior of these additions are similar to the chlorophenyl additions to C_6H_5Cl discussed previously, with ipso addition resulting in reaction to products for nearly every complex formed while addition at a non-ipso position produces a more complex temperature and pressure dependence.

Terphenyls result from subsequent phenyl and chlorophenyl additions to biphenyl species. These additions are complicated by differing thermodynamic properties for the different positional isomers. Formation of ortho and meta terphenyls, which doubles the number of combinations which need to be examined, will be considered separately.

Figure 2.13 is an energy diagram for m-C_6H_4Cl radical addition to dichlorobiphenyl at a position of chlorine substitution. The energetics of this system favor loss of Cl (the low energy exit channel). Activated complex/QRRK calculations suggest that nearly 100% of collisions which form the energized complex, dissociate by loss of Cl.

Figure 2.14 show the pressure dependence for apparent rate constants obtained for this system at 1000 K. Reaction to products proceeds at the rate of formation of the complex. At 1 atm (760 torr), stabilization accounts for only about 1% of collisions, while at 10 atm it accounts for only 10%. Dissociation of the adduct back to reactants is unimportant.

Figure 2.15 is an energy diagram for m-C_6H_4Cl addition at a position ortho to the phenyl-phenyl bond in chlorobiphenyl. The carbon/carbon bond formed is stronger by about 5 Kcal/mol than the carbon/hydrogen bond in the complex. As a result, some of this complex will dissociate by loss of H atom. The primary route of dissociation, however, is back to reactants as a result of the unfavorable entropy change for the displacement of H atom. Calculations show that greater than 47.5% of collisions forming the complex dissociate back to reactants, 52% are stabilized, and only 0.5% dissociate by loss of H atom to form dichlorinated terphenyl at 1000 K and 1 atm. Addition of m-C_6H_4Cl to a carbon/hydrogen position meta to the phenyl/phenyl bond is slightly more favorable due to the fact that meta-terphenyls are less sterically hindered and therefore more stable than are ortho-terphenyls. The carbon/carbon bond formed from addition to a meta hydrogen is stronger by nearly 7.5 kcal/mol than the carbon/hydrogen bond which must break for reaction to terphenyl + H to occur. This phenylation at a meta hydrogen is expected to be about 2.4 Kcal/mol more exothermic than phenylation at an ortho position due to the hindered internal rotation in an ortho-terphenyl. The entropy change for this reaction is also slightly more favorable (also due to the hindered internal rotation in the ortho-terphenyl).

Twelve of the various combinations for phenyl/chlorophenyl additions to biphenyls have been examined with input parameters and apparent rate constants tabulated in tables II.xv - II.xix. For additions where stabilization is important at our conditions we must also consider the thermal dissociation of the complex to both products and reactants. This is done using unimolecular QRRK theory of Dean et al.<7>

Triphenylenes may result from reaction of the o-terphenyl radical with itself (condensation). The o-terphenyl radicals may be produced by abstraction of an ortho hydrogen by Cl, H, or aryl radical. Figure 2.16 is an energy diagram showing conversion of o-terphenyl radical to triphenylene + H atom. In this case the carbon/carbon bond which is formed from this isomerization is stronger by nearly 15 Kcal/mol than the carbon/hydrogen bond in the complex. Calculations show reaction to triphenylene + H to be the dominant reaction channel with stabilization of little importance above 800 K. The pressure dependence for these rate constants are compared in figure 2.17 demonstrating again that addition rate constants for reactions which have low energy exit channels available (ie, those which are more than 10 kcal/mol exothermic) exhibit little pressure dependence. Two additional examples for chlorinated triphenylene formation have been examined and input parameters and results for these systems are presented in tables II.xx - II.xxii.

We have developed a detailed reaction mechanism for m-$C_6H_4Cl_2$ pyrolysis in hydrogen which appears in Table II.xxiii. Elementary reaction rate parameters for abstraction reactions are based upon literature comparison, thermodynamic estimations and Transition State Theory methods of Benson.<12> Qrrk calculations, as described above and elsewhere<1>, were used to estimate apparent rate parameters for addition and dissociation reactions (1 atm).

Experimental pyrolysis data are compared with model predictions for m-$C_6H_4Cl_2$ reactions at 1095 and 1198 K in figures 2.18 and 2.19. Predictions for m-$C_6H_4Cl_2$ loss match experiment well. Predictions for C_6H_5Cl formation and loss show slightly less agreement, but still match observed trends well. Prediction of benzene formation, on the other hand, is not quite as good except at low conversion. Modeling suggests that the phenyl/chlorophenyl addition reactions discussed earlier cannot explain the rapid solid carbon formation which is observed in this system by themselves.

Figure 2.20 compares predicted and experimental decomposition curves as function of residence time for several temperatures. Table II.xxiii contains only those reactions which are specific to m-$C_6H_4Cl_2$ pyrolysis, but our model contains the subset of reactions used in our separate study of C_6H_5Cl decomposition. These reactions have been discussed elsewhere (see SECTION I : Results for C_6H_5Cl). Our model is a first principles description of the m-$C_6H_4Cl_2$ chain decomposition, and we feel predictions shown are quite acceptable.

Rate parameters for reactions appearing in Table II.xxiii were varied over small intervals to obtain a better fit of our experimental data. Rate constants were required to be consistent with all reactions of the C_6H_5Cl mechanism. The A factor for m-$C_6H_4Cl_2$ dissociation to m-C_6H_4Cl radical and chlorine atom (6.0E+15) was set equal to that for C_6H_5Cl decomposition on a per Cl basis.

The high pressure A factor for hydrogen atom addition to m-$C_6H_4Cl_2$ (ipso position) was taken as being between 1.5E+13 and 3.0E+13, with an Ea the same as that determined for H atom addition to C_6H_5Cl<1>. Abstraction of hydrogen from H_2 by chlorophenyl radical was assumed to have an activation energy (Ea) of 9.0 kcal/mol; the same as that used in our C_6H_5Cl study. These reactions along with the H atom recombination $H + H + M = H_2 + M$ were found to have the greatest influence on concentrations predicted by our model.

Attempts to model carbon formation using an equilibrium approach and irreversible bleed reactions are discussed in SECTION I.

REFERENCES SECTION II

[1]Ritter, E. R., J.W. Bozzelli, and A. M. Dean, Manuscript submitted to J. Phys. Chem., 1989.

[2]Louw, R.; Dijks, J. H. and Mulder, P., J. Chem. Soc. Perkin Trans. II, **1973**, 40, 1635.

[3]Chuang, S. C. and Bozzelli, J., Environ. Sci. Technol., **1986**, 20, 568.

[4]Chang, S. H. and Bozzelli, J. W. AICHE Journal, **1987**, 33, 1207.

[5]Ritter, E. R., Master's Thesis, NJIT, 1986.

[6]Manion, J.; Mulder, P. and Louw, R., Environ. Sci. Technol., **1985**, 19, 280.

[7]Dean, A. M.,J. Phys. Chem. **1985**, 89, 4600.

[8]Ritter, E. R., J. W. Bozzelli, and A. M. Dean, Fall Technical Meeting Eastern States Section, The Combustion Institute, NBS Gaithersburg, Md., 1987.

[9]Ritter, E. R., J. W. Bozzelli, and M.Hung, Fall Technical Meeting, Eastern States Section, The Combustion Institute, Clearwater Beach, Fl., 1988.

[10] Manion, J. A.,Dijks,J H M., Mulder, P. and Louw, R., Recl. Trav. Chim. Pays-Bas **1988**,107 434-39.

[11] Stein, S. and Fahr, A., J. Phys. Chem. **1988**, 92, 4951.

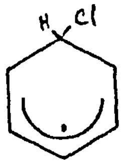
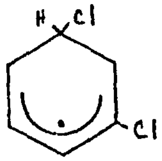
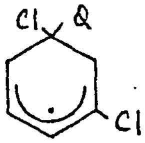
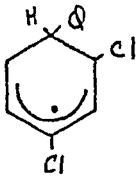
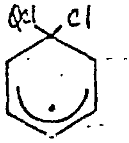
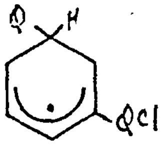
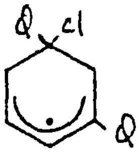
[12]Benson, S., Thermochemical Kinetics (New York: John Wiley & Sons, (**1976**).

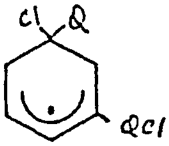
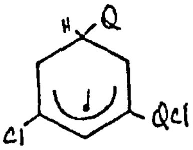
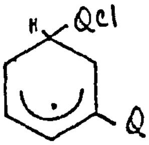
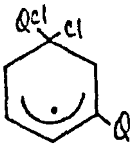
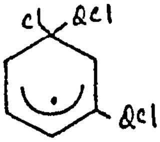
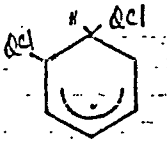
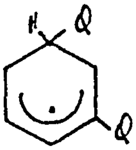
APPENDIX II-A

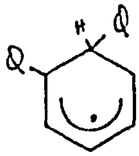
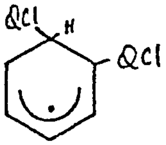
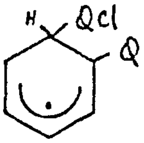
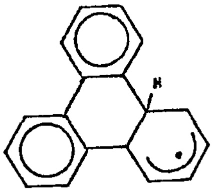
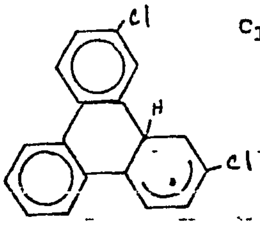
TABLES for SECTION II

TABLE II.i

PROPERTIES FOR RADICAL ADDUCTS CONSIDERED FOR QRRK ANALYSIS

NAME	STRUCTURE	FORMULA MW	ν_{gm} (cm^{-1})	σ (A)	e/k (K)
CLCHD.		C_6H_6Cl (113)	1110	5.69	526.9
CL2CHD.		$C_6H_5Cl_2$ (147)	1000	6.10	571.5
BPH1R		$C_{12}H_9Cl_2$ (223)	975	7.01	658.9
BPH2R		$C_{12}H_9Cl_2$ (223)	977	"	"
BPH3R		$C_{12}H_9Cl_2$ (223)	956	"	"
TPH1R		$C_{18}H_{14}Cl$ (265)	925	7.73	760.2
TPH2R		$C_{18}H_{14}Cl$ (265)	980	"	"

NAME	STRUCTURE	FORMULA MW	w_{gm} (cm^{-1})	σ (A)	e/k (K)
TPH3R		$C_{18}H_{13}Cl_2$ (299)	936	7.91	781.2
TPH4R		$C_{18}H_{13}Cl_2$ (299)	950	"	"
TPH5R		$C_{18}H_{14}Cl$ (265)	925	7.73	760.2
TPH6R		$C_{18}H_{13}Cl_2$ (299)	936	7.91	781.2
TPH7R		$C_{18}H_{12}Cl_3$ (333)	893	8.10	803.4
TPH8R		$C_{18}H_{13}Cl_2$ (299)	866	7.91	781.2
TPH9R		$C_{18}H_{15}$ (231)	985	7.53	731.7

NAME	STRUCTURE	FORMULA	w_{gm}	σ	e/k
		MW	(cm^{-1})	(Å)	(K)
TPH10R		$C_{18}H_{15}$ (231)	985	7.53	731.7
TPH11R		$C_{18}H_{13}Cl_2$ (299)	866	7.91	781.2
TPH12R		$C_{18}H_{14}Cl$ (265)	925	7.73	760.2
TPHF1R		$C_{18}H_{13}$ (229)	1040	7.68	738.8
TPHF2R		$C_{18}H_{11}Cl_2$ (297)	930	8.37	748.3

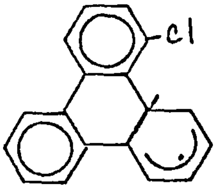
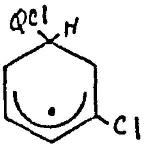
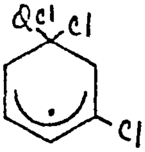
NAME	STRUCTURE	FORMULA	w (cm^{-1})	σ (A)	e/k (K)
TPHF3R		$C_{18}H_{12}Cl$ (262)	985	7.86	744.6
CLIPCLCD.		$C_{12}H_9Cl_2$ (223)	967	7.01	658.9
R46CLIPCL		$C_{12}H_8Cl_3$ (257)	901	7.23	698.4

TABLE II.ii

H + PHCL2 = [CL2CHDR]$^{\#}$ = PRODUCTS

k	A*	E_a**	source
1	2.0 E+13	7.5	a
-1	1.19E+13	34.86	a
2	7.9 E+13	12.2	b

[a] This work, reverse reaction (k-1) from thermodynamics.

[b] Based upon entropy change for Cl + C6H5Cl = [CL2CHDR] with A_{-2} = 1.2E+13 cc/mol sec. E_{-2} = 2kcal/mol.

UNITS:

* A_1 (cc/mol sec)
A_2 (1/sec)

**Ea (kcal/mol)

APPARENT REACTION RATE CONSTANT PREDICTED

USING BIMOLECULAR QRRK ANALYSIS (H_2 BATH GAS)

P (torr)	REACTION	A factor (cc/mol sec)	n	Ea (kcal/mol)	range
7.6	H+PHCL2 = C6H5CL + CL	2.0E+13	0.0	7.5	i
760.0	"	2.96E+13	-0.05	7.63	"

i valid from 300 - 2000 K
ii " " 800 - 1500 K
iii " " 800 - 2000 K

TABLE II.iii

C6H4CL+C6H5CL = [BPH3R]$^{\#}$ = PRODUCTS

k	A^*	E_a^{**}	source
1	1.0 E+12	4.3	a
-1	2.33E+15	33.14	a
2	2.9 E+14	8.87	b

[a] Louw, R.; Dijks, J.H.; Mulder, P. J. Chem. Soc. Perkin Trans. II, 1973, 40, 1635. Reverse reaction (k-1) from thermodynamics.

[b] Based upon entropy change for Cl + PhPhCl = [BPH3R] with A_{-2} = 6.0E+12 cc/mol sec per site of attack.
E_{-2} = 2 kcal/mol

UNITS:

* A_1 (cc/mol sec)
A_2 (1/sec)

**Ea (kcal/mol)

APPARENT REACTION RATE CONSTANT PREDICTED

USING BIMOLECULAR QRRK ANALYSIS (H_2 BATH GAS)

P (torr)	REACTION	A factor (cc/mol sec)	n	Ea (kcal/mol)	range
7.6	C6H4Cl+C6H5Cl=PhPhCl+Cl	1.5E+12	-0.06	4.36	i
760.0	"	1.78E+12	-0.08	4.41	"

i valid from 300 - 2000 K
ii " " 800 - 1500 K
iii " " 800 - 2000 K

TABLE II.iv

C6H4CL+C6H5CL = [CLIPCLCD.]$^{\#}$ = PRODUCTS

k	A^{*}	E_a^{**}	source
1	2.0 E+12	4.3	a
-1	3.68E+15	31.78	a
2	7.09E+13	25.60	b

[a] Louw, R.; Dijks, J.H.; Mulder, P. J. Chem. Soc. Perkin Trans. II, 1973, 40, 1635. Reverse reaction (k-1) from thermodynamics. 5 sites for attack at H.

[b] Based upon entropy change for H + PhClPhCl = [CLIPCLCD] with A_{-2} = 1.0E+13 cc/mol sec per site of attack and 2 sites for attack. E_{-2} = 4.3 kcal/mol.

UNITS:

* A_1 (cc/mol sec)
A_2 (1/sec)

**Ea (kcal/mol)

APPARENT REACTION RATE CONSTANT PREDICTED

USING BIMOLECULAR QRRK ANALYSIS (H_2 BATH GAS)

P (torr)	REACTION	A factor (cc/mol sec)	n	Ea (kcal/mol)	range
7.6	C6H4Cl+C6H5Cl=PhClPhCl+H	4.4E+20	-2.76	7.47	i
760.0	"	6.79E+60	-14.00	38.67	ii
7.6	C6H4Cl+C6H5Cl=[CLIPCLCD.]^{o}	2.5E+71	-18.99	21.01	"
760.0	"	6.97E+86	-22.39	39.37	"

i valid from 300 - 2000 K
ii " " 800 - 1500 K
iii " " 800 - 2000 K

TABLE II.v

C6H5+PHCL2 = [BPH1R]# = PRODUCTS

k	A*	E_a**	source
1	2.0 E+12	4.3	a
-1	4.9 E+15	37.42	a
2	5.75E+14	12.15	b

[a] Louw, R.; Dijks, J.H.; Mulder, P. J. Chem. Soc. Perkin Trans. II, 1973, 40, 1635. Reverse reaction (k-1) from thermodynamics.

[b] Based upon entropy change for Cl + PhPhCl = [BPH1R] with A_{-2} = 6.0E+12 cc/mol sec per site of attack. E_{-2} = 2 kcal/mol.

UNITS:

* A_1 (cc/mol sec)
A_2 (1/sec)

**Ea (kcal/mol)

APPARENT REACTION RATE CONSTANT PREDICTED

USING BIMOLECULAR QRRK ANALYSIS (H_2 BATH GAS)

P (torr)	REACTION	A factor (cc/mol sec)	n	Ea (kcal/mol)	range
7.6	C6H5+PHCL2 = PhPhCl+Cl	2.85+12	-0.05	4.35	i
760.0	"	4.87E+12	-0.11	4.52	"

i valid from 300 - 2000 K
ii " " 800 - 1500 K
iii " " 800 - 2000 K

TABLE II.vi

C6H5 + PHCL2 = [BPH2R]$^{\#}$ = PRODUCTS

k	A*	E$_a^{**}$	source
1	1.3 E+12	4.3	a
-1	3.06E+15	38.22	a
2	1.11E+14	30.77	b

[a] Louw, R.; Dijks, J.H.; Mulder, P. *J. Chem. Soc. Perkin Trans. II*, 1973, 40, 1635. Reverse reaction (k-1) from thermodynamics. 4 sites for attack at H.

[b] Based upon entropy change for H + PhClPhCl = [BPH2R] with A$_{-2}$ = 1.3E+13 cc/mol sec per site of attack and 1 sites for attack. E$_{-2}$ = 4.3 kcal/mol.

UNITS:

* A$_1$ (cc/mol sec)
A$_2$ (1/sec)

**Ea (kcal/mol)

APPARENT REACTION RATE CONSTANT PREDICTED

USING BIMOLECULAR QRRK ANALYSIS (H$_2$ BATH GAS)

P (torr)	REACTION	A factor (cc/mol sec)	n	Ea (kcal/mol)	range
7.6	C6H5 + PHCL2 =PhClPhCl+H	2.7E+32	-5.95	17.36	iii
760.0	"	2.62E+57	-12.69	42.38	"
7.6	C6H5 + PHCL2 =[BPH2R]^{o}	7.21+86	-23.08	33.49	ii
760.0	"	1.12E+80	-20.10	39.45	iii

i valid from 300 - 2000 K
ii " " 800 - 1500 K
iii " " 800 - 2000 K

TABLE II.vii

C6H4CL+PHCL2 = [R46CLIPCL]# = PRODUCTS

k	A*	E_a**	source
1	2.0 E+12	4.3	a
-1	4.6 E+16	37.84	a
2	5.91E+14	11.90	b

[a] Louw, R.; Dijks, J.H.; Mulder, P. J. Chem. Soc. Perkin Trans. II, 1973, 40, 1635. Reverse reaction (k-1) from thermodynamics. 2 sites for attack.

[b] Based upon entropy change for Cl + PhClPhCl = [R46CLIPCL] with A_{-2} = 6.0E+12 cc/mol sec per site of attack and 2 sites for attack. E_{-2} = 2 kcal/mol.

UNITS:

* A_1 (cc/mol sec)
A_2 (1/sec)

**Ea (kcal/mol)

APPARENT REACTION RATE CONSTANT PREDICTED

USING BIMOLECULAR QRRK ANALYSIS (H_2 BATH GAS)

P (torr)	REACTION	A factor (cc/mol sec)	n	Ea (kcal/mol)	range
7.6	C6H4Cl+PHCL2=PhClPhCl+Cl	2.25+13	-0.33	4.64	i
760.0	"	4.06E+13	-0.40	4.82	"

i	valid	from	300 - 2000 K
ii	"	"	800 - 1500 K
iii	"	"	800 - 2000 K

TABLE II.viii

C6H5 + C6H5C6H5 = $[TPH9R]^{\#}$ = PRODUCTS

k	A^{*}	E_a^{**}	source
1	1.3 E+12	4.3	a
-1	2.10E+15	30.90	a
2	6.42E+13	24.20	b

[a] Louw, R.; Dijks, J.H.; Mulder, P. J. Chem. Soc. Perkin Trans. II, 1973, 40, 1635. Reverse reaction (k-1) from thermodynamics. 4 sites for attack at H.

[b] Based upon entropy change for H + MPH3 = [TPH9R] with A_{-2} = 2.7E+13 cc/mol sec and 4 sites for attack. E_{-2} = 4.3 kcal/mol.

UNITS:

* A_1 (cc/mol sec)
A_2 (1/sec)

**Ea (kcal/mol)

APPARENT REACTION RATE CONSTANT PREDICTED

USING BIMOLECULAR QRRK ANALYSIS (H_2 BATH GAS)

P (torr)	REACTION	A factor (cc/mol sec)	n	Ea (kcal/mol)	range
7.6	C6H5+C6H5C6H5=MPH3 + H	9.34E+23	-3.66	10.16	iii
760.0	"	2.57E+71	-16.91	47.26	ii
7.6	C6H5+C6H5C6H5=$[TPH9R]^{o}$	7.80+65	-17.43	17.69	ii
760.0	"	7.22E+86	-22.37	40.19	iii

i valid from 300 - 2000 K
ii " " 800 - 1500 K
iii " " 800 - 2000 K

TABLE II.ix

C6H5 + C6H5C6H5 = [TPH1OR]$^{\#}$ = PRODUCTS

k	A*	E$_a^{**}$	source
1	1.3 E+12	4.3	a
-1	2.10E+15	30.90	a
2	2.80E+13	26.33	b

[a] Louw, R.; Dijks, J.H.; Mulder, P. *J. Chem. Soc. Perkin Trans. II*, 1973, 40, 1635. Reverse reaction (k-1) from thermodynamics. 4 sites for attack at H.

[b] Based upon entropy change for H + OPH3 = [TPH1OR] with A$_{-2}$ = 2.0E+13 cc/mol sec and 2 sites for attack. E$_{-2}$ = 4.3 kcal/mol.

UNITS:

* A$_1$ (cc/mol sec)
A$_2$ (1/sec)

**Ea (kcal/mol)

APPARENT REACTION RATE CONSTANT PREDICTED

USING BIMOLECULAR QRRK ANALYSIS (H$_2$ BATH GAS)

P (torr)	REACTION	A factor (cc/mol sec)	n	Ea (kcal/mol)	range
7.6	C6H5+C6H5C6H5=OPH3 + H	1.38E+24	-3.84	11.31	iii
760.0	"	3.91E+74	-17.91	51.36	ii
7.6	C6H5+C6H5C6H5=[TPH1OR]^{o}	2.07E+80	-21.55	26.09	ii
760.0	"	7.19E+86	-22.31	40.81	iii

i valid from 300 - 2000 K
ii " " 800 - 1500 K
iii " " 800 - 2000 K

TABLE II.x

C6H5 + PhPhCl = [TPH12R]$^{\#}$ = PRODUCTS

k	A*	E_a**	source
1	6.7 E+11	4.3	a
-1	7.87E+15	30.70	a
2	3.69E+13	26.33	b

[a] Louw, R.; Dijks, J.H.; Mulder, P. J. Chem. Soc. Perkin Trans. II, 1973, 40, 1635. Reverse reaction (k-1) from thermodynamics. 1 site for attack at H.

[b] Based upon entropy change for H + OPH3CL = [TPH12R] with A_{-2} = 1.3E+13 cc/mol sec and 1 sites for attack. E_{-2} = 4.3 kcal/mol.

UNITS:

* A_1 (cc/mol sec)
A_2 (1/sec)

**Ea (kcal/mol)

APPARENT REACTION RATE CONSTANT PREDICTED

USING BIMOLECULAR QRRK ANALYSIS (H_2 BATH GAS)

P (torr)	REACTION	A factor (cc/mol sec)	n	Ea (kcal/mol)	range
7.6	C6H5+PhPhCl=OPH3Cl + H	7.60E+18	-2.56	7.84	i
760.0	"	4.26E+71	-17.34	47.85	ii
7.6	C6H5 + PHPHCL=[TPH12R]^{o}	1.55E+70	-18.93	18.45	ii
760.0	"	7.20E+86	-22.61	38.63	"

i valid from 300 - 2000 K
ii " " 800 - 1500 K
iii " " 800 - 2000 K

TABLE II.xi

$C6H5 + PhPhCl = [TPH1R]^{\#} = PRODUCTS$

k	A^{*}	E_a^{**}	source
1	6.7 E+11	4.3	a
-1	5.28E+15	31.26	a
2	6.27E+13	24.82	b

[a] Louw, R.; Dijks, J.H.; Mulder, P. J. Chem. Soc. Perkin Trans. II, 1973, 40, 1635. Reverse reaction (k-1) from thermodynamics. 1 site for attack at H.

[b] Based upon entropy change for H + MPH3CL = [TPH1R] with A_{-2} = 1.3E+13 cc/mol sec and 1 sites for attack. E_{-2} = 4.3 kcal/mol.

UNITS:

* A_1 (cc/mol sec)
A_2 (1/sec)

**Ea (kcal/mol)

APPARENT REACTION RATE CONSTANT PREDICTED

USING BIMOLECULAR QRRK ANALYSIS (H_2 BATH GAS)

P (torr)	REACTION	A factor (cc/mol sec)	n	Ea (kcal/mol)	range
7.6	C6H5+PhPhCl=MPH3Cl + H	6.48E+21	-3.28	8.24	iii
760.0	"	5.26E+73	-17.82	47.83	ii
7.6	$C6H5 + PHCL2 = [TPH1R]^{o}$	2.68E+68	-18.39	17.88	i
760.0	"	7.23E+86	-22.59	38.89	"

i valid from 300 - 2000 K
ii " " 800 - 1500 K
iii " " 800 - 2000 K

TABLE II.xii

C6H5+PhPhCl = [TPH2R]$^{\#}$ = PRODUCTS

k	A*	E$_a$**	source
1	1.0 E+12	4.3	a
-1	2.25E+16	33.69	a
2	4.93E+14	9.45	b

[a] Louw, R.; Dijks, J.H.; Mulder, P. J. Chem. Soc. Perkin Trans. II, 1973, 40, 1635. Reverse reaction (k-1) from thermodynamics. 1 site for attack.

[b] Based upon entropy change for Cl + MPH3 = [TPH2R] with A$_{-2}$ = 6.0E+12 cc/mol sec per site of attack and 2 sites for attack. E$_{-2}$ = 2 kcal/mol.

UNITS:

* A$_1$ (cc/mol sec)
A$_2$ (1/sec)

**Ea (kcal/mol)

APPARENT REACTION RATE CONSTANT PREDICTED

USING BIMOLECULAR QRRK ANALYSIS (H$_2$ BATH GAS)

P	REACTION	A factor	n	Ea	range
(torr)		(cc/mol sec)		(kcal/mol)	
7.6	C6H5 + PhPhCl = MPH3 + Cl	5.65+12	-0.23	4.54	i
760.0	"	8.78E+12	-0.29	4.68	"

i valid from 300 - 2000 K
ii " " 800 - 1500 K
iii " " 800 - 2000 K

TABLE II.xiii

C6H5+PhClPhCl = [TPH3R]# = PRODUCTS

k	A*	E_a**	source
1	1.5 E+12	4.3	a
-1	2.0 E+16	33.7	a
2	1.77E+15	8.86	b

[a] Louw, R.; Dijks, J.H.; Mulder, P. J. Chem. Soc. Perkin Trans. II, 1973, 40, 1635. Reverse reaction (k-1) from thermodynamics. 2 sites for attack.

[b] Based upon entropy change for Cl + MPH3CL = [TPH3R] with A_{-2} = 6.0E+12 cc/mol sec per site of attack and 1 site for attack. E_{-2} = 2 kcal/mol.

UNITS:

* A_1 (cc/mol sec)
A_2 (1/sec)

**Ea (kcal/mol)

APPARENT REACTION RATE CONSTANT PREDICTED

USING BIMOLECULAR QRRK ANALYSIS (H_2 BATH GAS)

P (torr)	REACTION	A factor (cc/mol sec)	n	Ea (kcal/mol)	range
7.6	C6H5 + PhClPhCl=MPH3Cl+Cl	5.89+12	-0.18	4.49	i
760.0	"	8.21E+12	-0.22	4.60	"

i valid from 300 - 2000 K
ii " " 800 - 1500 K
iii " " 800 - 2000 K

TABLE II.xiv

C6H5+PhClPhCl = [TPH4R]# = PRODUCTS

k	A*	E_a**	source
1	6.7 E+11	4.3	a
-1	8.5 E+15	33.9	a
2	8.47E+13	25.20	b

[a] Louw, R.; Dijks, J.H.; Mulder, P. J. Chem. Soc. Perkin Trans. II, 1973, 40, 1635. Reverse reaction (k-1) from thermodynamics. 1 site for attack.

[b] Based upon entropy change for H + MPH3CL2 = [TPH4R] with A_{-2} = 1.3E+13 cc/mol sec per site of attack and 1 site for attack. E_{-2} = 4.3 kcal/mol.

UNITS:

* A_1 (cc/mol sec)
A_2 (1/sec)

**Ea (kcal/mol)

APPARENT REACTION RATE CONSTANT PREDICTED

USING BIMOLECULAR QRRK ANALYSIS (H_2 BATH GAS)

P (torr)	REACTION	A factor (cc/mol sec)	n	Ea (kcal/mol)	range
7.6	C6H5 + PhClPhCl=MPH3Cl2+H	2.17+27	-4.79	11.52	iii
760.0	"	8.35E+77	-18.94	50.81	ii
7.6	C6H5+PhClPhCl = [TPH4R]°	7.59E+74	-20.16	22.48	"
760.0	"	7.22E+86	-22.51	39.73	"

i valid from 300 - 2000 K
ii " " 800 - 1500 K
iii " " 800 - 2000 K

TABLE II.xv

C6H4Cl+C6H5C6H5 = [TPH5R]# = PRODUCTS

k	A*	E_a**	source
1	1.3 E+12	4.3	a
-1	2.3 E+15	31.32	a
2	6.26E+13	23.90	b

[a] Louw, R.; Dijks, J.H.; Mulder, P. J. Chem. Soc. Perkin Trans. II, 1973, 40, 1635. Reverse reaction (k-1) from thermodynamics. 4 sites for attack.

[b] Based upon entropy change for H + MPH3CL = [TPH5R] with A_{-2} = 1.3E+13 cc/mol sec per site of attack and 1 site for attack. E_{-2} = 4.3 kcal/mol.

UNITS:

* A_1 (cc/mol sec)
A_2 (1/sec)

**Ea (kcal/mol)

APPARENT REACTION RATE CONSTANT PREDICTED

USING BIMOLECULAR QRRK ANALYSIS (H_2 BATH GAS)

P (torr)	REACTION	A factor (cc/mol sec)	n	Ea (kcal/mol)	range
7.6	C6H4Cl+C6H5C6H5=MPH3Cl+H	1.48+25	-3.99	10.83	iii
760.0	"	2.35E+72	-17.17	47.87	ii
7.6	C6H4Cl+C6H5C6H5=[TPH5R]°	1.54E+65	-17.21	17.43	"
760.0	"	7.20E+86	-22.36	40.33	"

i	valid	from	300 - 2000 K
ii	"	"	800 - 1500 K
iii	"	"	800 - 2000 K

TABLE II.xvi

C6H4Cl+C6H5C6H5 = [TPH12R]$^{\#}$ = PRODUCTS

k	A^{*}	E_a^{**}	source
1	1.3 E+12	4.3	a
-1	2.3 E+15	31.32	a
2	3.73E+13	26.33	b

[a] Louw, R.; Dijks, J.H.; Mulder, P. J. Chem. Soc. Perkin Trans. II, 1973, 40, 1635. Reverse reaction (k-1) from thermodynamics. 4 sites for attack.

[b] Based upon entropy change for H + OPH3CL = [TPH12R] with A_{-2} = 1.3E+13 cc/mol sec per site of attack and 1 site for attack. E_{-2} = 4.3 kcal/mol.

UNITS:

* A_1 (cc/mol sec)
A_2 (1/sec)

**Ea (kcal/mol)

APPARENT REACTION RATE CONSTANT PREDICTED

USING BIMOLECULAR QRRK ANALYSIS (H_2 BATH GAS)

P (torr)	REACTION	A factor (cc/mol sec)	n	Ea (kcal/mol)	range
7.6	C6H4Cl+C6H5C6H5=OPH3Cl+H	4.51E+36	-7.42	18.91	ii
760.0	"	1.18E+76	-18.29	52.61	"
7.6	C6H4Cl+C6H5C6H5=[TPH12R]^{o}	1.29E+82	-22.06	27.19	"
760.0	"	7.18E+86	-22.30	40.94	"

i valid from 300 - 2000 K
ii " " 800 - 1500 K
iii " " 800 - 2000 K

TABLE II.xvii

C6H4Cl+PhPhCl = [TPH8R]# = PRODUCTS

k	A*	E_a**	source
1	6.7 E+11	4.3	a
-1	5.77E+15	31.05	a
2	2.84E+13	26.60	b

[a] Louw, R.; Dijks, J.H.; Mulder, P. J. Chem. Soc. Perkin Trans. II, 1973, 40, 1635. Reverse reaction (k-1) from thermodynamics. 2 sites for attack.

[b] Based upon entropy change for H + OPH3CL2= [TPH8R] with A_{-2} = 2.0E+13 cc/mol sec per site of attack and 2 site for attack. E_{-2} = 4.3 kcal/mol.

UNITS:

* A_1 (cc/mol sec)
A_2 (1/sec)

**Ea (kcal/mol)

APPARENT REACTION RATE CONSTANT PREDICTED

USING BIMOLECULAR QRRK ANALYSIS (H_2 BATH GAS)

P (torr)	REACTION	A factor (cc/mol sec)	n	Ea (kcal/mol)	range
7.6	C6H4Cl+PhPhCl =OPH3Cl2+H	6.42E+19	-2.84	8.30	ii
760.0	"	1.06E+74	-18.04	49.54	"
7.6	C6H4Cl+PhPhCl = [TPH8R]°	3.17E+72	-19.58	19.84	"
760.0	"	7.17E+86	-22.59	38.81	"

i valid from 300 - 2000 K
ii " " 800 - 1500 K
iii " " 800 - 2000 K

TABLE II.xviii

C6H4Cl+PhPhCl = [TPH6R]# = PRODUCTS

k	A*	E_a**	source
1	1.0 E+12	4.3	a
-1	3.2 E+16	33.8	a
2	5.3 E+14	8.86	b

[a] Louw, R.; Dijks, J.H.; Mulder, P. J. Chem. Soc. Perkin Trans. II, 1973, 40, 1635. Reverse reaction (k-1) from thermodynamics. 1 site for attack.

[b] Based upon entropy change for Cl + MPH3CL = [TPH6R] with A_{-2} = 6.0E+12 cc/mol sec per site of attack and 1 site for attack. E_{-2} = 2 kcal/mol.

UNITS:

* A_1 (cc/mol sec)
A_2 (1/sec)

**Ea (kcal/mol)

APPARENT REACTION RATE CONSTANT PREDICTED

USING BIMOLECULAR QRRK ANALYSIS (H_2 BATH GAS)

P (torr)	REACTION	A factor (cc/mol sec)	n	Ea (kcal/mol)	range
7.6	C6H4CL+PhPhCl=MPH3Cl+Cl	5.12+12	-0.22	4.53	i
760.0	"	7.12E+12	-0.26	4.63	"

i valid from 300 - 2000 K
ii " " 800 - 1500 K
iii " " 800 - 2000 K .

TABLE II.xix

C6H4Cl+PhClPhCl = [TPH7R]$^{\#}$ = PRODUCTS

k	A*	E_a**	source
1	1.5 E+12	4.3	a
-1	2.2 E+16	34.1	a
2	5.3 E+14	8.86	b

[a] Louw, R.; Dijks, J.H.; Mulder, P. J. Chem. Soc. Perkin Trans. II, 1973, 40, 1635. Reverse reaction (k-1) from thermodynamics. 2 sites for attack.

[b] Based upon entropy change for Cl + MPH3CL2 = [TPH7R] with A_{-2} = 6.0E+12 cc/mol sec per site of attack and 2 sites for attack. E_{-2} = 2 kcal/mol.

UNITS:

* A_1 (cc/mol sec)
A_2 (1/sec)

**Ea (kcal/mol)

APPARENT REACTION RATE CONSTANT PREDICTED

USING BIMOLECULAR QRRK ANALYSIS (H_2 BATH GAS)

P (torr)	REACTION	A factor (cc/mol sec)	n	Ea (kcal/mol)	range
7.6	C6H4CL+PhClPhCl=MPH3Cl2+Cl	5.97+12	-0.19	4.49	i
760.0	"	8.85E+12	-0.23	4.62	"

i valid from 300 - 2000 K
ii " " 800 - 1500 K
iii " " 800 - 2000 K

TABLE II.xx

o-TERPHENYL RADICAL = $[TPHF1R]^{\#}$ = PRODUCTS

k	A^*	E_a^{**}	source
1	1.43E+11	4.3	a
-1	6.0 E+13	23.90	a
2	2.81E+12	8.78	b

[a] Transition State Theory (est at 1000 K).
A_1 = 6E+10 * T * exp(ΔS/R) ; reverse from thermo

[b] Based upon entropy change for H + TRIPHENYLENE= [TPHF1R] with A_{-2} = 4.0E+13 cc/mol sec and 6 sites for attack. E_{-2} = 4.3 kcal/mol.

UNITS:

* A_1 (1/sec)
A_2 (1/sec)

**Ea (kcal/mol)

APPARENT REACTION RATE CONSTANT PREDICTED

USING BIMOLECULAR QRRK ANALYSIS (H_2 BATH GAS)

P (torr)	REACTION	A factor (1/sec)	n	Ea (kcal/mol)	range
7.6	OPH3R = TRIPHENYLENE +H	1.14E+14	-0.90	5.36	i
760.0	"	1.89E+18	-2.06	8.86	"
7.6	OPH3R = $[TPHF1R]^{o}$	3.64E+24	-5.38	5.48	"
760.0	"	3.58E+30	-6.48	8.94	"

i valid from 300 - 2000 K
ii " " 800 - 1500 K
iii " " 800 - 2000 K

TABLE II.xxi

o-PH3CL2 RADICAL = $[TPHF2R]^{\#}$ = PRODUCTS

k	A^{*}	E_a^{**}	source
1	1.80E+11	4.3	a
-1	6.0 E+13	23.90	a
2	4.20E+12	8.78	b

[a] Transition State Theory (est at 1000 K).
A_1 = 6E+10 * T * exp($\triangle$S/R) ; reverse from thermo

[b] Based upon entropy change for H + TRIPHCL2 = [TPHF2R]
with A_{-2} = 2 E+13 cc/mol sec and
2sites for attack. E_{-2} = 4.3 kcal/mol.

UNITS:

* A_1 (1/sec)
A_2 (1/sec)

**Ea (kcal/mol)

APPARENT REACTION RATE CONSTANT PREDICTED

USING BIMOLECULAR QRRK ANALYSIS (H_2 BATH GAS)

P (torr)	REACTION	A factor (1/sec)	n	Ea (kcal/mol)	range
7.6	OPH3CL2R = TRIPHCL2 +H	3.12E+13	-0.69	5.16	i
760.0	"	1.64E+18	-2.00	8.88	"
7.6	OPH3Cl2R = $[TPHF2R]^{o}$	1.45E+23	-4.98	4.72	"
760.0	"	6.20E+29	-6.27	8.46	"

i valid from 300 - 2000 K
ii " " 800 - 1500 K
iii " " 800 - 2000 K

TABLE II.xxii

o-PH3CL RADICAL = [TPHF3R]$^{\#}$ = PRODUCTS

k	A*	E_a**	source
1	1.43E+11	4.3	a
-1	6.0 E+13	23.90	a
2	5.48E+12	8.78	b

[a] Transition State Theory (est at 1000 K).
A_1 = 6E+10 * T * exp(ΔS/R) ; reverse from thermo

[b] Based upon entropy change for H + TRIPHCL = [TPHF3R]
with A_{-2} = 1.3E+13 cc/mol sec and
1 site for attack. E_{-2} = 4.3 kcal/mol.

UNITS:

* A_1 (1/sec)
A_2 (1/sec)

**Ea (kcal/mol)

APPARENT REACTION RATE CONSTANT PREDICTED

USING BIMOLECULAR QRRK ANALYSIS (H_2 BATH GAS)

P (torr)	REACTION	A factor (1/sec)	n	Ea (kcal/mol)	range
7.6	OPH3CLR = TRIPHCL + H	7.70E+12	-0.53	4.96	i
760.0	"	1.47E+17	-1.73	8.30	"
7.6	OPH3ClR = [TPHF3R]^{o}	7.32E+22	-4.94	4.81	"
760.0	"	1.15E+29	-6.10	8.17	"

i valid from 300 - 2000 K
ii " " 800 - 1500 K
iii " " 800 - 2000 K

TABLE II.xxiii

REACTION	A	n	E_a(kcal/mol)	source
1 - 39. see TABLE I.xiv (SECTION I)				
40. PHCL2=C6H4CL+CL	6.0E+15	0.0	93500.0	a
41. PHCL2=C6H3CL2+H	1.0E+16	0.0	108500.0	b
42. PHCL2+H=C6H5CL+CL	2.96E+13	-0.05	7630.0	c
43. CYC6H5+PHCL2=PHCLPHCL+H	1.92E+12	0.0	6740.0	"
44. CYC6H5+PHCL2=PHPHCL+CL	4.87E+12	-0.11	4520.0	"
45. C6H4CL+PHCL2=PHCLPHCL+CL	2.62E+57	-12.69	42380.0	"
46. C6H4CL+C6H5CL=PHPHCL+CL	1.50E+12	-0.06	4360.0	"
47. C6H4CL+C6H5CL=PHCLPHCL+H	6.79E+60	-14.00	38670.0	"
48. C6H4CL+PHCL2=PHCLPHCL+CL	4.06E+13	-0.40	4820.0	"
49. CYC6H5+C6H5C6H5=MPH3+H	2.57E+71	-16.91	47260.0	"
50. CYC6H5+C6H5C6H5=OPH3+H	3.91E+74	-17.91	51360.0	"
51. CYC6H5+PHPHCL=OPH3CL+H	4.26E+71	-17.34	47850.0	"
52. CYC6H5+PHPHCL=MPH3CL+H	5.26E+73	-17.82	47830.0	"
53. CYC6H5+PHPHCL=MPH3+CL	8.78E+12	-0.29	4680.0	"
54. CYC6H5+PHCLPHCL=MPH3CL+CL	8.21E+12	-0.22	4600.0	"
55. CYC6H5+PHCLPHCL=MPH3CL2+H	8.35E+77	-18.94	50810.0	"
56. C6H4CL+C6H5C6H5=MPH3CL+H	2.35E+72	-17.17	47870.0	"
57. C6H4CL+C6H5C6H5=OPH3CL+H	1.18E+76	-18.29	52610.0	"
58. C6H4CL+PHPHCL=OPH3CL2+H	1.06E+74	-18.04	49540.0	"
59. C6H4CL+PHPHCL=MPH3CL+CL	7.12E+12	-0.26	4630.0	"
60. C6H4CL+PHCLPHCL=MPH3CL2+CL	8.85E+12	-0.23	4620.0	"
61. OPH3R=TRIPHENE+H	1.89E+18	-2.06	8860.0	"
62. OPH3CL2R=TRIPHCL2+H	1.64E+18	-2.00	8880.0	"
63. OPH3CLR=TRIPHCL+H	1.47E+17	-1.73	8300.0	"
64. PHCL2+H=C6H3CL2+H2	2.0E+13	0.0	18600.0	d
65. PHCL2+H=C6H4CL+HCL	2.0E+13	0.0	11300.0	"
66. CL+PHCL2=HCL+C6H3CL2	1.0E+13	0.0	12500.0	"

References for Mechanism

a.
Twice rate for C_6H_5Cl dissociation by loss of Cl. See Chlorobenzene mechanism: SECTION I (Table I.xiv.)

b.
A factor 2/3 that for C6H6 dissociation (see section I).

c.
Pressure and temperature dependence accounted for with the CHEMACT computer program (see Tables II.ii-II.xxii). This work.

d.
Optimized reaction. This work.

APPENDIX II-B

FIGURES for SECTION II

Figure 2.1

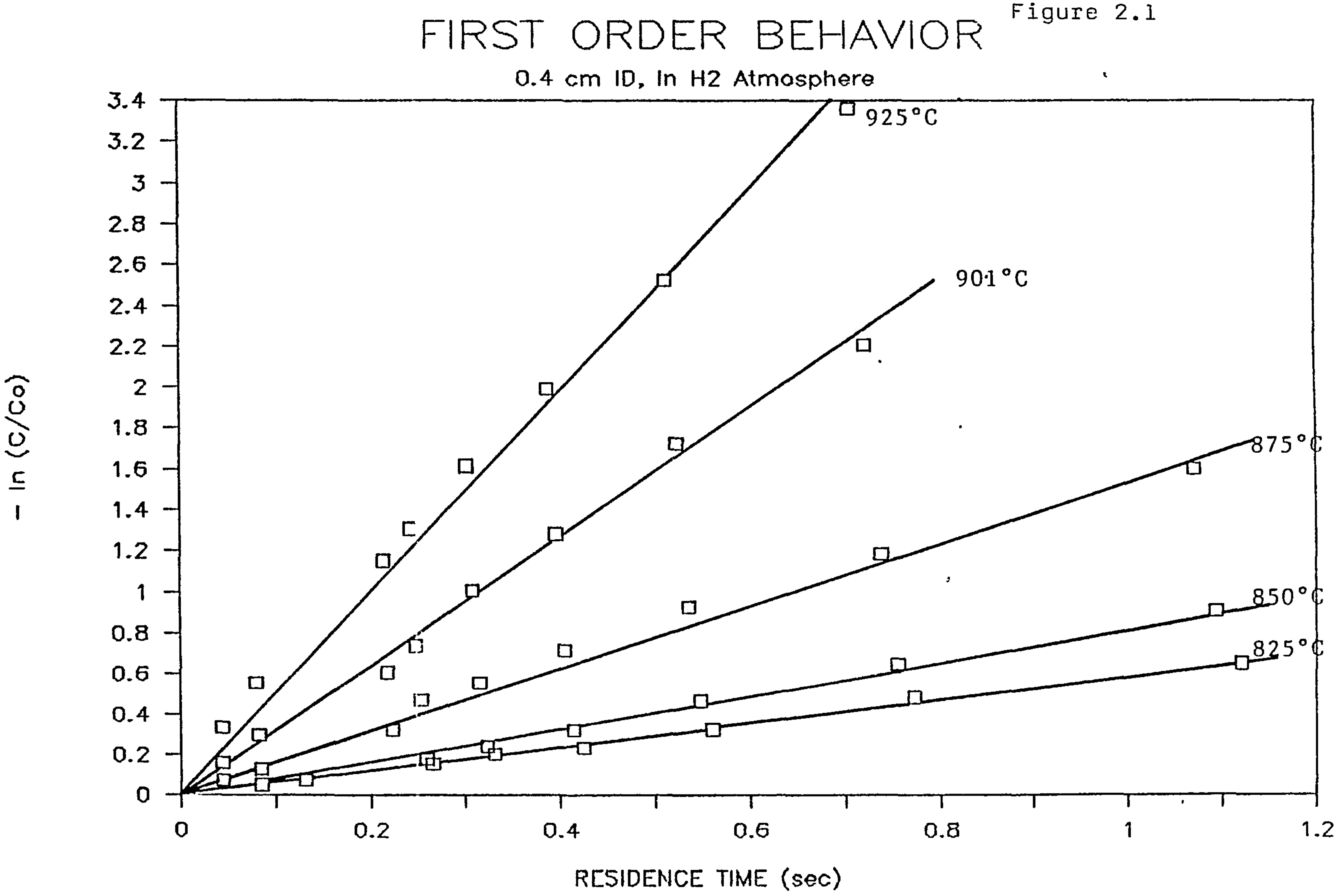

FIRST ORDER BEHAVIOR
0.4 cm ID, In H2 Atmosphere
– ln (C/Co)
3.4
3.2
3
2.8
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
1.2
RESIDENCE TIME (sec)
925°C
901°C
875°C
850°C
825°C

Figure 2.2

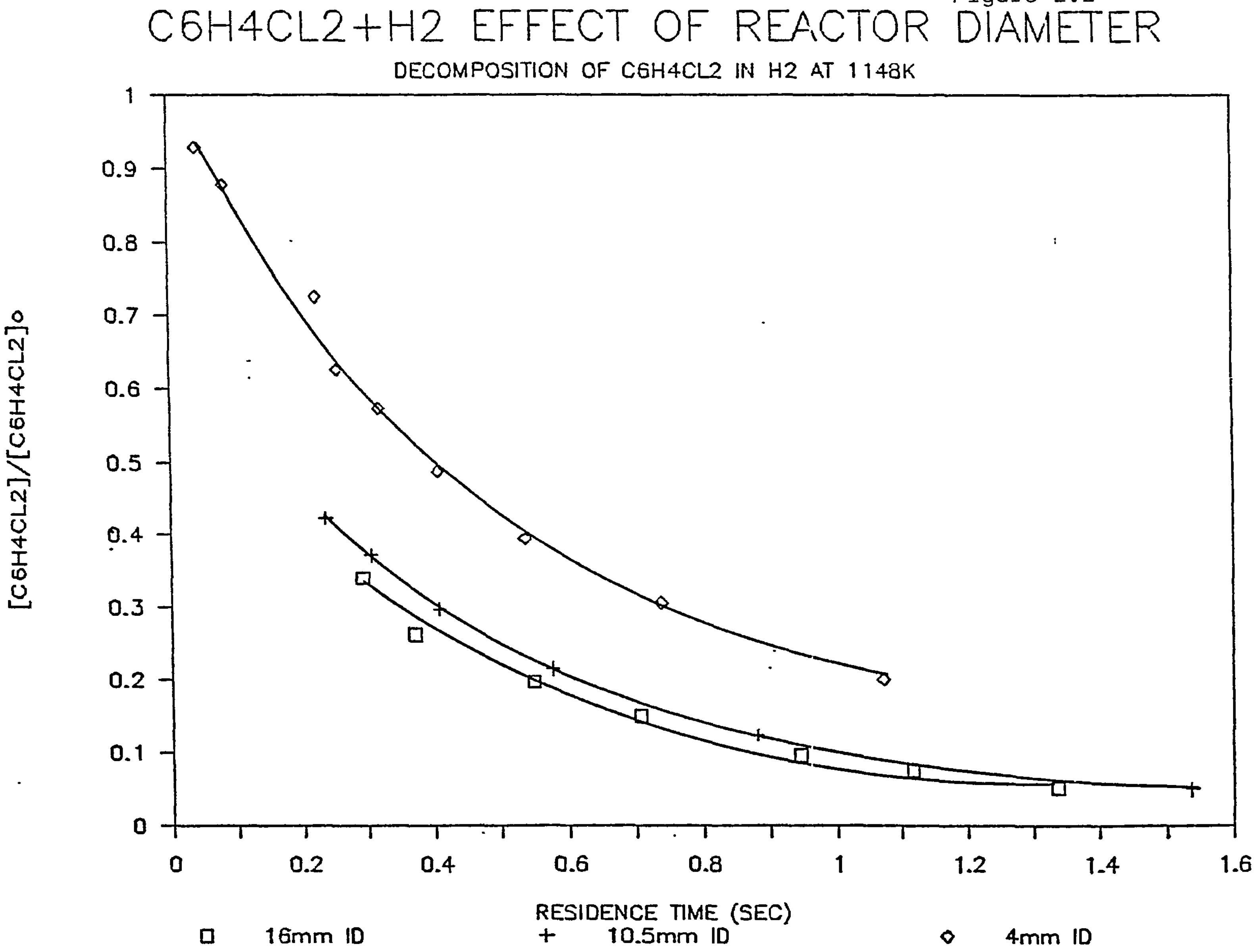
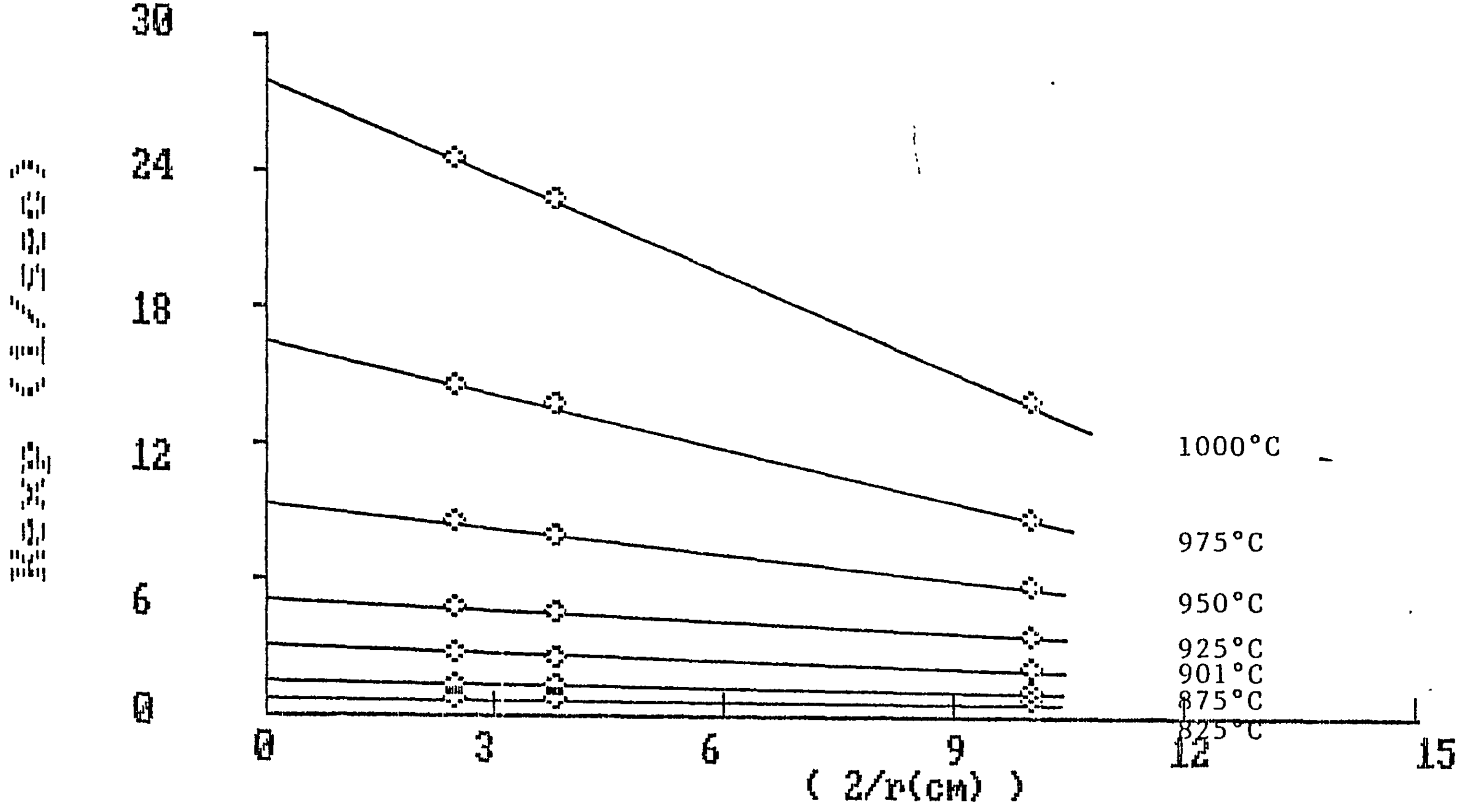

C6H4CL2+H2 EFFECT OF REACTOR DIAMETER
DECOMPOSITION OF C6H4CL2 IN H2 AT 1148K
[C6H4CL2]/[C6H4CL2]o
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
RESIDENCE TIME (SEC)
□ 16mm ID
+ 10.5mm ID
◇ 4mm ID

Figure 2.3

DECOUPLING of Kb and Kw with the Plug Flow Model



m-$C_6H_4Cl_2$ Pyrolysis in H_2 : 1123 K
Experimental Product Distribution

Figure 2.4

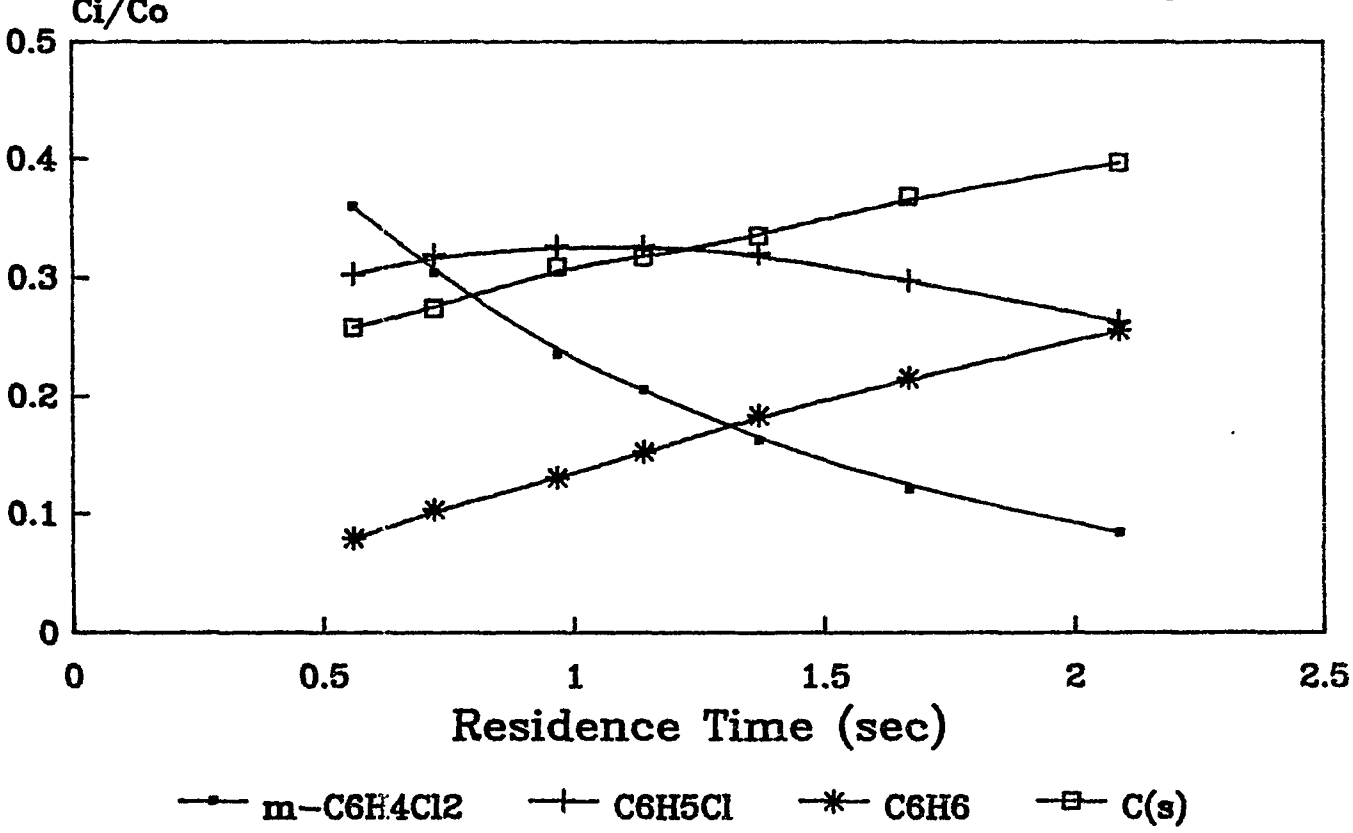


Figure 2.5

1.6 cm ID rxtr : 1 sec residence time

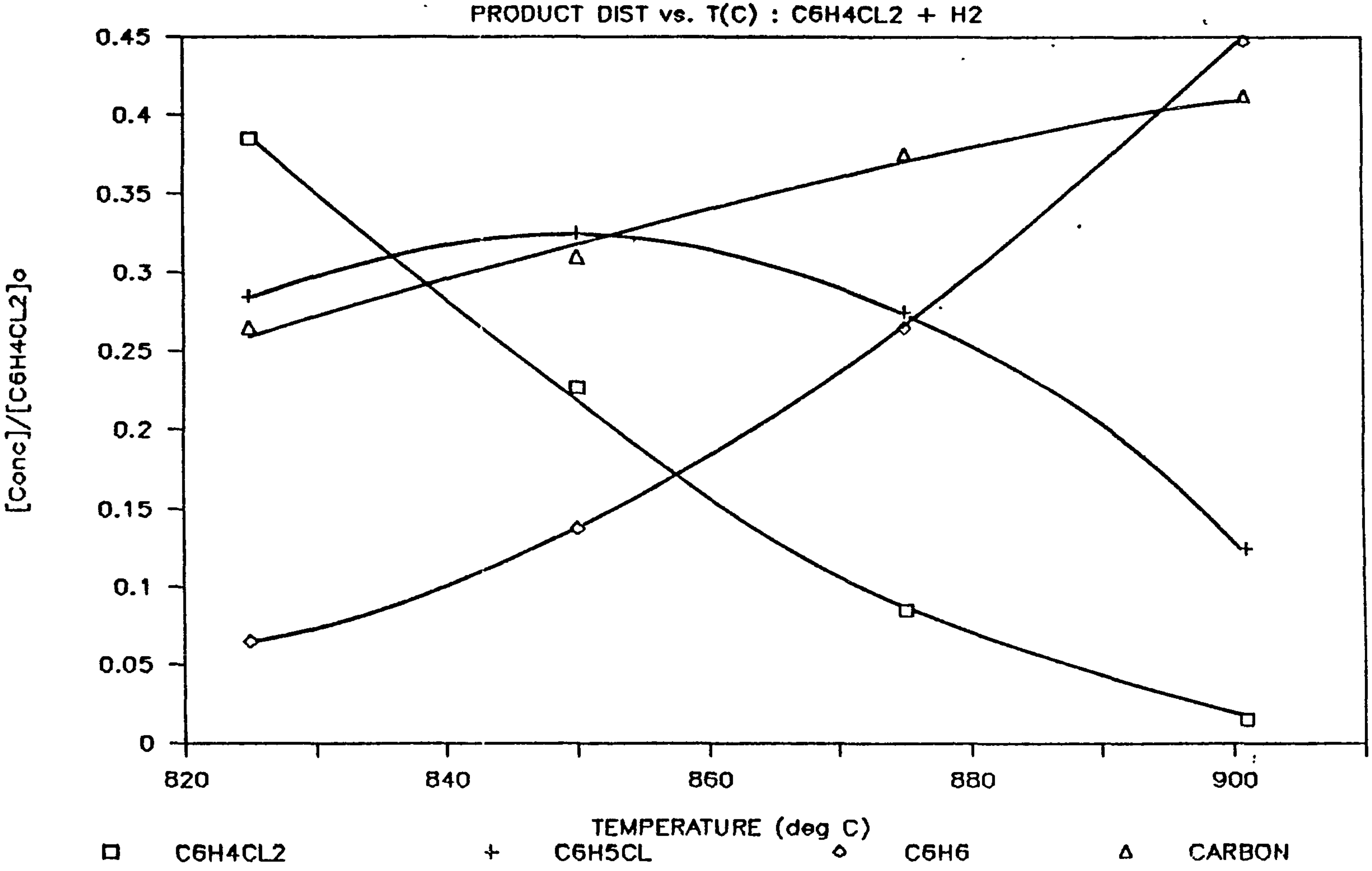
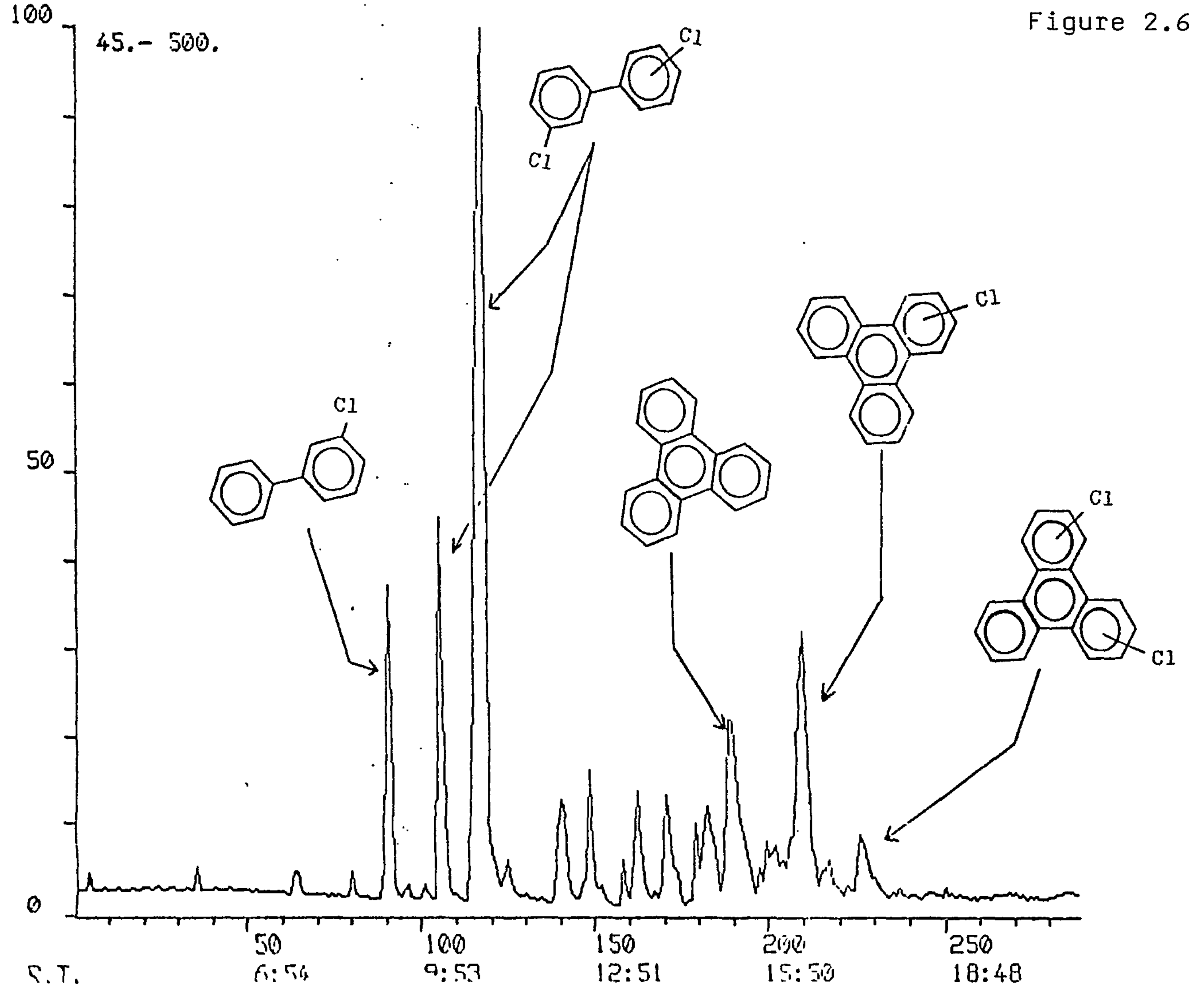
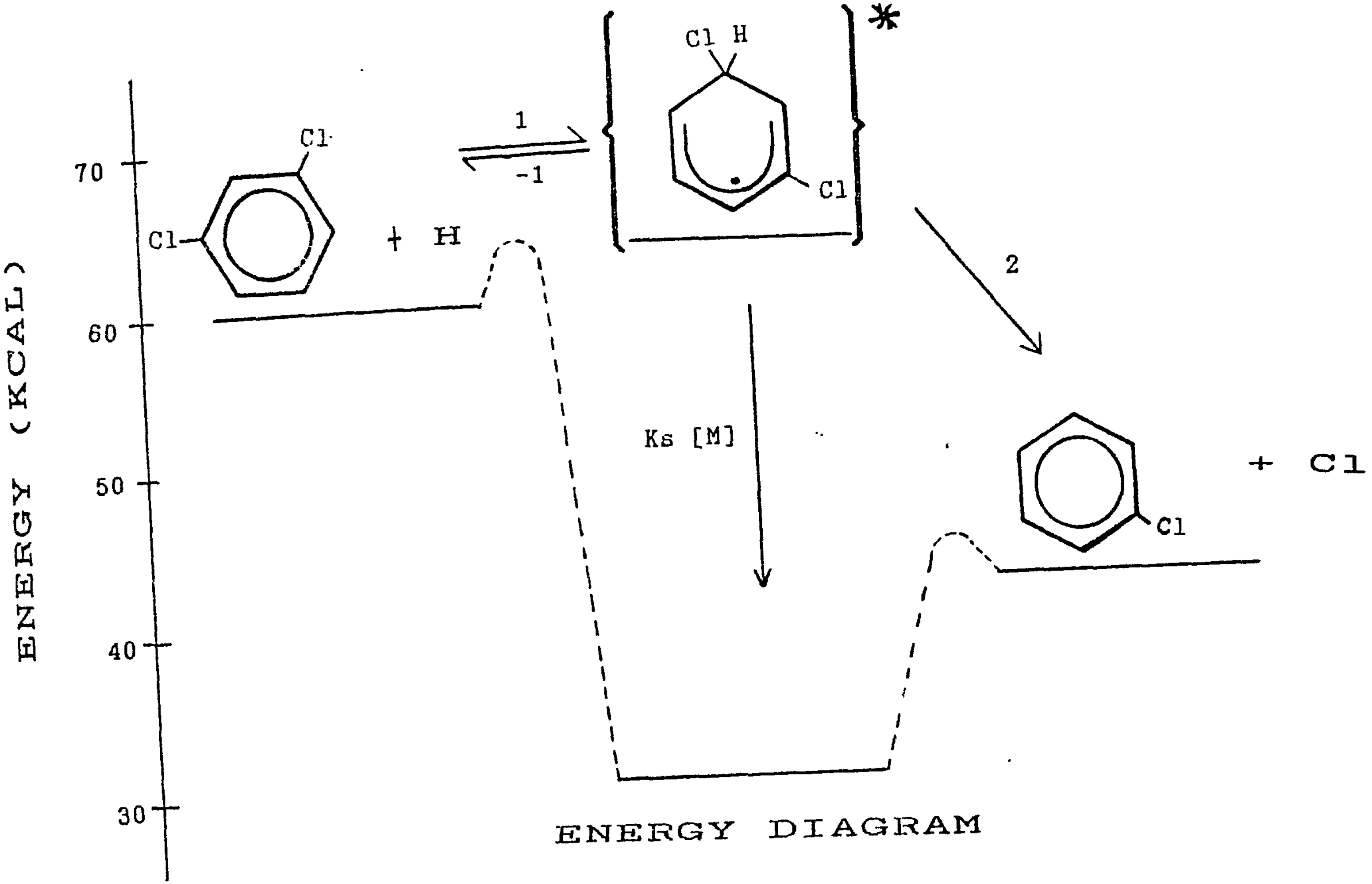


Figure 2.6



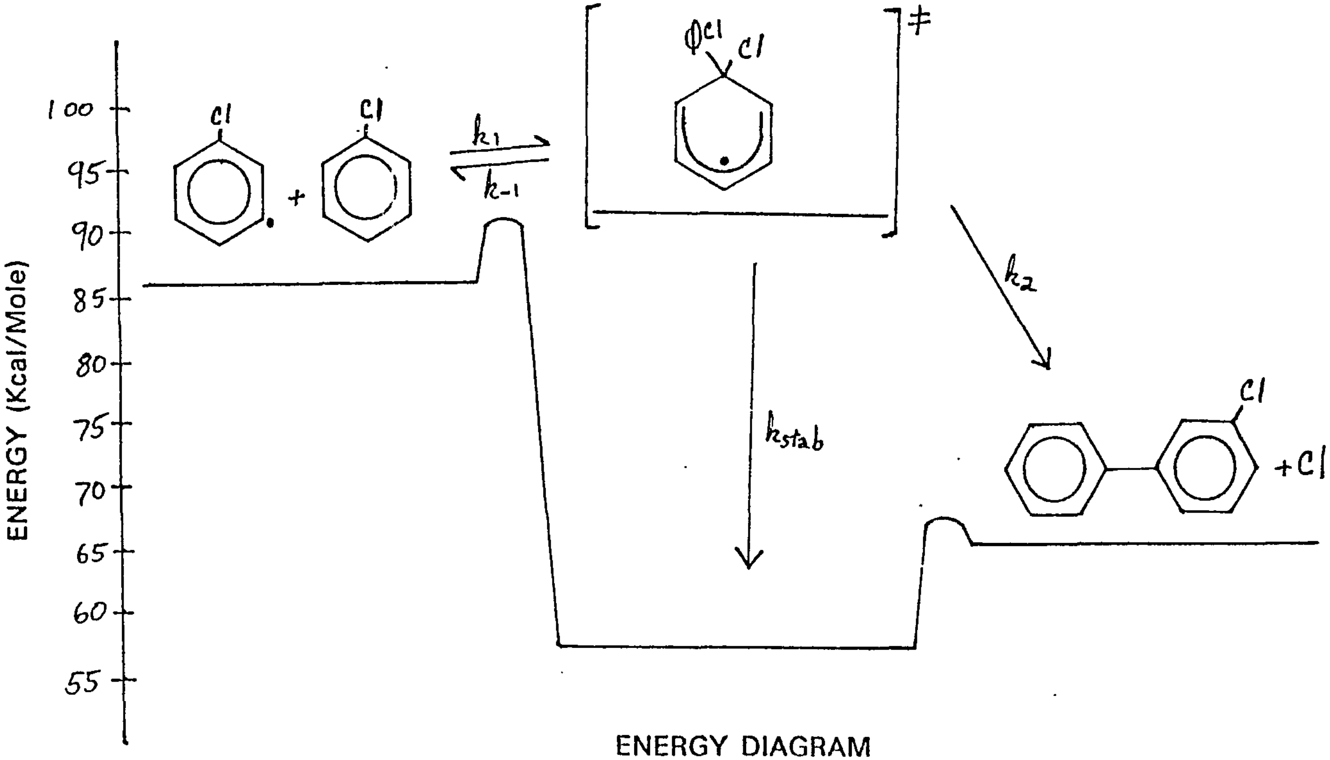
45.- 500.
100
50
0
Cl
Cl
Cl
Cl
Cl
Cl
Cl
S.T.
50
6:54
100
9:53
150
12:51
200
15:50
250
18:48

Figure 2.7



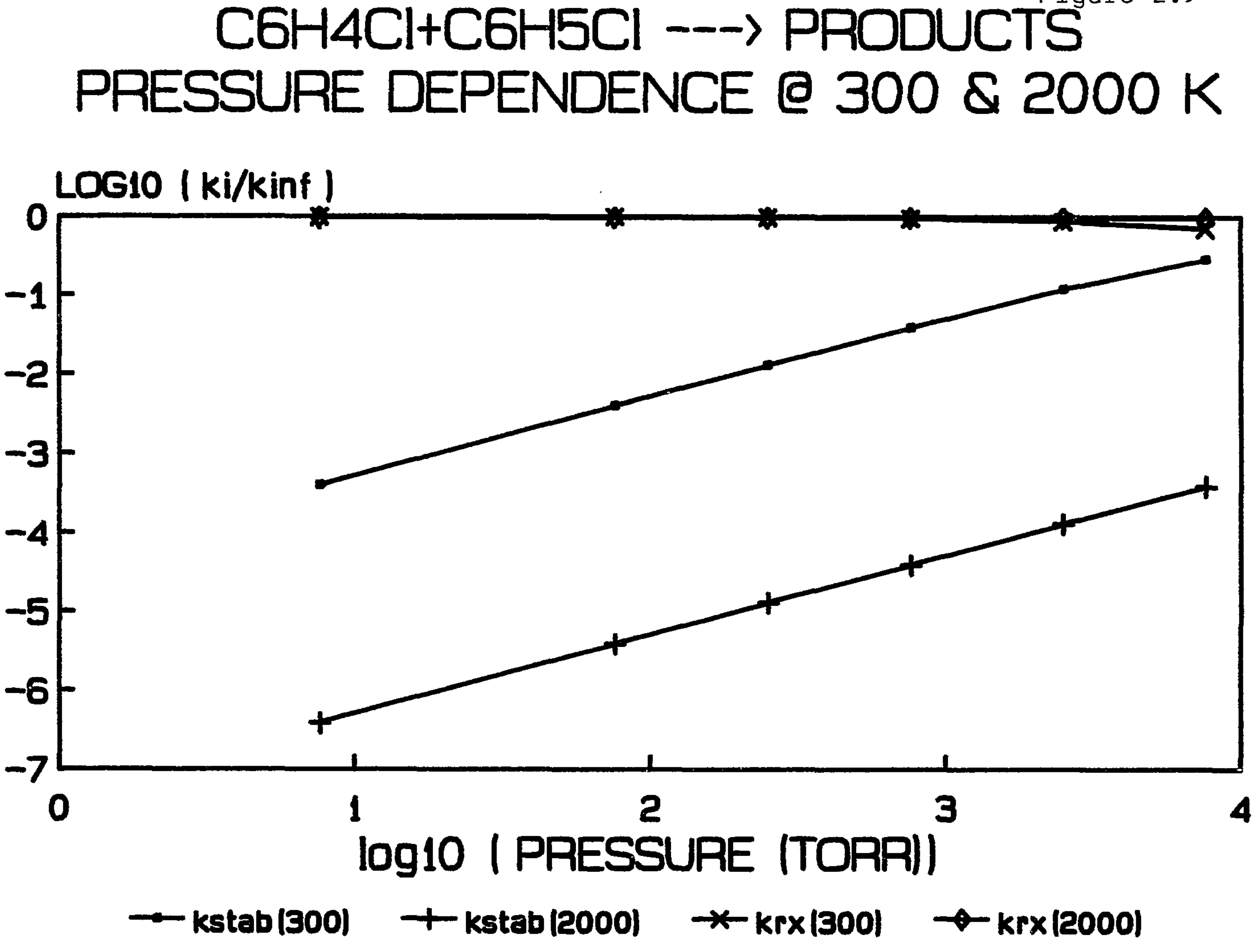
ENERGY DIAGRAM

Figure 2.8



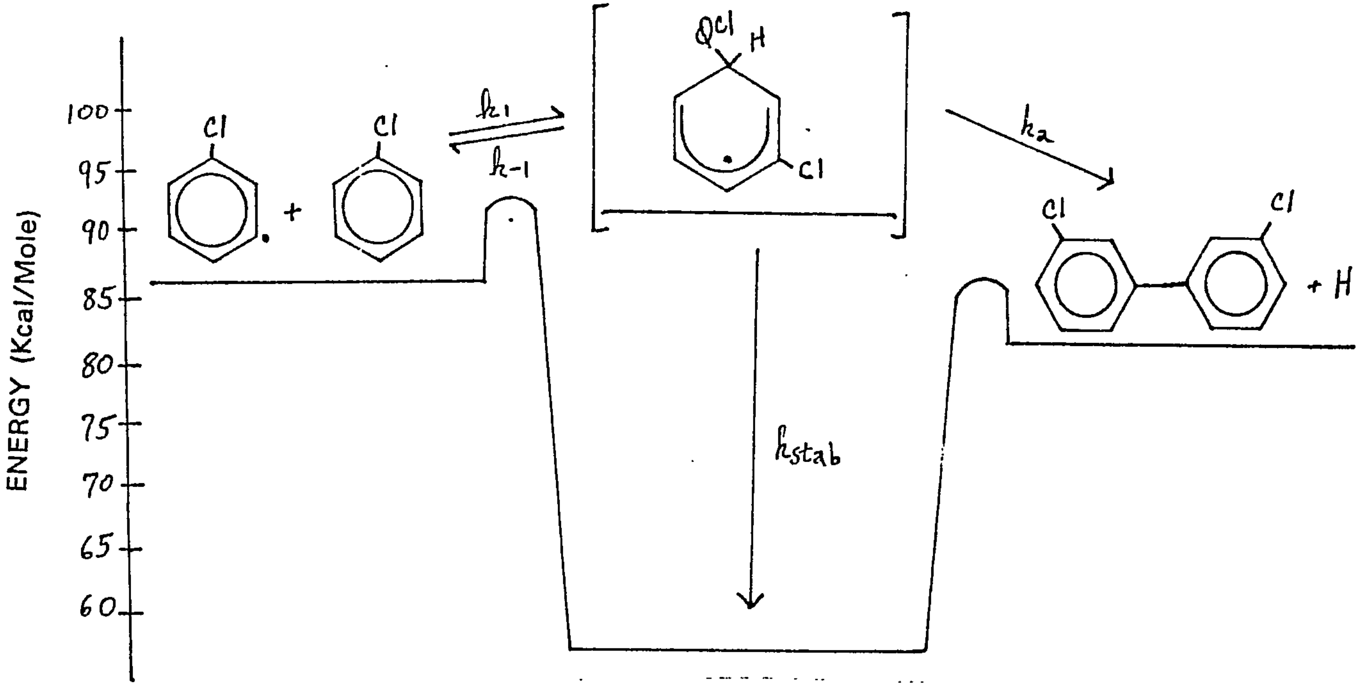
ENERGY DIAGRAM

Figure 2.9



C6H4Cl+C6H5Cl ---> PRODUCTS
PRESSURE DEPENDENCE @ 300 & 2000 K

Figure 2.10



ENERGY DIAGRAM

Figure 2.11

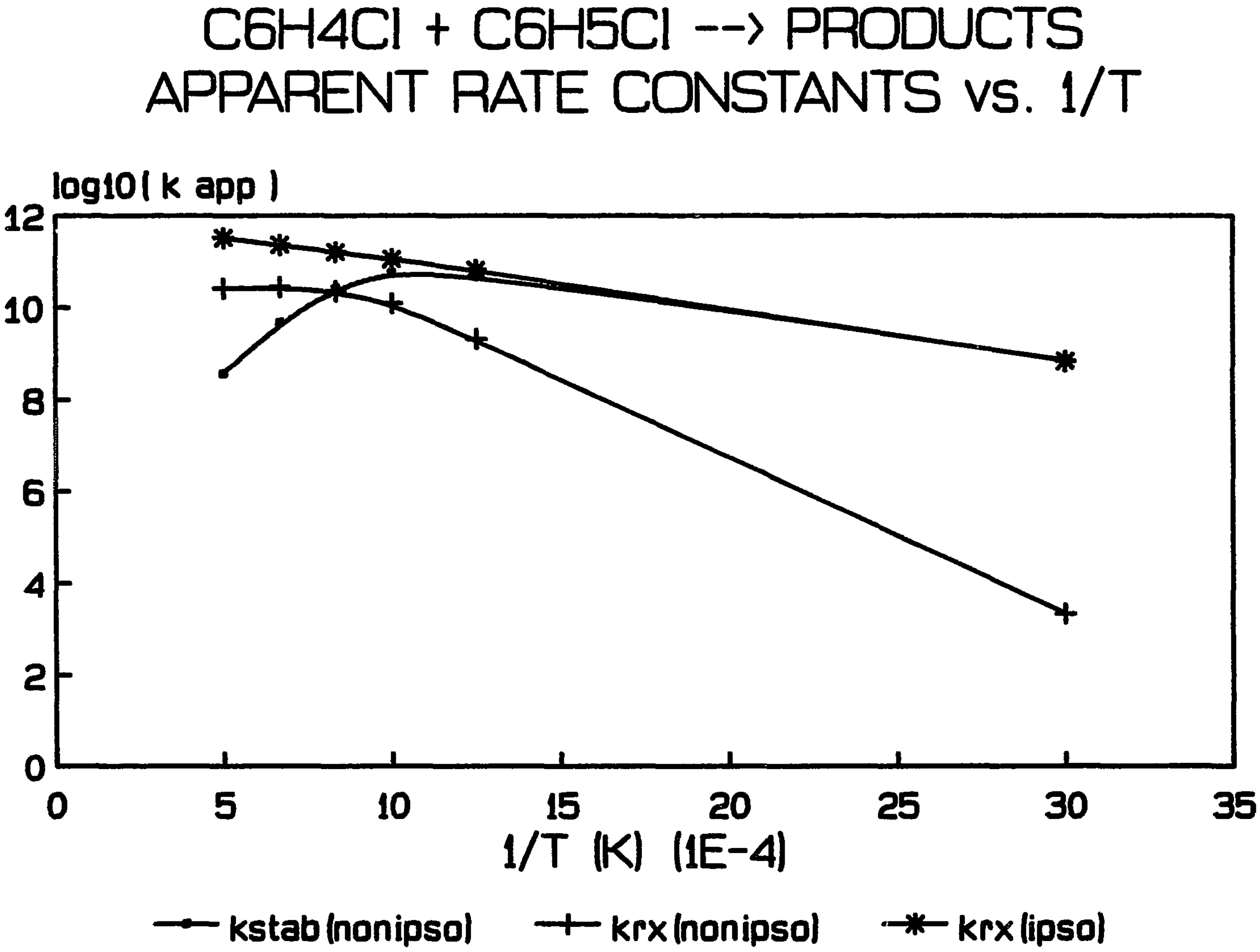

C6H4Cl + C6H5Cl --> PRODUCTS
APPARENT RATE CONSTANTS vs. 1/T
log10(k app)
12
10
8
6
4
2
0
0
5
10
15
20
25
30
35
1/T (K) (1E-4)
kstab (nonipso)
krx (nonipso)
krx (ipso)

Figure 2.12

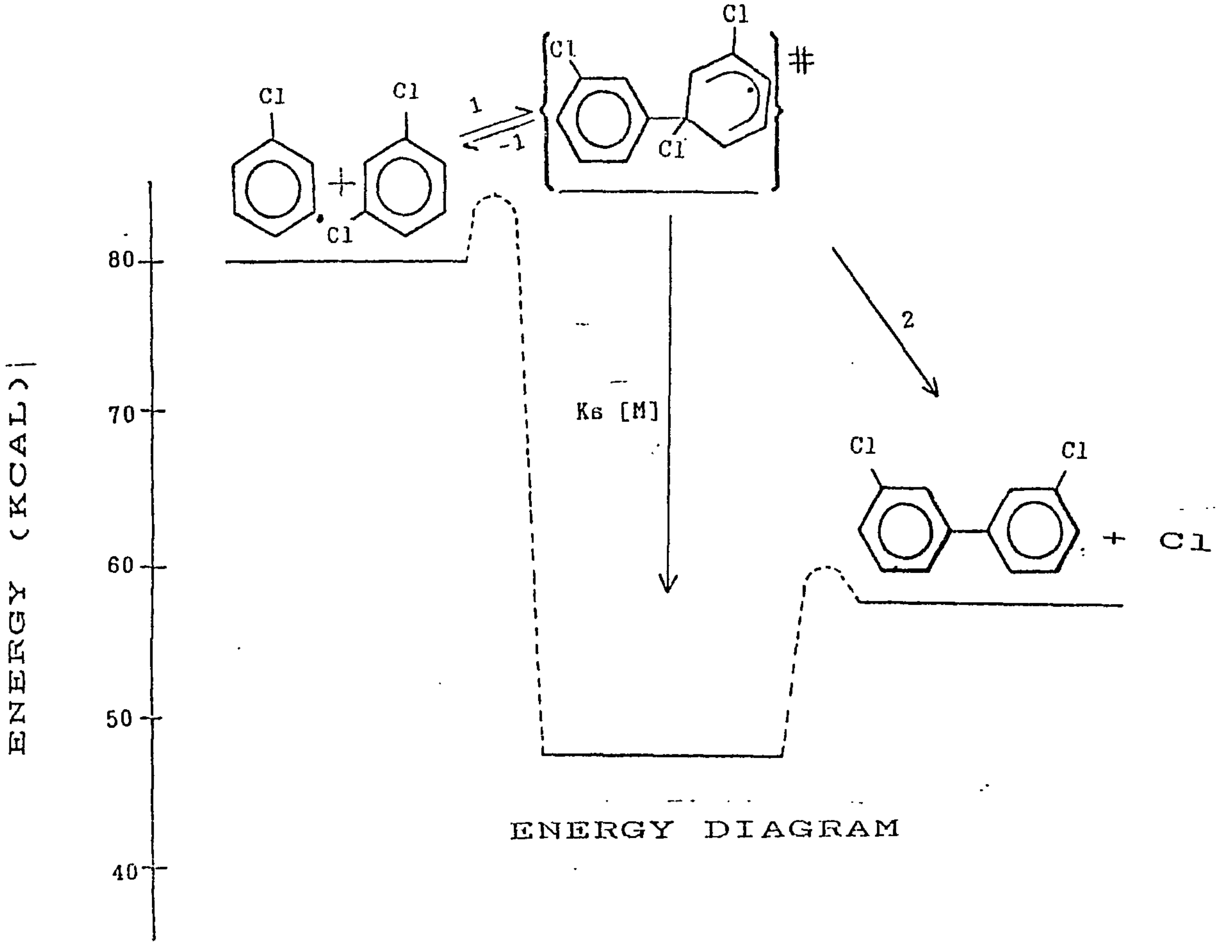

Cl
Cl
Cl
1
-1
Cl
Cl
Cl
#
Ks [M]
2
Cl
Cl
+ Cl
80
70
60
50
40
ENERGY (KCAL)
ENERGY DIAGRAM

Figure 2.13

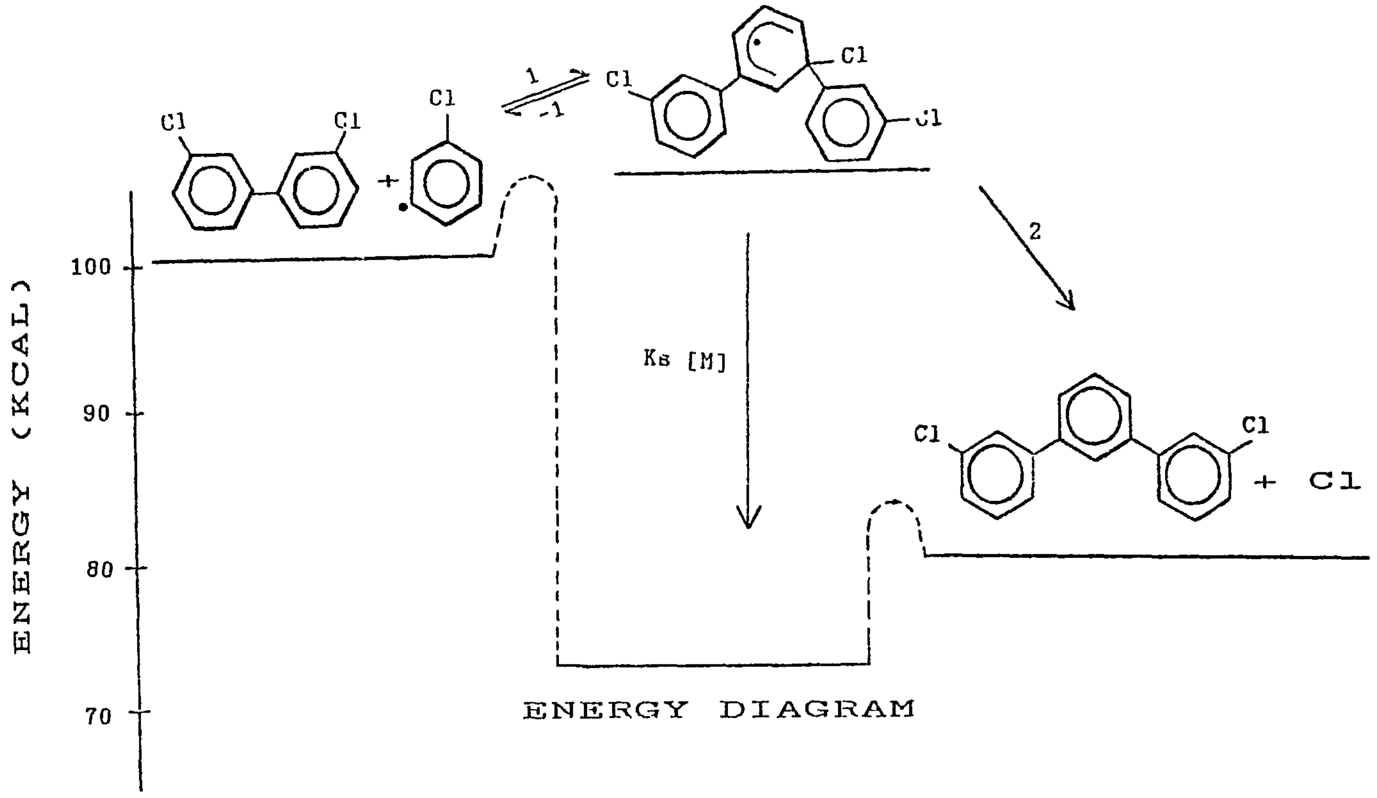

Cl
Cl
+
Cl
1
-1
Cl
Cl
Cl
Ks [M]
2
Cl
Cl
+ Cl
ENERGY (KCAL)
100
90
80
70
ENERGY DIAGRAM

Figure 2.14

C6H4Cl+PHClPHCl ---> PRODUCTS
PRESSURE DEPENDENCE @ 1000 K

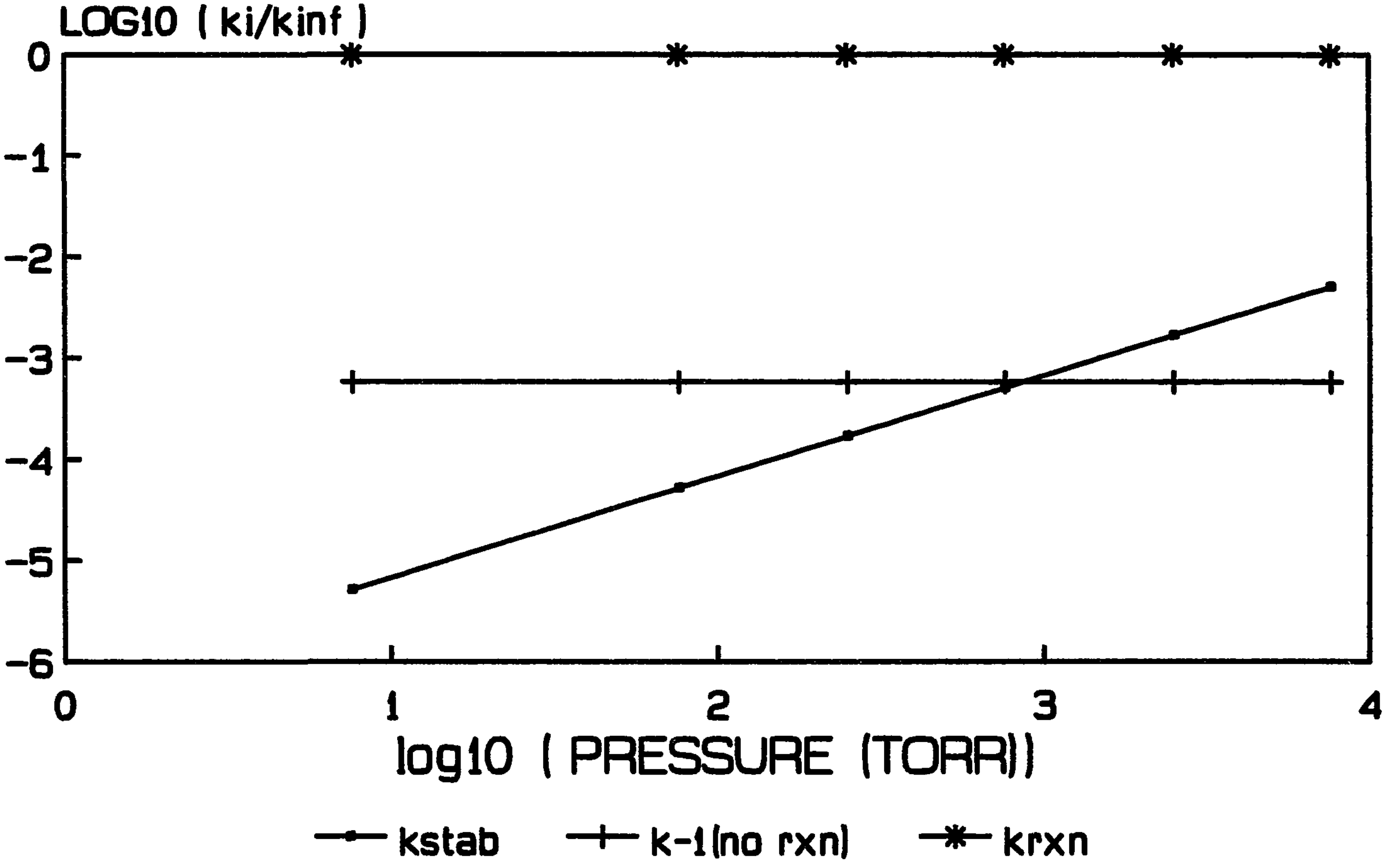


Figure 2.15

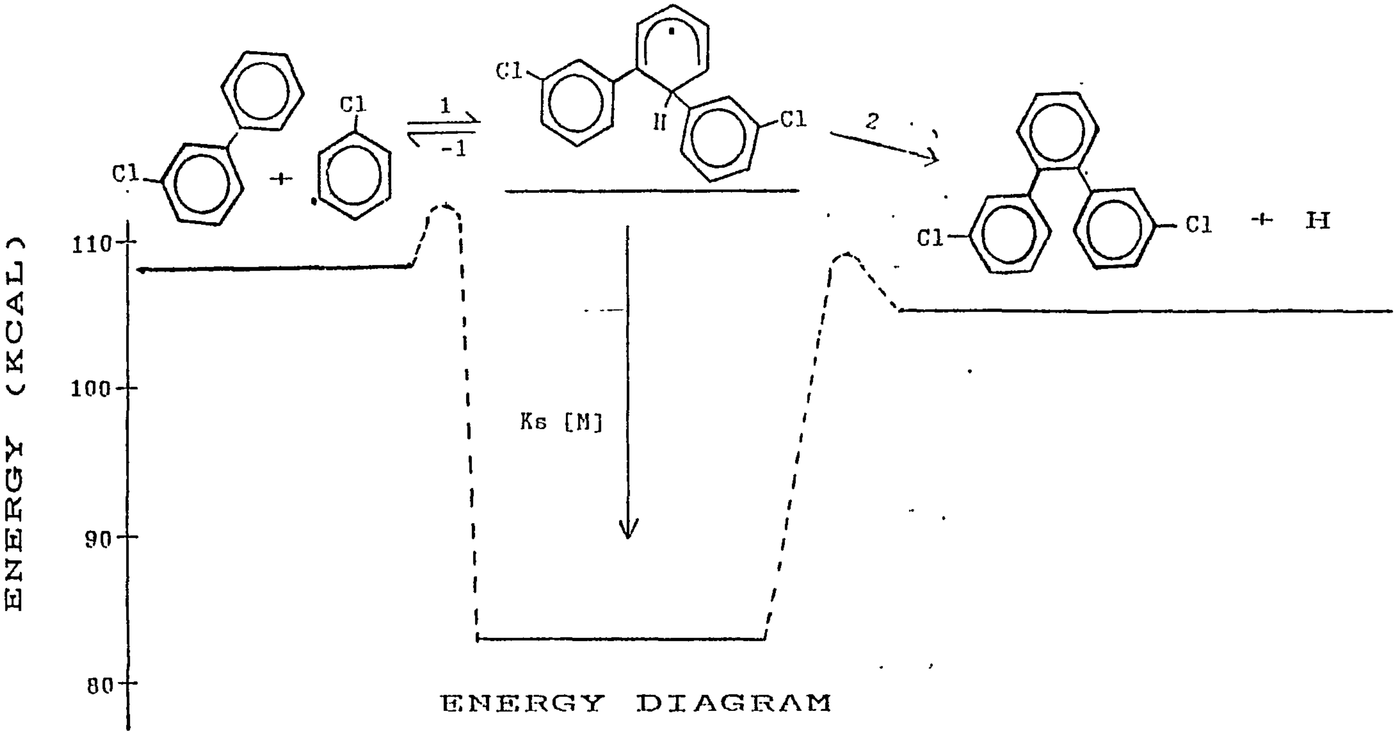
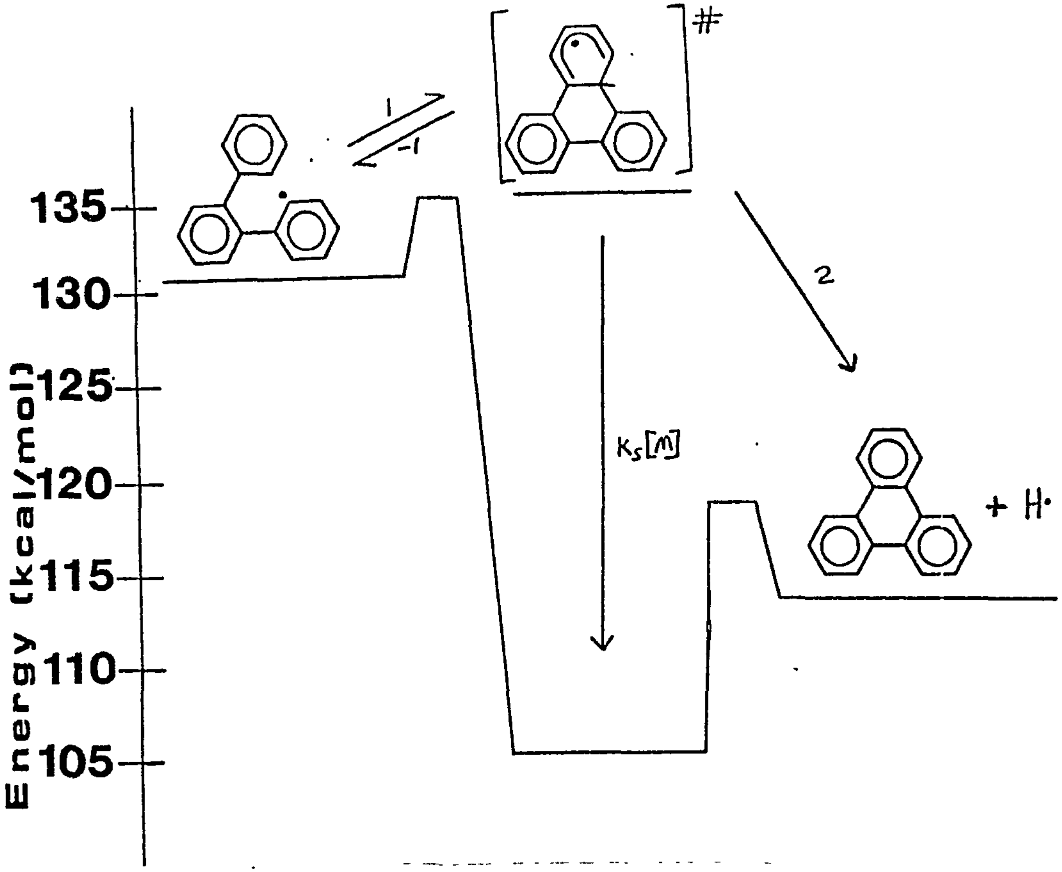

Cl
Cl
1
-1
Cl
H
Cl
2
Cl
Cl
+ H
110
100
90
80
ENERGY (KCAL)
Ks [M]
ENERGY DIAGRAM

Figure 2.16



ENERGY DIAGRAM

Figure 2.17

o-TERPHENYL RADICAL => TRIPHENYLENE + H
PRESSURE DEPENDENCE @ 1000 K

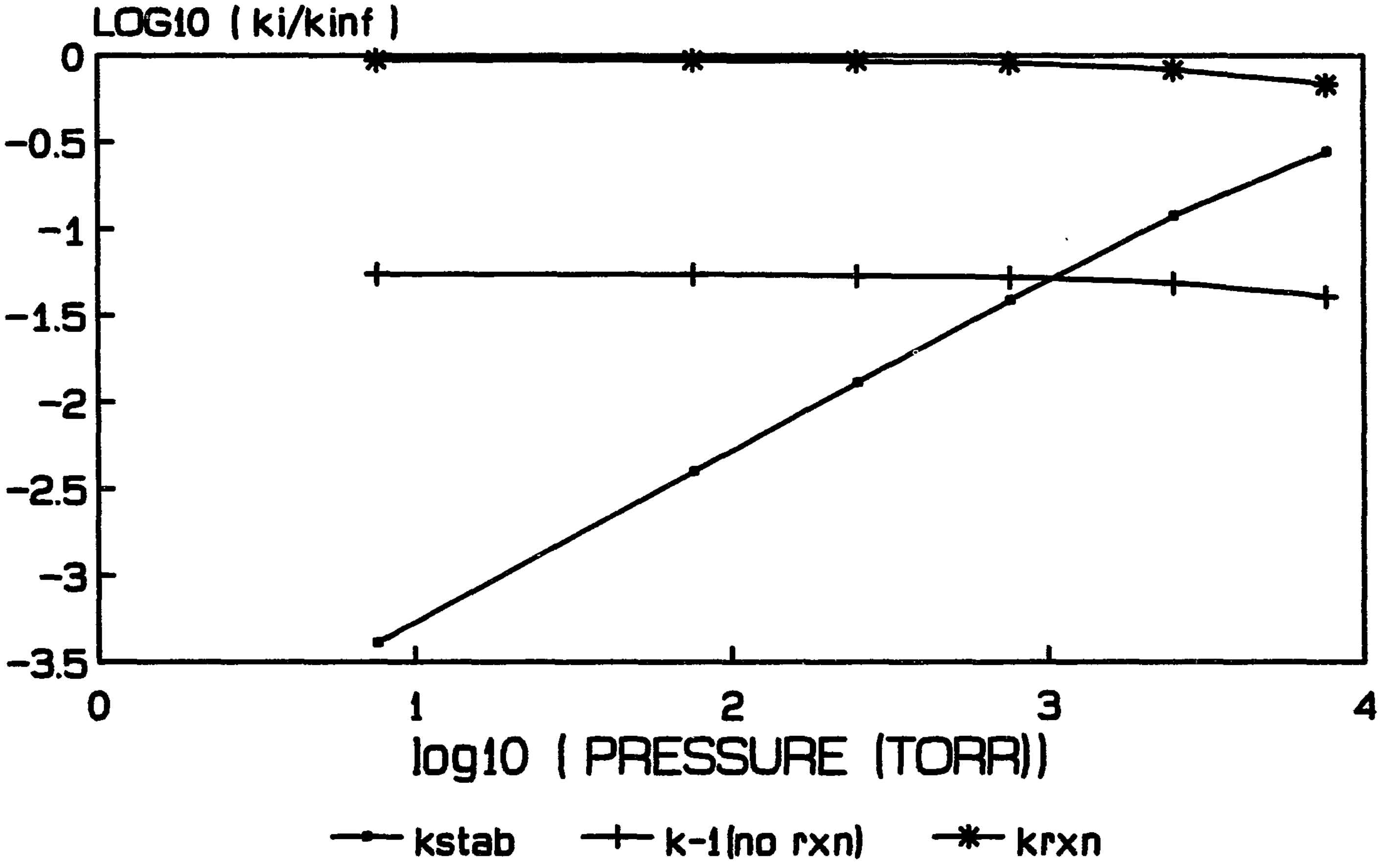


Figure 2.18

C6H4Cl2 PYROLYSIS IN H2 : 1096 deg K
EXPERIMENTAL vs. PREDICTED RESULTS

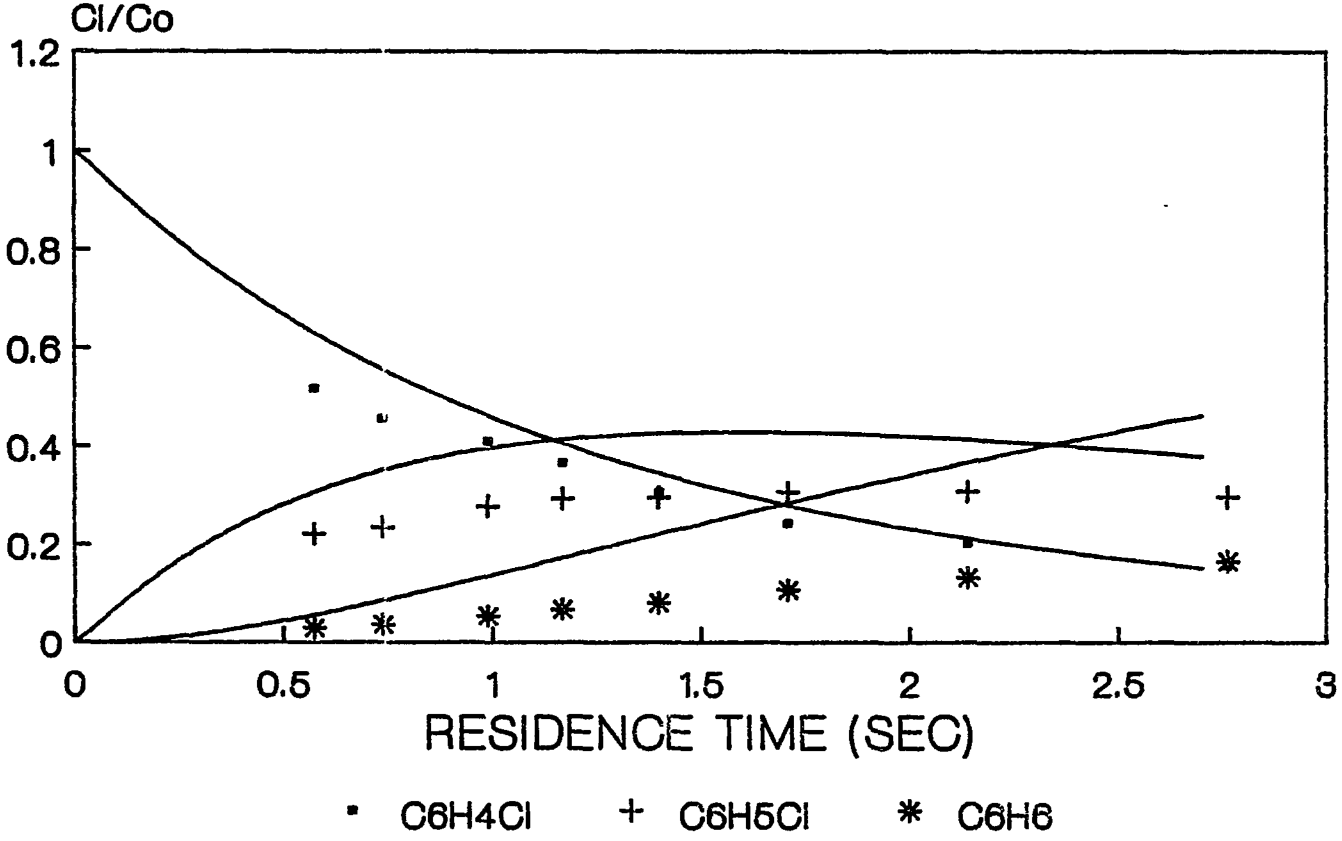


Figure 2.19

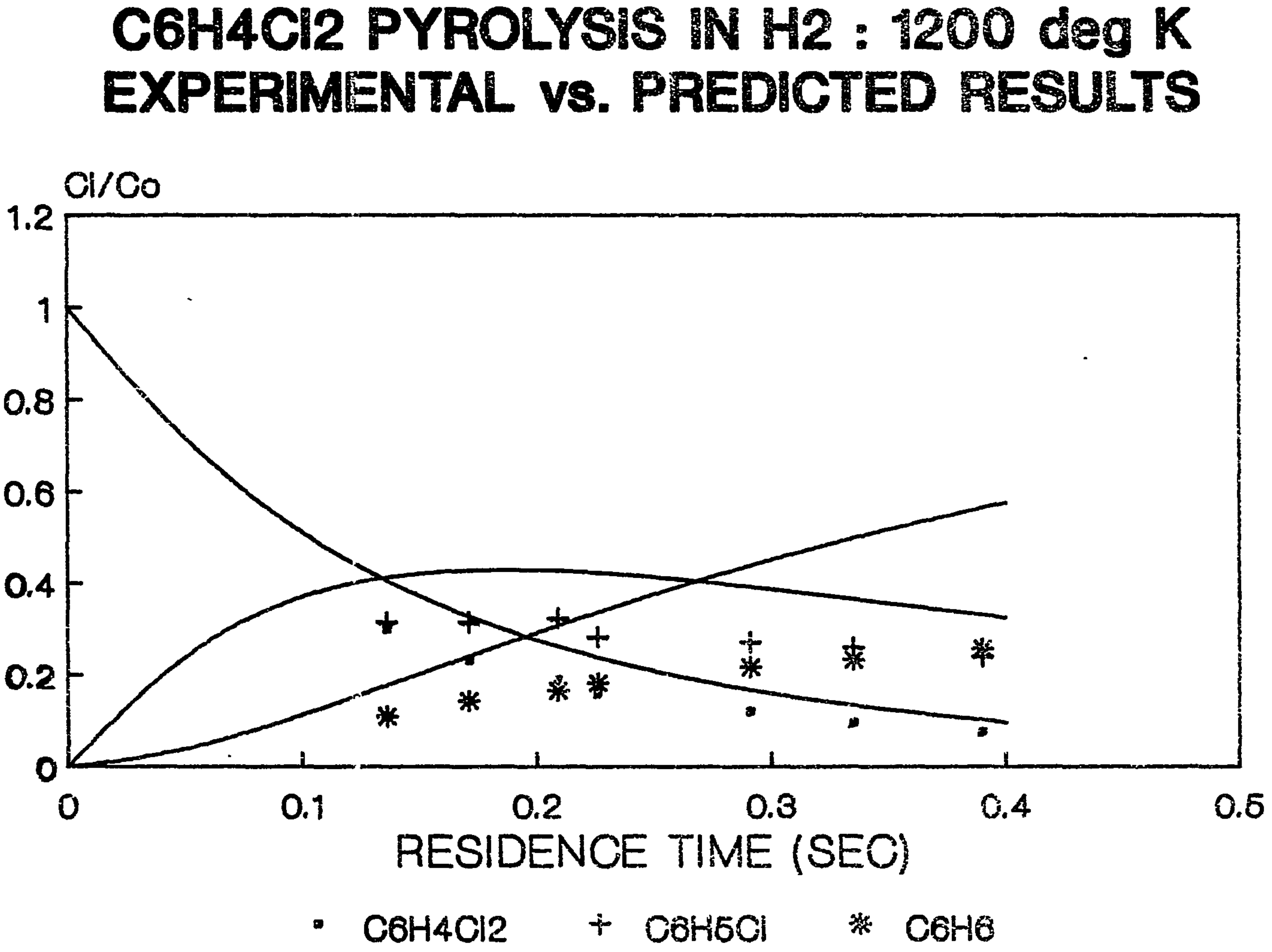
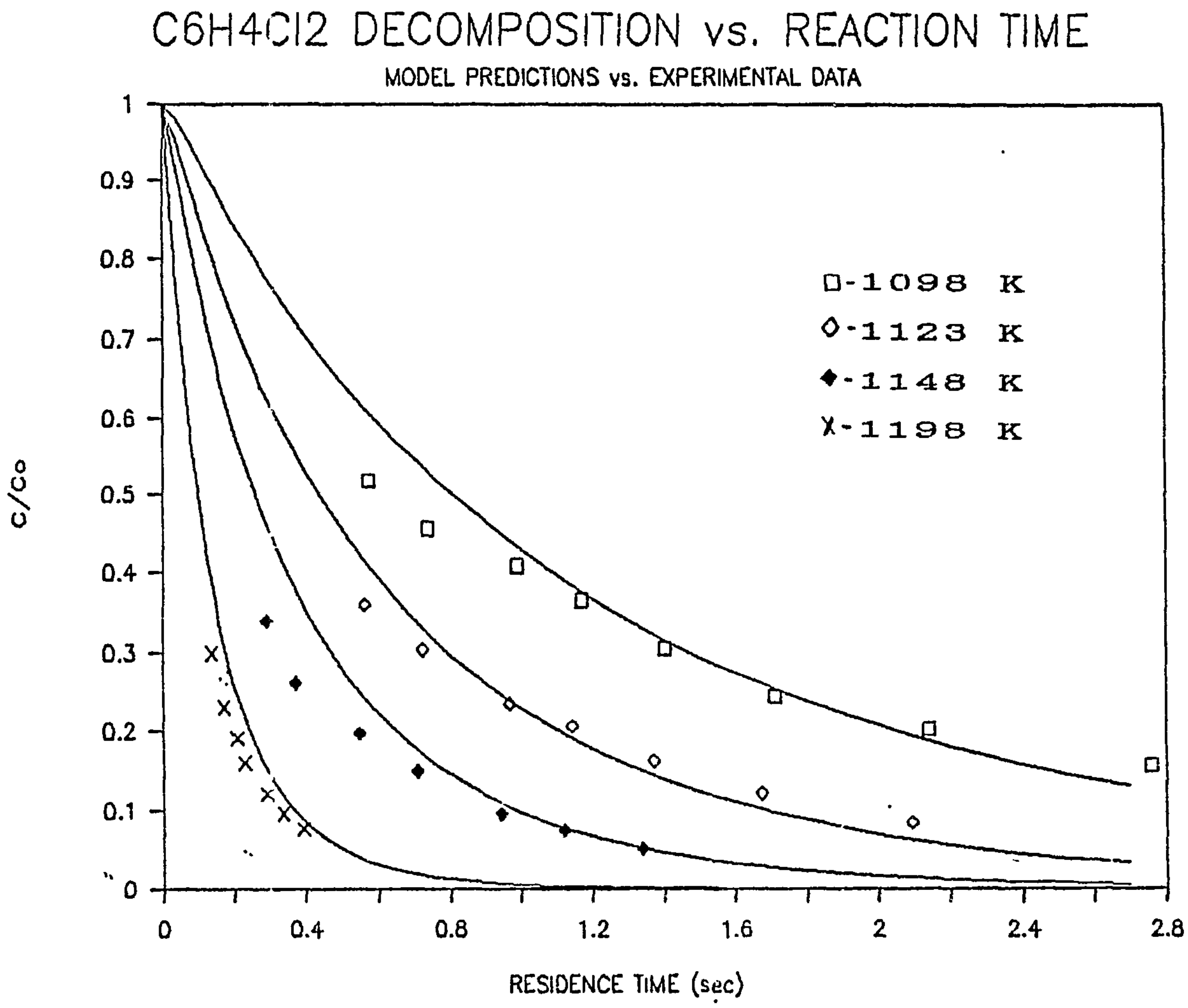

C6H4Cl2 PYROLYSIS IN H2 : 1200 deg K
EXPERIMENTAL vs. PREDICTED RESULTS
Ci/Co
1.2
1
0.8
0.6
0.4
0.2
0
0
0.1
0.2
0.3
0.4
0.5
RESIDENCE TIME (SEC)
▪ C6H4Cl2
+ C6H5Cl
✳ C6H6

Figure 2.20


C6H4Cl2 DECOMPOSITION vs. REACTION TIME
MODEL PREDICTIONS vs. EXPERIMENTAL DATA
c/Co
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.4
0.8
1.2
1.6
2
2.4
2.8
RESIDENCE TIME (sec)
□-1098 K
◇-1123 K
◆-1148 K
X-1198 K

APPENDIX II-C

THERMO PROPERTY LIST for SPECIES CONSIDERED IN DETAILED KINETIC MECHANISM

APPENDIX II-C: THERMO PROPERTY LIST

UNITS (KCAL/MOL; CAL/MOL K)

SPECIES	Hf	S	Cp 300	400	500	600	800	1000	1500	FORMULA
CH3	34.82	46.38	9.26	10.05	10.81	11.54	12.90	14.09	16.26	CH3
CH4	-17.90	44.49	8.49	9.81	11.15	12.48	14.98	17.13	20.57	CH4
C2H2	54.19	48.00	10.60	11.94	13.03	13.93	15.29	16.30	18.34	C2H2
C2H4	12.54	52.39	10.28	12.74	14.92	16.85	20.03	22.45	26.21	C2H4
C2H6	-20.24	54.85	12.58	15.77	18.68	21.31	25.80	29.33	34.91	C2H6
CYC6H10	-1.28	74.27	25.24	34.78	42.74	49.37	59.51	66.65	77.29	C6H10
CYC6H9	45.61	80.05	25.10	34.07	41.50	47.65	56.95	63.42	73.01	C6H9
MECY24PD1	48.10	75.50	23.11	30.25	36.37	41.60	49.85	55.84	64.70	C6H7
BICYC6H7	71.86	71.52	27.62	30.93	34.98	39.40	48.20	55.52	63.32	C6H7
BICYC6H8	39.54	69.19	18.61	27.92	35.67	42.07	51.65	58.09	67.71	C6H8
H	52.11	27.39	4.97	4.97	4.97	4.97	4.97	4.97	4.97	H
CYC6H5	81.37	69.20	18.79	25.18	30.52	34.96	41.70	46.37	53.18	C6H5
CYC6H6	19.82	64.34	19.70	26.93	33.09	38.05	45.35	50.40	57.80	C6H6
CYC6H7	49.86	72.01	20.79	28.59	35.11	40.54	48.74	54.33	62.34	C6H7
LINC6H7	93.05	84.80	27.27	34.43	40.09	44.55	50.85	55.00	62.12	C6H7
CHD14	26.16	70.80	22.63	30.67	37.47	43.20	52.04	58.27	67.47	C6H8
CHD	25.40	72.49	22.67	30.85	37.73	43.49	52.30	58.45	67.56	C6H8
CY13PD5	54.30	62.40	17.24	24.03	29.40	33.61	39.46	43.13	49.10	C5H5
CY13PD	32.40	64.20	17.64	24.43	30.01	34.56	41.29	45.81	52.80	C5H6
MECY24PD3	81.76	75.80	22.91	29.99	36.09	41.31	49.58	55.57	64.44	C6H7
MECY14PD4	79.66	75.40	22.95	30.01	36.09	41.31	49.58	55.57	64.44	C6H7
MECY14PD3	48.06	75.36	23.11	30.25	36.37	41.60	49.85	55.84	64.70	C6H7
MECY24PD	28.30	73.00	23.03	30.31	36.74	42.37	51.51	58.23	67.83	C6H8
NAPTH	36.10	80.21	31.89	43.00	52.13	59.61	70.77	78.43	90.21	C10H8
C*CC*C	26.33	66.62	19.22	24.22	28.36	31.78	36.95	40.57	46.34	C4H6
C*CC*C.	79.82	70.70	18.40	22.71	26.28	29.24	33.75	36.95	42.09	C4H5
C*CC*CC*C	39.60	80.70	28.09	35.79	41.91	46.76	53.70	58.33	66.33	C6H8
C#CC.*C	115.57	68.20	17.50	20.52	22.98	24.97	27.91	29.93	33.27	C4H3
C#CC*C.	126.27	69.60	17.50	20.52	22.98	24.97	27.91	29.93	33.27	C4H3
C#CC#C	112.30	59.86	17.19	19.77	21.77	23.30	25.33	26.55	28.72	C4H2
C#CC#C.	192.20	62.72	14.98	17.00	18.59	19.82	21.50	22.53	24.21	C4H
C#CC#CC#C	170.66	73.74	20.65	24.21	27.51	30.49	35.35	38.60	40.79	C6H2
ME.CY24PD	75.10	76.42	23.11	30.27	36.36	41.53	49.64	55.53	64.71	C6H7
C6H5CH3	11.95	76.41	24.88	33.38	40.53	46.52	55.73	62.22	71.78	C7H8
CYC6H4	112.52	68.88	16.38	22.95	28.22	32.42	38.45	42.42	48.70	C6H4
H2	.00	31.21	6.90	6.95	6.99	7.02	7.10	7.21	7.72	H2
CL	28.90	39.47	5.22	5.37	5.44	5.45	5.39	5.31	5.17	CL
CL2	-.01	53.29	8.10	8.38	8.59	8.74	8.91	8.99	9.10	CL2
C2H	131.98	49.55	8.92	9.61	10.21	10.72	11.55	12.18	13.31	C2H
C6H5CL	12.35	74.79	23.27	30.58	36.47	41.18	47.96	52.41	59.17	C6H5CL
PHCLPHCL	28.72	112.76	44.07	56.76	66.94	75.04	86.58	93.99	104.92	C12H8CL2
PHCL2	6.32	82.09	27.35	34.19	39.70	44.10	50.37	54.36	59.96	C6H4CL2
C6H5C6H5	43.50	93.85	35.74	48.75	59.44	68.19	81.20	90.05	103.37	C12H10
TERPHEN	66.04	123.30	58.70	79.26	95.92	109.33	128.74	141.44	160.44	C18H14
PHPHCL	35.82	105.39	39.90	52.74	63.17	71.60	83.88	92.03	104.14	C12H9CL
HCL	-22.06	44.64	6.96	6.95	6.99	7.07	7.29	7.56	8.10	HCL
C6H4CL	74.34	77.78	22.92	29.16	34.33	38.62	45.16	49.86	57.75	C6H4CL

LINC6H6	82.73	77.88	26.31	32.66	37.79	41.91	47.86	51.75	57.77	C6H6
LINC6H5	136.55	80.70	25.50	31.62	36.41	40.12	45.21	48.39	53.56	C6H5
CY13HD	25.40	72.90	22.44	30.74	37.70	43.51	52.35	58.47	67.44	C6H8
NAPTH.H	127.32	33.89	41.77	46.37	52.56	59.60	73.82	85.40	95.91	C10H9
CHD.C4H3	135.33	89.24	42.06	49.46	57.08	64.36	76.42	84.12	95.22	C10H9
BICYC10H9	115.33	82.47	41.50	53.03	62.44	70.07	81.11	88.15	97.07	C10H9
BICYPD	79.15	90.58	33.71	51.29	64.12	73.26	84.08	89.66	101.32	C10H10
C6H3CL2	66.01	82.84	26.90	34.25	38.96	42.01	46.21	51.46	76.84	C6H3CL2
CYC6H5CL2	28.16	76.24	28.70	35.98	41.70	46.23	52.98	58.28	71.60	C6H5CL2
NAPTHC#C	94.51	94.04	40.07	51.52	61.11	69.08	81.06	89.09	99.78	C12H8
NAPH.C#C	156.06	95.42	39.20	49.90	58.88	66.36	77.66	85.25	95.29	C12H7
PHPH	42.92	92.51	39.07	52.99	64.34	73.54	87.00	95.95	109.41	C12H10
NAPTH	36.00	79.49	31.92	42.95	52.07	59.57	70.74	78.27	89.14	C10H8
NAPH.	97.55	83.62	31.05	41.34	49.85	56.86	67.33	74.42	84.64	C10H7
PHC#C	78.31	79.62	27.59	35.29	41.79	47.24	55.57	61.28	69.02	C8H6
PH.C#C	139.86	82.36	26.72	33.67	39.57	44.54	52.17	57.43	64.52	C8H5
ACENAPTH	66.36	48.35	38.39	49.15	58.07	65.39	76.17	83.14	92.47	C12H8
PHC#C.	156.10	80.98	25.90	32.95	38.95	44.02	51.81	57.13	64.00	C8H5
BICYC6H8	36.88	71.46	21.42	29.39	36.10	41.74	50.52	56.94	67.52	C6H8
BICYC6H7.	68.95	76.97	21.04	29.02	35.51	40.78	48.62	54.14	63.67	C6H7
DHNAPH	60.50	85.79	34.40	46.79	57.02	65.43	78.03	86.69	99.90	C10H10
H.NAPH	82.90	86.69	32.52	44.54	54.41	62.48	74.44	82.53	94.68	C10H9
DHNAPH.	113.94	90.98	34.24	46.12	55.85	63.78	75.48	83.38	95.40	C10H9
CHDC4H3	100.58	98.24	37.50	49.73	59.64	67.63	79.24	86.91	98.68	C10H10
CHD.C4H3	125.04	97.76	35.63	47.46	57.01	64.67	75.68	82.80	93.46	C10H9
ICLPHCHD	40.45	107.18	44.32	59.87	72.37	82.36	96.67	105.98	120.26	C12H11CL
ICLPHCHD.	64.91	106.70	42.45	57.60	69.74	79.40	93.11	101.87	115.04	C12H10CL
CLIPHCHD	40.90	107.91	44.26	59.51	71.85	81.78	96.12	105.50	119.57	C12H11CL
CLIPHCHD.	65.36	107.43	42.39	57.24	69.22	78.82	92.56	101.39	114.35	C12H10CL
PHCLCHD	44.49	109.63	44.82	60.01	72.26	82.08	96.21	105.41	119.35	C12H11CL
PHCLCHD.	68.95	109.15	42.95	57.74	69.63	79.12	92.65	101.30	114.13	C12H10CL
ICLC4H3CD	89.44	104.54	41.21	53.50	63.31	71.09	82.13	89.25	100.50	C10H9CL
CLC4H3CD.	113.90	105.44	39.34	51.23	60.68	68.13	78.56	85.14	95.28	C10H8CL
4CLICLCD	5.68	88.93	28.92	37.51	44.36	49.80	57.50	62.46	70.22	C6H6CL2
4CLICLCD.	30.14	88.45	27.05	35.25	41.74	46.84	53.94	58.36	65.00	C6H5CL2
CLHNAPH	56.54	90.84	35.60	49.87	61.03	69.69	81.62	89.17	101.99	C10H9CL
CLHNAPH.	109.98	94.65	35.44	49.20	59.86	68.04	79.07	85.86	97.49	C10H8CL
CL.NAPH	78.94	90.36	33.73	47.60	58.40	66.73	78.06	85.06	96.77	C10H8CL
CLICLPH	29.76	114.21	47.91	63.35	75.68	85.46	99.26	108.04	121.36	C12H10CL2
CLICLPH.	54.22	113.70	46.04	61.09	73.06	82.50	95.70	103.93	116.14	C12H9CL2
R46CLIPCL	22.66	122.96	52.08	67.34	79.41	88.86	101.94	110.02	122.23	C12H9CL3
R46CLIPCL.	47.12	122.42	50.21	65.08	76.79	85.91	98.38	105.91	117.01	C12H8CL3
ICLPCLCD	33.35	115.93	48.48	63.86	76.10	85.77	99.35	107.95	121.14	C12H10CL2
ICLPCLCD.	57.81	115.45	46.61	61.59	73.47	82.81	95.79	103.85	115.92	C12H9CL2
CLIPCLCD	34.15	116.05	48.41	63.51	75.59	85.18	98.78	107.49	120.91	C12H10CL2
CLIPCLCD.	58.61	115.57	46.54	61.25	72.96	82.22	95.21	103.39	115.69	C12H9CL2
TRIPHENE	61.80	104.80	56.89	75.42	90.70	103.22	121.71	133.90	150.66	C18H12
TERPHEN	66.04	123.30	58.70	79.26	95.92	109.33	128.74	141.44	160.44	C18H14
TERPHEN.	127.59	126.86	57.84	77.65	93.70	106.63	125.34	137.59	155.94	C18H13
CL2CHDR	30.48	88.45	27.55	34.75	40.06	44.91	53.01	57.69	.00	C6H5CL2
CL2CHD	5.96	88.93	29.42	37.04	42.70	47.91	56.57	61.72	.00	C6H6CL2
CLCHD.	41.17	81.42	23.71	31.38	37.37	42.41	51.46	56.74	.00	C6H6CL
BZ	19.80	64.24	19.44	26.64	32.76	37.80	45.24	50.46	58.38	C6H6
CLCHD	16.65	81.90	25.58	33.67	40.01	45.41	55.02	60.77	.00	C6H7CL

CYC6H7	49.93	72.01	20.79	28.43	35.12	40.56	48.71	54.41	.00	C6H7
CHD13	25.41	72.49	22.66	30.72	37.76	43.56	52.27	58.44	67.52	C6H8
CLMECHD.	32.31	86.43	29.67	39.82	47.40	53.30	63.07	69.46	.00	C7H8CL
CLMECHD	7.79	86.91	31.54	42.11	50.04	56.30	66.63	73.49	.00	C7H9CL
MECHD.	43.45	80.13	26.02	35.32	43.54	50.05	59.97	66.89	.00	C7H9
MECHD	18.93	80.61	27.89	37.61	46.18	53.05	63.53	70.92	82.02	C7H10
CPICL2CD.	47.15	122.13	50.46	65.26	76.10	84.77	97.48	105.60	.00	C12H8CL3
BPH3R	57.84	115.10	46.62	61.89	73.41	82.27	95.93	104.65	.00	C12H9CL2
BPH1R	54.56	113.73	46.53	61.35	72.40	81.36	94.68	103.33	.00	C12H9CL2
BPH2R	54.36	113.85	46.72	60.26	71.19	80.59	93.13	101.55	.00	C12H9CL2
CPICL2CD	22.63	122.61	52.33	67.55	78.74	87.77	101.04	109.63	.00	C12H9CL3
BPH2	29.84	114.33	48.59	62.55	73.83	83.59	96.69	105.58	.00	C12H10CL2
BPH1	30.04	114.21	48.40	63.64	75.04	84.36	98.24	107.36	.00	C12H10CL2
BPH3	33.32	115.58	48.49	64.18	76.05	85.27	99.49	108.68	.00	C12H10CL2
PHICLCHD.	65.05	106.82	42.88	56.89	68.50	78.09	91.58	100.60	.00	C12H10CL
PHICLCHD	40.53	107.30	44.75	59.18	71.14	81.09	95.14	104.63	.00	C12H11CL
PHCHD.	75.39	100.40	39.04	53.48	65.85	75.61	90.03	99.81	.00	C12H11
PHCHD	50.87	100.88	40.91	55.77	68.49	78.61	93.59	103.84	118.58	C12H12
TRIPHCL2	46.98	121.02	68.82	83.18	98.12	110.08	127.28	139.66	.00	C18H10CL2
OPH3CL2R	162.17	143.27	66.71	84.76	99.47	110.64	126.26	136.31	.00	C18H11CL2
TRIPHCL	54.39	115.38	64.89	79.27	94.42	106.67	124.48	137.39	.00	C18H11CL1
TPHF2R	94.04	129.44	60.51	81.27	98.84	112.16	131.22	143.19	.00	C18H11CL2
TRIPHENE	61.80	104.80	60.96	75.36	90.72	103.26	121.68	135.12	153.96	C18H12
TPHF3R	101.45	122.42	56.58	77.36	95.14	108.75	128.42	140.92	.00	C18H12CL1
TPHF2	69.52	129.92	62.38	83.56	101.48	115.16	134.78	147.22	.00	C18H12CL2
PHPHPHCL2	51.22	139.52	66.54	86.86	103.40	116.10	134.20	146.08	.00	C18H12CL2
TPH7R	72.67	149.40	70.58	92.31	108.91	121.58	140.56	152.75	.00	C18H12CL3
OPH3CLR	121.04	132.57	63.56	82.04	97.39	109.32	126.47	137.84	.00	C18H12CL1
MPH3CL	51.22	139.52	66.54	86.86	103.40	116.10	134.20	146.08	.00	C18H12CL2
OPH3CL2	53.61	138.47	68.36	87.56	103.31	115.43	132.68	143.95	.00	C18H12CL2
OPH3CL2	53.61	138.47	68.36	87.56	103.31	115.43	132.68	143.95	.00	C18H12CL2
TPHF1R	108.86	115.40	52.65	73.45	91.44	105.34	125.62	138.65	.00	C18H13
PHPHPHCL	58.63	132.50	62.61	82.95	99.70	112.69	131.40	143.81	.00	C18H13CL
TPH4R	79.88	141.12	66.84	87.31	104.00	117.40	136.21	148.70	.00	C18H13CL2
TPHF3	76.93	121.52	58.45	79.65	97.78	111.75	131.98	144.95	.00	C18H13CL
TPH3R	80.08	141.00	66.65	88.40	105.21	118.17	137.76	150.48	.00	C18H13CL2
TPH7	48.15	149.88	72.45	94.60	111.55	124.58	144.12	156.78	.00	C18H13CL3
TPH8R	82.81	143.10	66.93	87.81	105.05	118.33	137.46	150.18	.00	C18H13CL2
TPH6R	80.08	141.00	66.65	88.40	105.21	118.17	137.76	150.48	.00	C18H13CL2
OTERPHENR	129.98	125.55	59.63	78.13	93.69	105.90	123.67	135.57	156.07	C18H13
TPH11R	82.81	143.10	66.93	87.81	105.05	118.33	137.46	150.18	.00	C18H13CL2
OPH3CL	61.02	131.45	64.43	83.65	99.61	112.02	129.88	141.68	.00	C18H13CL
OTERPHEN	68.43	121.68	60.50	79.74	95.91	108.61	127.08	139.41	160.57	C18H14
MPH3	66.04	122.73	58.68	79.04	96.00	109.28	128.60	141.54	160.42	C18H14
MPHPHPH	66.04	122.73	58.68	79.04	96.00	109.28	128.60	141.54	160.42	C18H14
TPH4	55.36	141.60	68.71	89.60	106.64	120.40	139.77	152.73	.00	C18H14CL2
TPH3	55.56	141.48	68.52	90.69	107.85	121.17	141.32	154.51	.00	C18H14CL2
TPH2R	87.49	132.61	62.72	84.49	101.51	114.76	134.96	148.21	.00	C18H14CL
TPH12R	90.22	134.70	63.00	83.90	101.35	114.92	134.66	147.91	.00	C18H14CL
TPH1R	90.22	134.70	63.00	83.90	101.35	114.92	134.66	147.91	.00	C18H14CL
TPH11	58.29	142.20	68.80	90.10	107.69	121.33	141.02	154.21	.00	C18H14CL2
MTERPH	66.04	122.73	58.68	79.04	96.00	109.28	128.60	141.54	160.42	C18H14
TPHF1	84.34	114.50	54.52	75.74	94.08	108.34	129.18	142.68	161.66	C18H14
TPH8	58.29	143.58	68.80	90.10	107.69	121.33	141.02	154.21	.00	C18H14CL2

TPH6	55.56	141.48	68.52	90.69	107.85	121.17	141.32	154.51	.00	C18H14CL2
TPH5R	90.22	134.70	63.00	83.90	101.35	114.92	134.66	147.91	.00	C18H14CL
TPH2	62.97	133.09	64.59	86.78	104.15	117.76	138.52	152.24	.00	C18H15CL
TPH1	65.70	135.18	64.87	86.19	103.99	117.92	138.22	151.94	.00	C18H15CL
TPH12	65.70	135.18	64.87	86.19	103.99	117.92	138.22	151.94	.00	C18H15CL
TPH9R	97.63	126.31	59.07	79.99	97.65	111.51	131.86	145.64	.00	C18H15
TPH10R	97.63	126.31	59.07	79.99	97.65	111.51	131.86	145.64	.00	C18H15
TPH5	65.70	135.18	64.87	86.19	103.99	117.92	138.22	151.94	.00	C18H15CL
TPH10	73.11	126.79	60.94	82.28	100.29	114.51	135.42	149.67	170.08	C18H16
TPH9	73.11	126.79	60.94	82.28	100.29	114.51	135.42	149.67	170.08	C18H16
OPH3CL2R	113.63	140.97	67.49	85.95	101.09	112.73	129.27	140.11	.00	C18H12CL2

SECTION III

THERM: THERMODYNAMIC PROPERTY ESTIMATION COMPUTER CODE FOR USE WITH DETAILED REACTION KINETIC MECHANISMS

THERM: THERMODYNAMIC PROPERTY ESTIMATION COMPUTER CODE FOR USE WITH DETAILED REACTION KINETIC MECHANISMS

ABSTRACT

Detailed reaction kinetic models using mechanisms, based upon fundamental thermodynamic and kinetic principles, are currently being developed by researchers attempting to emulate reaction systems in combustion, CVD, and other related fields. A major requirement for these simulations is accurate thermodynamic property data (estimated or experimental) for all molecular and radical species considered in a mechanism.

We have developed a computer code for an IBM PC/XT/AT or compatible which can be used to estimate, edit, or enter thermodynamic property data for gas phase radicals and molecules using Benson's group additivity method.<1,2,3,4> The computer code is called **THERM** (**TH**ermo **E**stimation for **R**adicals and **M**olecules). All group contributions considered for a species are recorded and thermodynamic properties are generated in NASA polynomial format (for compatibility with CHEMKIN<5>) in addition to listings more convenient for thermodynamic, kinetic and equilibrium calculations. Polynomial coefficients are valid from 300-5000 K. In addition, THERM contains a chemical reaction interpreter which can be used to calculate thermodynamic property changes for chemical reactions as functions of temperature. This interpreter can process CHEMKIN input files.

INTRODUCTION

Benson's group estimation technique is an accurate method for the estimation of ideal gas phase heat capacities, heats of formation, and entropies of molecules. This technique is thoroughly discussed in Benson's text **Thermochemical Kinetics**<1> as well as other sources.<2,3,4,6> This method assumes that the properties for a chemical substance are the sum of the contributions from each group or polyvalent atom (central atom) in that molecule. It is referred to as a second order estimation technique since next-nearest-neighbor corrections, and to some extent, chemical structure are accounted for. Estimation based upon chemical bond contributions alone are known as first order estimation techniques, while those based upon atomic contributions alone are referred to as zero order techniques.<7> The second order estimates are naturally more accurate than lower order techniques, however, more information about the molecule is required. Several other estimation techniques based upon group contribution principles are presented in Reid, Prausnitz, and Sherwood.<4> This method, however, has become the most widely accepted due, in part, to its ease of use and relative accuracy when compared with other techniques.

Group contributions were derived by Benson and coworkers<2> and various other researchers <6,8,9,10,11,12,13,14,15> by breaking similar groups of molecules with known thermodynamic properties into their constituent groups and then performing multivariable linear regression to find group contributions which gave the best fit to available experimental property data.

To use Benson's method, one must become familiar with the notation used to describe the groups. For example if we consider chlorobenzene (C_6H_5Cl) we say that it has one C_B-(Cl) group and five C_B-(H) groups. C_B-(Cl) refers to a carbon atom in a benzene ring which is attached to a Cl atom, while C_B-(H) refers to a carbon atom in a benzene ring which is attached to an H atom. THERM identifies C_B-(Cl) as CB/CL and C_B-(H) as CB/H. For a second example consider propane (C_3H_8). This molecule contains two C-(H_3)(C) groups for the two terminal methyls, and one C-(H_2)(C_2) for the center CH_2 group. THERM identifies C-(H_3)(C) as C/C/H3, with C-(H_2)(C_2) identified as C/C2/H2. In addition, various corrections for symmetry, optical isomers, ring structure, steric effects must be included where necessary. More examples are presented in the THERM Users Manual which appears in appendix III - C.

OTHER COMPUTER IMPLEMENTATIONS OF BENSON'S METHOD:

Benson's method has been programmed by others.<16,17,18> The most notable implementation of this estimation technique on computer is the code called CHETAH<16,17> which is distributed by ASTM. This program has been in use since 1974. Its primary purpose is to predict chemical reactivity hazards. Many publications are available which demonstrate its usefulness.<7>

THERM was not designed to compete with or replace this code. It was written to automate several tasks which are routine to high temperature reaction modeling studies. There are several important differences between THERM and CHETAH. THERM estimates properties between 300 and 5000 K while CHETAH estimates are valid from 300 - 1500 K, a range less suitable for combustion studies. THERM generates thermodynamic property functions in the NASA format but does not address hazard assessment. Properties for both radicals and molecules can be estimated using THERM, while CHETAH is used for stable molecules only. The user interface for THERM is a little more friendly; CHETAH requires the user to look up a reference number for each group in a molecule and then enter the reference numbers to build a species (reference numbers consist of the table and line number for a particular group). THERM, on the other hand, identifies the group name directly. Other codes have been developed, however, which break a molecule down into its constituent groups automatically after the user enters a structural representation of the species;<18> THERM makes no attempt to determine groups from a graphical representation such as this.

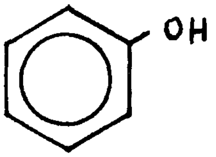
The method for calculation of radical species is similar to that implemented by Muller, Scacchi, and Come<18>, but was developed independently.

SOURCE OF GROUP CONTRIBUTION DATA:

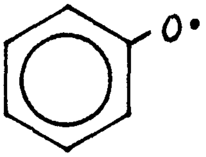
Group data for molecules containing C, O, N, S, and halogen atoms were obtained from Benson<1,2,3>; Reid, et al.<4>; Stein and Fahr<12>; Cohen<6>; and Patel, Ritter, and Bozzelli.<13> Radical group contributions were derived from Benson<1>, McMillen and Golden<19>, Griller<20> and Lossing.<21,22> Bond dissociation increments<14> were derived from Tsang<23>, Slagle and Gutman<24,25>, McMillian and Golden.<19> Ring corrections for C, O, N, and S containing rings were obtained by Patel, Ritter, and Bozzelli.<13>

SOME EXAMPLES OF THE USE OF THIS PROGRAM:

In Table III.i, $\triangle H_f^o$ for 27 assorted species are compared with literature values obtained from various sources as noted. In most cases, estimates agree within 1 Kcal/mol. Deviations range from 0.0 Kcal/mol for chloroethane (C_2H_5Cl) to 3.61 Kcal/mol for pyrene ($C_{16}H_{10}$).



PHENOL



PHENOXY

The structures for phenol and phenoxy radical are shown above. $\triangle H_f^o$(298 K) for phenol (C_6H_6O) is estimated as -22.30 Kcal/mol; while Burcat<26> gives -23.03 Kcal/mol. Seven group contributions are considered for phenol (see

Fig. 3.1). Phenol is comprised of 5 CB/H groups [CB-(H)], one CB/O group [CB-(O)], and one O/CB/H group [O-(H)(C)]. Groups in brackets refer to names appearing in Benson<1>; while those with slashes are the equivalent group name identified by THERM (see Fig. 3.1). Figure 3.1 is a sample of the documentation generated when estimating a species. Elemental formula (C6H5OH), Cp infinity (limiting high temperature heat capacity), symmetry number used to correct entropy, and the number of rotors in the molecule (if any) are also recorded (see table III.ii for examples).

Properties for radicals may be estimated two different ways. One may estimate thermodynamic properties directly with radical group contributions as described in Benson.<1> A second method applies a bond dissociation (BD) increment to stable molecules reflecting loss of an H atom from that species. We prefer this method, especially when estimating resonantly stabilized radicals. Documentation for phenoxy radical, which was estimated in this way, is shown in figure 3.1. Additional notes which are recorded for radicals include bond dissociation energy (298 K) for loss of hydrogen atom from the parent molecule, entropy correction for a free electron, and the name of the parent molecule. ΔH_f(298) for phenoxy is estimated as 12.10 Kcal/mol while Burcat<26> gives ΔH_f equal to 11.40 Kcal/mol a difference of only 0.70 Kcal/mol as shown in Table III.i.

Figure 3.2 shows NASA format polynomial coefficients generated by THERM for phenol and phenoxy radical (this format is required by CHEMKIN). Polynomial coefficients are valid over the temperature range from 300 - 5000 K. The first line of figure 3.2 includes the species name, creation date/reference, elemental composition, phase, lower and upper temperature limits for the polynomial, and the break point temperature (T_{bk}). Fourteen coefficients follow. The first seven coefficients are used to calculate properties at temperatures above T_{bk}, while the second seven coefficients apply to temperatures below T_{bk} (as described by Kee<31> and Burcat<26>). The break point temperature (T_{bk}) is located at a point of tangency between the two polynomials. T_{bk} is determined by matching Cp, H_f^o, and S^o to 0.005 % ; while dCp/dT, dHf/dT, and dS/dT are matched within an absolute tolerance of 0.005. Solving for a point of tangency between two polynomials differs from the method cited by Kee, et.at.<31> It should be noted that midpoint temperatures for our polynomials determined by this matching criteria are usually not the more common 1000 K calculated with other fitting methods.<27,28,29>

In Fig. 3.3, a portion of a thermo table lists data for several substituted aromatics. The list contains species name, $\triangle H_f(298)$, $\triangle S(298)$, and Cp at 300 - 1500 K. In addition, creation date, elemental composition, and phase (g) are recorded. An extended table to 5000 K can also be generated. Units for this table are Kcal/mol for $\triangle H_f$ and cal/mol K for $\triangle S$ and Cp, however tables can also be created with units of KJ/mol (J/mol K).

Note that figure 3.1 lists Cp (1500K) as 0.0 since no contribution was available at this temperature for one of the groups in phenol. However, the temperature range for Cp data can be extended with THERM to 5000 K by using available low temperature Cp data (300 - 1000 K). This is done by projecting an asymptote to the high temperature limiting heat capacity (Cp infinity). To do this we use an harmonic oscillator equation (HOE) derived from statistical mechanics to describe vibrational contributions to Cp. Two polynomials which overlap by 500 K are then determined by linear regression of data approximated with this equation. A point of tangency (T_{bk}) is determined between the polynomials using the Newton/Raphson method.

Heat capacity estimates determined in this fashion are compared with literature data in figure 3.4 for dimethyl ether, phenol, and biphenyl. It should be emphasized that the solid lines are generated from group additivity values alone, while symbols (literature values) result from rigorous calculations requiring frequency data for all vibrational modes in addition to contributions from all restricted rotors in the particular species. We obtain excellent agreement with typical maximum errors less than two percent. Literature data for this comparison are obtained from Stull<30> for dimethyl ether and from

Burcat<26> for biphenyl and phenol. Heat capacity estimates for other species agree similarly when compared to available literature data.

Once one records an estimate with THERM (figure 3.1), polynomials and tables (figures 3.2 and 3.3) can be recalculated automatically to reflect changes or improvements made to the group contribution database.

One may enter data from the literature dirctly into the data files. Automatic recalculation does not effect literature data, or any species where auto-recalculation has been disabled.

Reaction analysis is another useful function incorporated into this code. One may calculate thermodynamic property changes for a desired reaction or an entire CHEMKIN input file. The thermodynamic properties which are calculated are ΔUr {ΔEr}, ΔHr, ΔSr, ΔGr, and Af/Ar: ratio forward to reverse Arrhenius A factors. All species appearing in a reaction must be defined in a database similar to Figure 3.2. Property changes are estimated as a function of temperature over a range specified by the user. This is useful for kineticists wishing to check thermodynamic and kinetic consistency of reaction mechanisms being developed.

Some important aspects of this program are heat capacity extrapolation to 5000 K, thermo property polynomial determination, and calculation of thermodynamic property changes for reaction.

More examples of the use of THERM appear in appendix III-C, the THERM Users Manual. Extensive source code listings are given in appendicies V-A to V-F for all fortran programs and subroutines which constitute the THERM package.

REFERENCES SECTION III

1. Benson, S., Thermochemical Kinetics, John Wiley and Sons, New York (1976).
2. Benson, S.W. and J. H. Buss, J. Chem. Phys.,29,546 (1958).
3. Benson, S.W., F. R. Cruickshank, D. M. Golden, G. R. Haugen, H. E. O'Neal, A. S. Rogers, R. Shaw, and R. Walsh, Chem. Rev., 69, 279 (1969).
4. Reid, R.C., J. M. Prausnitz, and T. K. Sherwood, Properties of Gases and Liquids, McGraw Hill Book Co, New York (1977).
5. Kee, R.J., Miller, J.A., Jefferson, T. H., Sanida Report # SAND80-8003, 1980.
6. Cohen, N., Aerospace Report No. ATR-88(7073)-2, Aerospace Corporation, El Segundo, California (1988).
7. Hoffmann, J. M. and D. C. Maser, eds., ACS Symposium Series 274 : Chemical Hazard Process Review, American Chemical Society, Washington, D.C. 1985.
8. Stein, S. E., D. M. Golden, and S. W. Benson, J. Phys. Chem., 81, 4, 1977.
9. Shaw, R., D. M. Golden, S. W. Benson, J. Phys. Chem, 81, 18, 1977.
10. Stein, S. E., D. M. Golden, and S. W. Benson, J Phys Chem., 81, 4, 1977.
11. Eigenmann, H. K., D. M. Golden, and S. W. Benson, J Phys. Chem, 77, 13, 1973.
12. Stein, S. E. and Fahr, A., J. Phys. Chem. 1985, 89, 3714.
13. Patel, S., E. R. Ritter, and J. W. Bozzelli, manuscript in preparation, 1989
14. This work.
15. Domalski, E. S. and E. D. Hearing, J. Phys. Chem. Ref. Data, 17, 4, 1988.
16. Freedman, E. and W. H. Seaton, USA Ballistic Research Laboratories Memorandum Report No. 2320, Aberdeen Proving Ground, Maryland, 1973.
17. Frurip, D. J., E. Freedman, and G. R. Hertel, Proceedings of the Int. Symp. on Runaway Reactions, Cambridge, Mass. 1989.
18. Muller, C., G. Scacchi, and G. M. Come, AICHE 77th Annual Meeting, San Francisco, 1984.
19. McMillian, D. F. and Golden, D. M., Ann. Rev. Phys. Chem. 1982, 33,493.
20. Griller, D. and Castelhano, A.L., J. Am. Chem. Soc., 1982, 104, 3655.
21. Griller, D. and Lossing, F. P., J. Am. Chem. Soc., 1981, 103, 1586.
22. Holmes, J.L., Lossing, F. P.., and Maccoll, A., preprint submitted to J. Am. Chem. Soc., 1988.
23. Tsang, W., J. Am. Chem. Soc., 1985, 107, 2872.
24. Slagle, I. and Gutman, D., J. Am. Chem. Soc., 1988, 110, 3084.
25. Slagle, I. and Gutman, D., J. Am. Chem. Soc., 1988,

110, 3092.

26. Burcat, A., Zeleznik, F.J., and McBride, B.J., NASA Technical Memorandum # 83800, 1985.

27. Hanson, R. J. and K. H. Haskell, Sandia National Laboratories Report, SAND77-0552, 1978.

28. Hanson, R. J. and K. H. Haskell, Sandia National Laboratories Report, SAND78-1290, 1979.

29. McBride, B. J. and S. Gordon, NASA Technical Memo: NASA TN-D-4097, 1967.

30. Stull, D.R., Westrum, E.F.Jr., and Sinke, G.C., THE CHEMICAL THERMODYNAMICS OF ORGANIC COMPOUNDS, Robert E. Krieger Publishing Company, Malibar, Florida (1987).

31. Kee, R.J., Rupley, F.M., and Miller, J.A., Sandia National Laboratories Report, SAND87-8215, 1987.

32. Cox,J.D., Pilcher, G., Thermochemistry of Organic and Organometalic Compounds, Academic Press, New York, 1970.

APPENDIX III-A

TABLES for SECTION III

TABLE III.i

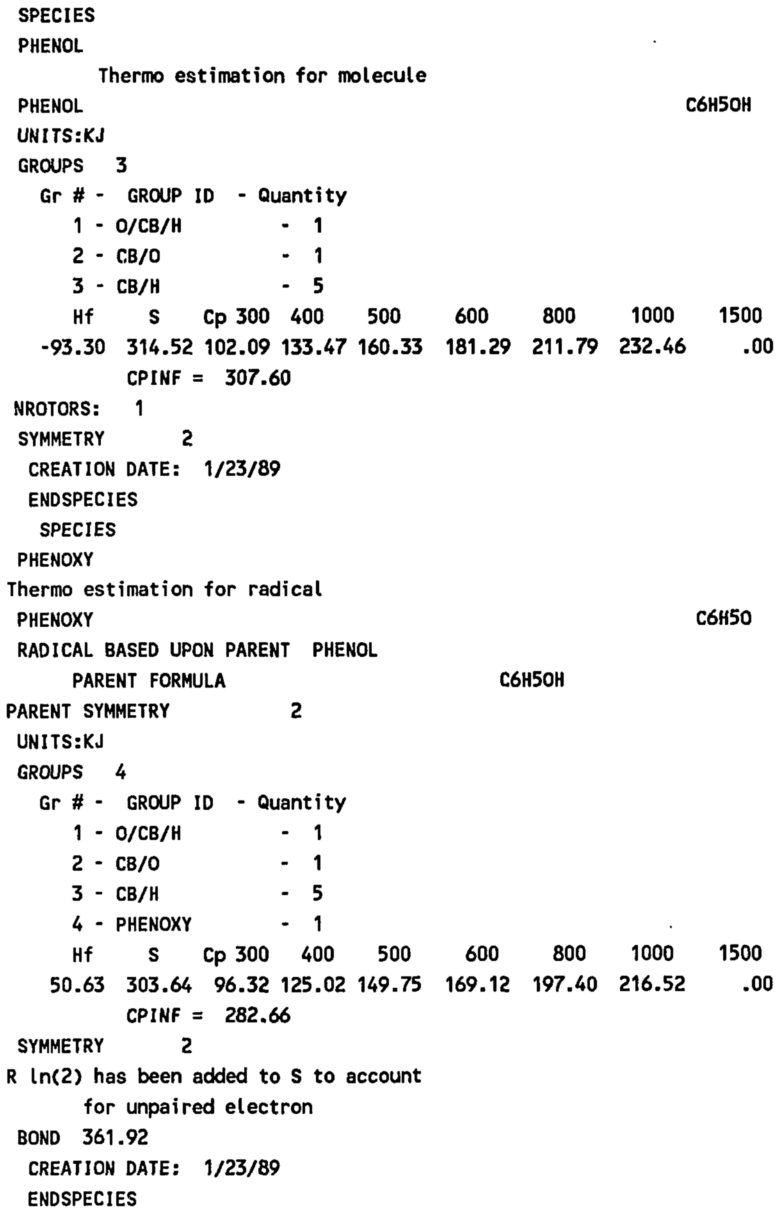
formula	species	Hfo(g) [kcal/mol] calc	exp	deviation	ref
C5H6	cyclopentadiene	32.4	31.94	0.46	32
C7H8	toluene	11.81	11.99	-0.18	"
C8H8	styrene	35.22	35.30	-0.08	"
C8H10	ethyl benzene	6.95	7.15	-0.2	"
C9H8	indene	40.42	39.08	1.34	"
C9H10	indane	14.74	14.42	0.32	"
C9H12	n-propyl benzene	2.09	1.89	0.2	"
C9H12	i-propyl benzene	0.63	0.96	-0.33	"
C10H8	azulene	74.92	73.5	1.42	"
C10H8	naphthalene	36.0	36.05	-0.05	"
C12H10	acenaphthane	36.65	37.4	-0.75	"
C14H10	anthracene	52.2	55.2	-3.0	"
C14H10	phenanthrene	50.0	49.52	0.48	"
C16H10	fluoranthene	68.39	69.78	-1.39	"
C16H10	pyrene	55.20	51.59	3.61	"
C18H12	triphenylene	61.80	63.4	-1.6	"
C2H6O	dimethyl ether	-43.40	-43.99	0.59	30
C6H7N	aniline	20.80	20.76	0.04	"
C2H5Cl	chloroethane	-26.70	-26.70	0.0	"
C6H5Cl	chlorobenzene	12.70	12.39	0.31	"
C6H5F	fluorobenzene	-26.30	-27.86	1.56	"
C6H5I	iodobenzene	39.00	38.85	0.15	"
C7H8	toluene	11.81	11.95	-0.14	"
C6H6O	phenol	-22.30	-23.03	0.73	26
C6H5O	phenoxy radical	12.10	11.40	0.70	"
C12H10	biphenyl	42.92	43.53	-0.61	"

TABLE III.ii

SPECIES	# ROTORS	SYMMETRY NUMBER
CH_4	0	12
CH_3Cl	0	3
C_2H_6	1	18
C_3H_8	2	18
C_2H_5Cl	1	3
biphenyl	1	8
o,m-chlorobiphenyl	1	2
benzene	0	12
chlorobenzene	0	2
toluene	1	6
m-terphenyl	2	6
p-terphenyl	2	16

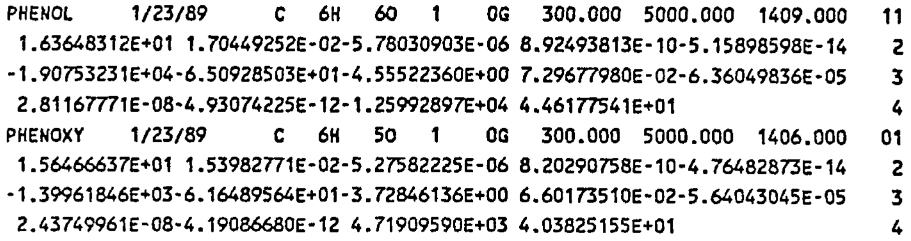
APPENDIX III-B

FIGURES for SECTION III



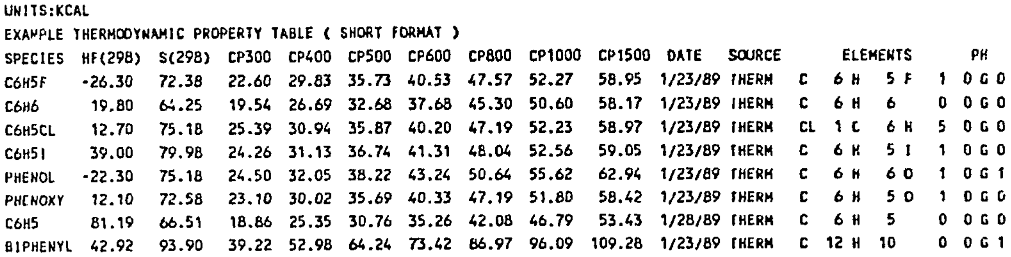
```
 SPECIES
 PHENOL
        Thermo estimation for molecule
 PHENOL                                                  C6H5OH
 UNITS:KJ
 GROUPS   3
   Gr # -   GROUP ID  - Quantity
      1 - O/CB/H        -  1
      2 - CB/O          -  1
      3 - CB/H          -  5
      Hf     S     Cp 300  400    500     600     800     1000    1500
   -93.30  314.52 102.09 133.47 160.33  181.29  211.79  232.46     .00
           CPINF =  307.60
NROTORS:   1
 SYMMETRY       2
  CREATION DATE:  1/23/89
  ENDSPECIES
   SPECIES
 PHENOXY
Thermo estimation for radical
 PHENOXY                                                  C6H5O
 RADICAL BASED UPON PARENT  PHENOL
      PARENT FORMULA                          C6H5OH
PARENT SYMMETRY           2
 UNITS:KJ
 GROUPS   4
   Gr # -   GROUP ID  - Quantity
      1 - O/CB/H        -  1
      2 - CB/O          -  1
      3 - CB/H          -  5
      4 - PHENOXY       -  1
      Hf     S     Cp 300   400    500     600     800     1000    1500
    50.63  303.64  96.32 125.02 149.75  169.12  197.40  216.52     .00
           CPINF =  282.66
 SYMMETRY       2
R ln(2) has been added to S to account
       for unpaired electron
 BOND  361.92
  CREATION DATE:  1/23/89
  ENDSPECIES
```

Figure 3.1.



```
PHENOL      1/23/89       C   6H   60   1    0G   300.000  5000.000  1409.000   11
 1.63648312E+01 1.70449252E-02-5.78030903E-06 8.92493813E-10-5.15898598E-14    2
-1.90753231E+04-6.50928503E+01-4.55522360E+00 7.29677980E-02-6.36049836E-05    3
 2.81167771E-08-4.93074225E-12-1.25992897E+04 4.46177541E+01                   4
PHENOXY     1/23/89       C   6H   50   1    0G   300.000  5000.000  1406.000   01
 1.56466637E+01 1.53982771E-02-5.27582225E-06 8.20290758E-10-4.76482873E-14    2
-1.39961846E+03-6.16489564E+01-3.72846136E+00 6.60173510E-02-5.64043045E-05    3
 2.43749961E-08-4.19086680E-12 4.71909590E+03 4.03825155E+01                   4
```

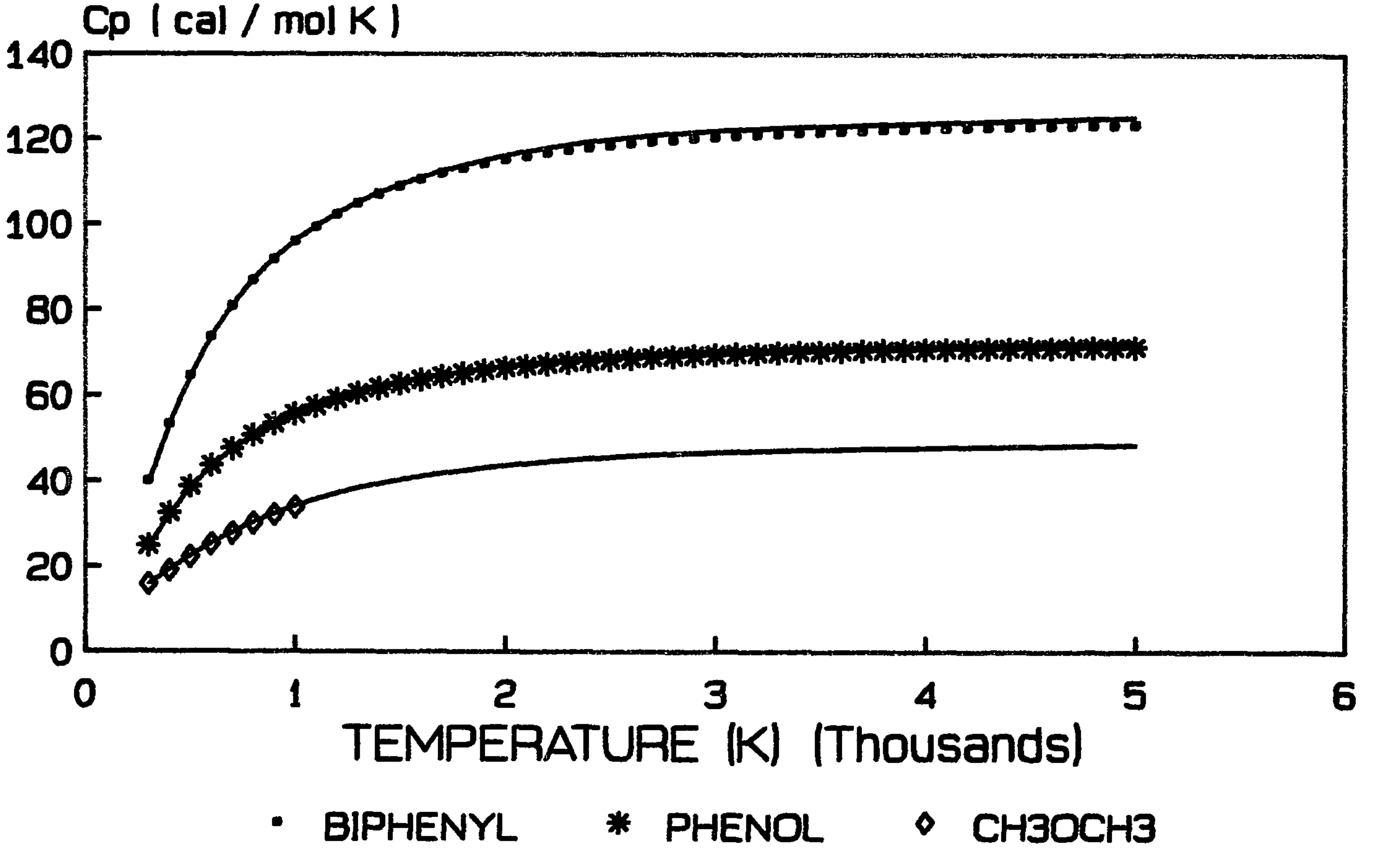
Figure 3.2.



```
UNITS:KCAL
EXAMPLE THERMODYNAMIC PROPERTY TABLE ( SHORT FORMAT )
SPECIES  HF(298)  S(298)  CP300  CP400  CP500  CP600  CP800  CP1000  CP1500  DATE    SOURCE      ELEMENTS        PH
C6H5F    -26.30   72.38   22.60  29.83  35.73  40.53  47.57  52.27   58.95  1/23/89 THERM   C    6 H   5 F   1  0 G 0
C6H6      19.80   64.25   19.54  26.69  32.68  37.68  45.30  50.60   58.17  1/23/89 THERM   C    6 H   6     0  0 G 0
C6H5CL    12.70   75.18   25.39  30.94  35.87  40.20  47.19  52.23   58.97  1/23/89 THERM   CL   1 C   6 H   5  0 G 0
C6H5I     39.00   79.98   24.26  31.13  36.74  41.31  48.04  52.56   59.05  1/23/89 THERM   C    6 H   5 I   1  0 G 0
PHENOL   -22.30   75.18   24.50  32.05  38.22  43.24  50.64  55.62   62.94  1/23/89 THERM   C    6 H   6 O   1  0 G 1
PHENOXY   12.10   72.58   23.10  30.02  35.69  40.33  47.19  51.80   58.42  1/23/89 THERM   C    6 H   5 O   1  0 G 0
C6H5      81.19   66.51   18.86  25.35  30.76  35.26  42.08  46.79   53.43  1/28/89 THERM   C    6 H   5     0  0 G 0
BIPHENYL  42.92   93.90   39.22  52.98  64.24  73.42  86.97  96.09  109.28  1/23/89 THERM   C   12 H  10     0  0 G 1
```

Figure 3.3.

Figure 3.4 : Cp (T)
COMPARISON OF ESTIMATES with LITERATURE



APPENDIX III-D

THERM USERS MANUAL

THERM

THermodynamic property

Estimation

for

Radicals and Molecules

USER'S MANUAL

March 16,1987

by

Edward R. Ritter and Joseph W. Bozzelli

Department of Chemical Engineering, Chemistry,
and Environmental Science

New Jersey Institute of Technology
Newark, NJ. 07102

Questions or requests for copies of this program should be addressed to:

Joseph W. Bozzelli
Chemical Engineering Dept.
New Jersey Inst. of Tech.
323 King Blvd.
Newark, NJ. 07102

Copies of this program and documentation are available on disk free of charge. To obtain a copy of this software and documentation, send one of the following to the above adress:

A. 1 - 1.2 Mb floppy disk (5.25")

B. 1 - 720 kb floppy disk (3.5")

C. 2 - 360 kb floppy disks (5.25")

Make sure the disk (s) is formatted and contains the MS-DOS system (*command.com, etc.*). *Your disk (s) will be returned with* THERM and this manual installed.

Contact the authors at the above address for further details.

This program may be copied freely, however, we request that users notify us so that we may keep them informed of any improvements or problems with this program.

Please contact us if you have any questions, comments, or problems with this program package.

E. R.

THERM

THermo Estimation for Radicals and Molecules

System Requirements:

- IBM PC / AT or COMPATIBLE
- 512 K available memory
- 1 - 1.2 Mb (5.25 in.), or 720 kb (3.5 in.), or 2 - 360 K floppy drives (hard disk may be used if all files are copied into a subdirectory)
- 8087 or 80287 "math processor"
  NOTE: the program will not run if the computer does not have a math processor.

Programs included in this package:

EXE files:

- THERM.EXE - therm main program.
  group estimation procedure.
- THERMFIT.EXE - thermo fitting procedure:
  fit Cp data to polynomials and output in NASA format.
  input file : list file "*.LST"
  output file: polynomial file ".DAT"*
- THERMLST.EXE - thermo listing procedure:
  create a thermo table from polynomials.
  input file : polynomial file "*.DAT"
  output file: list file "*.LST"
- THERMRXN.EXE - reaction analysis procedure:
  calculate ΔHr, ΔSr, ΔU, (ΔE), ΔGr, and ratio Arrhenius A factors forward to reverse (Af/Ar).
  CHEMKIN input files can be processed (only gas phase molecules and radicals are supported:
  no ions, liquids, or solids !)
- STOD.EXE - sequential to direct file conversion utility. THERM requires polynomial files "*.DAT" and documentation files "*.DOC" to be formatted to specific record lengths (DIRECT ACCESS FORMATTED). DAT files require 80 character records while DOC files require 70 character records. In sequential files, record length varies from line to line, therefore if one edits a file using a word processor this utility may be used to pad each record to the required length. All editing must be done as ASCII files w/ no formatting.

EXAMPLE SPECIES DATABASE FILES:

* THERMO.DOC - sample documentation file "*.DOC".
* THERMO.LST - sample database listing file "*.LST".
* THERMO.DAT - sample nasa format database listing "*.DAT".

GROUP DATABASE FILES:

* HC. - hydrocarbon groups.
* HCO. - oxygen containing groups.
* CDOT. - radical groups.
* BD. - bond dissociation increment groups.
* CLC. - Chlorinated hydrocarbon groups.
* CYCH. - ring corrections.
* HCN. - nitrogen containing groups.

BOOTING UP the THERM program:

To boot up THERM place the disk marked "THERM PROGRAM DISK: DRIVE A:" in drive A and then reboot your computer ("Ctrl-Alt-Delete"). At the prompt for drive A:, type "THERM" and hit return. The program title page should appear with the message:

```
      ***Loading Configuration File***
     ***LOADING GROUP DATABASE FILES***
```

When all files are loaded, the number of groups in the database is displayed. Hit return and you advance to the main menu:

```
          THERM MAIN MENU.
```

SAMPLE SESSION: Calculate and file thermodynamic properties of PHENOL and PHENOXY RADICAL.

note: Help is available for most input if "?" is entered instead of the requested input. The letter "Q" (upper or lower case) is a reserved character used to "QUIT" back to a previous menu or point in the program. As a result the first character of a species ID

should not begin with either of these characters (Q or ?).
SHORT SAMPLE SESSION:

In the following paragraphs, quote marks are used for emphasis only. You should only enter what is inside the quote marks.

1. At the main menu prompt hit return to accept the default option "1 - ENTER/ESTIMATE SPECIES".

2. You are now asked to "enter species name (ID)". Enter "phenol" followed by a return (upper or lower case are accepted; but THERM converts all strings to capital letters).

3. At this point you are asked to enter the "elemental formula" for the species. Enter "C6H5OH" or "C6H6O" whichever you prefer.

4. You are now asked to verify that the formula was interpreted correctly. Hit return if correct or enter "Q" if it was not.

5. After a brief description of how THERM calculates radicals, you are asked to enter the number of groups in this species. Phenol is comprised of 5 - CB/H groups, 1 - CB/O group, and 1 - O/CB/H group. This is a total of 7 groups. Enter "7" at this prompt.

If one wants to enter literature data instead of estimating thermo properties using Benson's method, they should simply enter "0" when prompted for the number of groups in the species. THERM will then prompt for literature data to be input.

6. You are now asked to enter the number of different groups which are present in this species. Above we see that PHENOL only has 3 different groups. Enter "3" at this prompt.

7. Enter the number of rotors (tops) in the species. Default is no rotors. The number of rotors is used to adjust the high temperature limiting heat capacity. Phenol has 1 rotor (-OH), enter 1 and press return.

8. It is now time to enter the groups which are present in phenol. Enter the following:

```
        "CB/H,5"              (note 1: a blank or comma are the
                                same. "CB/H 5" is equivalent.)
        "CB/O"                (note 2: if no quantity is entered
                                quantity "1" is understood)
        "O/CB/H"
```

In group names, slashes delimit atoms or functional groups. The letters appearing before the first slash represent the central atom or next nearest neighbor (or other) correction (ex: C/, O/,or /N (central atom) or CIS/, ORT/, /MET, GAUCHE/, or OI/ (corrections)).

The atoms or functional groups that follow are each delimited with slashes "/" and appear in alphabetic order (ex: Bensons

O-(H)(C) would be O/C/H, C-(Cd)(C)2(H) would be C/C2/CD/H2, AND C-(Cd)(CB)(H)2 would be C/CB/CD/H2).

9. The symmetry number for phenol is 2. Enter "2" at this prompt.

The SPECIES SCREEN appears showing all data entered in addition to the thermodynamic properties which have been calculated. An EDIT option is available; enter "?" for more details (we will not edit at this point).

10. Hit return to advance to the SPECIES ESTIMATION OPTIONS MENU.

11. Choose option "F" to add this species to a file.

12. You are asked for a filename. Enter "B:THERMO".

13. Since "B:THERMO.LST" already exists, the following message appears:

```
          THERMO.LST already exists:

          1 - DELETE (OVERWRITE)
          2 - APPEND (ADD TO THE END OF THE FILE)
          3 - ENTER A DIFFERENT FILENAME

          ENTER OPTION:
```

We want to add Phenol to the end of this file so enter "2".

14. Similarly "B:THERMO.DOC" already exists. Again enter "2" and the data for phenol will be added to the end of these files.

You now have returned to the SPECIES ESTIMATION OPTIONS MENU. Note that you are notified that PHENOL is the last species that you have filed. Now that these files are connected you will not be asked to input the filename when filing. To close or change output files "Q" quit back to the MAIN menu and choose option "*".

15. One can now calculate the properties of PHENOXY radical. To do this choose option "R". The RADICAL ESTIMATION MENU appears listing all bond dissociation "BD" increments which are defined. For a description of these increments enter "?" and then the increment group name (not the group number) when prompted. We are interested in the "PHENOXY" bond dissociation increment. Enter "PHENOXY" when you are done looking at the help on these groups.

16. You must now enter the symmetry number for PHENOXY radical. Enter "2" at this prompt. Had the symmetry number been 1 you could have just hit return since if no symmetry number is entered "1" is assumed.

17. Now enter the name for your radical (9 characters maximum). Enter "PHENOXY" or "C6H5O." whichever you prefer.

18. *The formula for phenoxy is C6H5O; so enter "C6H5O".*

19. Hit return to confirm that the elemental composition is correct.

20. The species screen now displays the data for PHENOXY radical. Hit return to advance to the SPECIES ESTIMATION OPTIONS MENU.

21. Choose option "F" and PHENOXY's data is added to the current file (since the file is already initilized you are not asked for any additional input). You return to the SPECIES ESTIMATION OPTIONS MENU automatically and are notified that PHENOXY is the last species which you filed.

22. At this point you can review the help which is available on the other options. Entering "M" will return you to PHENOL where you could use the edit option to change/add groups to estimate a substituted phenol for example meta-ChloroPhenol.

Note: Phenoxy radical could have been calculated directly (without calculating PHENOL first) if you had used the RADICAL GROUP "O/CB/." when first entering groups. Both methods of radical calculation are supported, however, use of RADICAL GROUPS in the case of resonantly stabilized allylic species may result in a significant error. These groups do not address the resonance stabilization energy accounted by BD BOND INCREMENT GROUPS.

For a more detailed example see SAMPLE SESSION.

THERM CONFIGURATION: The "THERM.CFG" file

```
#FILES 7                   the number of group files
BD.
HC.
CYCH.
CLC.                           GROUP FILE NAMES
HCO.
HCN.
CDOT.
#ELEMENTS 11               the number of elements defined
C
H
O
CL                                  ELEMENTS
S
N
F
P
I
BR
B
#HELP 8                       the number of help files
THM1P1.HLP
THM1P2.HLP
THS1.HLP
THM2P1.HLP                        HELP FILE NAMES
THM2P2.HLP
THM3.HLP
THE1.HLP
THE2.HLP
OPTIONS:                        other values alowed:
THERMFIT.EXE                    THERMFIT.EXE D:   where D: is the disk
THERMRXN.EXE                    THERMRXN.EXE D:   drive where this
THERMLST.EXE                    THERMLST.EXE D:   EXE file resides
#KEYS 3                         The number of keys used in sorting
UNITS:KCAL                      UNITS:KJ    ( default to KJ units)
ERRORCK=ON                      ERRORCK=OFF  (defeat error checking)
TRANGE=5000                     could also be TRANGE=2000
```

MAIN MENU - options and explanations:

```
                        THERM MAIN MENU

   Units:  Kcal/mol  ( cal/mol K )                  3/ 5/89
 ____________________________________________________________

 1 - ENTER/ESTIMATE SPECIES     5 - REVIEW (LST FORMAT FILE)
 2 - VIEW GROUPS                6 - GET SPECIES FROM FILE {*.DOC}
 3 - SELECT SHORT MENU          7 - ReWrite THERMO FILES {*.DOC, *.LST}
 4 - SELECT LIST FILE {*.LST}   8 - RUN THERMFIT create NASA {*.DAT}
          9 - RUN THERMRXN (calculate dU, dH, dS, dG for reaction)
         10 - Sort & ReWrite THERMO FILES { *.DOC, *.LST }
         11 - Sort/Write *.LST file { no *.DOC is needed }
         12 - RUN THERMLST (LIST NASA FORMAT FILE) /create *.LST
         13 - DOS utilities
          U - CHANGE CURRENT UNITS TO : KJ/mol  (J/mol K)
                   *  -  CLOSE ALL OPENED FILES
                   X  -  EXIT PROGRAM
                   ?  -  HELP

                     enter option : { 1 }
```

The following pages describe each option listed above.

OPTION 1:

1 - ENTER/ESTIMATE SPECIES

This option invokes THERM'S code to calculate thermodynamic properties using Benson's group additivity method.

To estimate molecular or radical species you will be asked for specific information about the species, such as:

* Species ID { limited to 9 characters so as to fit CHEMKIN format}

* # of groups in the species

* # of different groups to be entered (DO NOT INCLUDE BD GROUPS)

* Elemental formula { limited to 70 characters }

* Groups contained in the species

* species symmetry number

OPTION 2:

2 - VIEW GROUPS

This option allows you to review the group database currently in memory. This is useful if you are looking for a group, but have forgotten its name.

OPTION 3:

3 - SELECT SHORT/LONG MENU

This option toggles the amount of information displayed when a menu is on the screen.

The MAIN menu is not affected by this option.

LONG menu shows all information

SHORT menu only lists available options with no description

In both cases this help information is available.

OPTION 4:

4 - CHANGE SPECIES LIST FILE {*.LST}

THIS OPTION IS USED TO CHANGE THE "LST" FILE CURRENTLY IN MEMORY.

THERE ARE A MAXIMUM OF 300 SPECIES ALLOWED AT ANY TIME.

THIS DATA IS NOT PASSED TO THE REACTION ANALYSIS PROCEDURE BUT IS USED FOR SORTING AND REWRITING THE DATABASE.

THE "LST" FILE CONTAINS THE SPECIES ID (9 CHARACTERS MAX); Hf, S, Cp 300, Cp 400, Cp 500, Cp600, Cp 800, Cp 1000, Cp 1500; ELEMENTAL COMPOSITION; PHASE (G = gas), AND CAN HAVE SOME ABREVIATED REFERENCE INFORMATION. THE "LST" FILE IS THE INPUT FOR THE NASA FORMAT DATABASE GENERATION PROCEDURE "THERMFIT".

OPTION 5:

5 - LIST THERMO D BASE FILE {*.LST}

THIS OPTION ALLOWS ONE TO REVIEW THE CURRENT "LST" FILE.

YOU CAN NOT ONLY REVIEW SPECIES NAMES BUT CAN REQUEST EXTENDED OUTPUT SUCH AS:

Hf, S, Cp 300, Cp400, Cp 500, Cp600, Cp 800, Cp 1000, Cp 1500

A DELETE/UNDELETE OPTION IS AVAILABLE IN CASE YOU HAVE DUPLICATE OR INCORRECT RECORDS.

THE "LST" FILE MUST BE REWRITTEN USING OPTION " 10 or 11" IN ORDER TO DELETE SPECIES FROM THE FILE.

OPTION 6:

6 - GET SPECIES FROM DOCUMENTATION FILE {*.DOC}

THIS OPTION REQUIRES YOU TO HAVE A DOCUMENTATION FILE WHICH CONTAINS ALL PERTINENT INFORMATION ABOUT THE SPECIES.

THE DOCUMENTATION FILE CONTAINS:

- SPECIES ID & ELEMENTAL FORMULA
- ALL GROUPS
- THERMO DATA
- BOND DISSOCIATION INCREMENT (IF A RADICAL)
- SYMMETRY CORRECTION
- ANY TEXT ADDED FOR REFERENCE
- ANY ALTERATIONS MADE TO A CALCULATED VALUE (EDIT OPTION) ETC.

The user is required to enter a species name. THERM then reads into memory all data from a DOC file. The species can be edited or modified at this point just as if the user had typed in the data.

OPTION 7:

7 - ReWrite THERMO FILES {*.DOC, *.LST}

THIS OPTION RECALCULATES THERMO PROPERTIES. IT IS USEFUL IF ANY CHANGES HAVE BEEN MADE TO GROUP VALUES.
THE DOCUMENTATION FILE IS READ FOR GROUPS AND THE THERMO PROPERTIES ARE RECALCULATED USING THE CURRENT GROUP VALUES.

BE WARNED: ON AN "AT" CLASS MACHINE (10 MHz) 100 SPECIES WILL REQUIRE ABOUT 5 MINUTES FOR REGENERATION OF THE DATABASE.

A "PC" CLASS MACHINE W/ ONLY FLOPPY DRIVES WILL TAKE ~15 MINUTES.

OPTION 8:

8 - RUN THERMFIT: CREATE NASA FORMAT FILE {*.DAT}

This option spawns THERM'S heat capacity extrapolation/polynomial fitting program : THERMFIT.

"LST" file format is used as input to the program "THERMFIT.EXE".

This program converts thermo data to polynomials in NASA format for use with CHEMKIN or other codes which require this format and allow one to specify a common temperature other than 1000 K.

Two sets of polynomials are generated with a common (breakpoint) temperature determined so as to match Cp and dCp/dT. This break point is typically not the more common 1000 K as with SANDIA'S fitting code.

OPTION 9:

9 - RUN THERMRXN: REACTION ANALYSIS (del H, del U, del S, del G)

This program allows one to enter reaction expressions such as:

```
C6H5CL=CYC6H5. + CL
```

and have property change of reaction calculated from NASA format polynomials in addition to the ratio of forward/reverse A factors.

Data are taken from "DAT" files and one is prompted for a *.DAT filename. {OBVIOUSLY ALL SPECIES ID'S ENTERED IN THE REACTION EXPRESSION MUST APPEAR IN THE SPECIFIED "DAT" FILE }.

Entire CHEMKIN input files can also be processed automatically.

OPTION 10:

10 - Sort, Recalculate, & ReWrite THERMO FILES { *.DOC, *.LST }

This option is essentially the same as OPTION 7 except the "DOC" file is first sorted in order of elemental composition.

Sorting is performed on the # of levels defined in the configuration file "THERM.CFG". The key word "#KEYS" is followed by the number of sort keys (or levels). The priority for sorting is determined by the order elements appear in "THERM.CFG" (configuration file).

For example: if the configuration file contains the following elements in the following order (each element actually appears on a separate line):

```
C  H O  CL N BR   (and #KEYS 2 )
```

Data is first sorted in order of increasing carbon number; then each sub group is sorted by increasing hydrogen number. If #KEYS 3 appears in the file "THERM.CFG" then sorting would continue by oxygen number. If the order elements appear in "THERM.CFG" is changed, so is the order of sorting.

Literature data are not recalculated.

OPTION 11:

11 - SORT/REWRITE *.LST FILE

Same sorting procedure as OPTION 10; but DATABASE RECALCULATION is not performed.
This option is useful if one wants to sort a "LST" file but has no documentation file to recalculate the database.

OPTION 12:

12 - RUN THERMLST: Uses NASA format "DAT" file to create "LST" format

This option spawns THERM'S listing program : THERMLST.EXE

A *.LST FORMAT FILE IS CREATED FROM A NASA FORMAT *.DAT FILE. COEFFICIENTS ARE READ AND HEAT CAPACITY VALUES ARE GENERATED AT 300, 400, 500, 600, 800, 1000, 1500 deg K. (An extended listing to 5000 K may also be created.)

OPTION 13:

13 - DOS UTILITIES

ALLOWS ONE TO INVOKE THE DOS INTERPRETER FROM WITHIN THE PROGRAM. SEVERAL OPTIONS ARE AVAILABLE. CHOOSE THIS OPTION (13) FOR MORE DETAILS.

OPTION U:

U - CHANGE UNITS

This option toggles the units which results are displayed on screen and saved in documentation files. Units can either be Kcal/mol (Hf) and cal/mol K (S and Cp) or KJ/mol (Hf) and J/mol K (S and Cp).

This has no effect on units saved in a "LST" file. If one adds data to a "LST" file, units remain the same as those already in use in that file. If one starts a new "LST" file they will be prompted for units.

Units may be changed at various other parts of the program.

OPTION *:

* - CLOSE ALL OPENED FILES

This option is necessary to change the ouput or other files that are connected.
All files are closed automatically when exiting THERM as below.

OPTION X:

X - EXIT PROGRAM
ENDS THERM SESSION , CLOSES ALL FILES, AND RETURNS TO DOS.

```
EDIT OPTIONS ARE AVAILABLE:    for both MOLECULES and RADICALS


EDITING A MOLECULE


Thermo estimation for molecule
BZ                                                              C6H6


  Gr # -  GROUP ID  - Quantity
     1 - CB/H          -  6

UNITS:  Hf{Kcal/mol}   S & Cp {cal/mol K}
   Hf       S     Cp 300   400      500     600      800     1000     1500
  19.80   64.24   19.44   26.64   32.76   37.80   45.24   50.46   58.38
     CPINF    67.56
Symmetry number used to correct S value is         12
  EDIT THESE RESULTS?    "Y" OR {"N"}
Y  <-------------------------------- to edit enter "Y" at this prompt.

====================== next screen ========================================


                              EDIT SPECIES MENU
                         ________________________________
(this menu for
molecules only)          1  -  CHANGE SYMMETRY NUMBER
                         2  -  CHANGE/MODIFY THERMO
                         3  -  CHANGE SPECIES ID / FORMULA
                         4  -  ADD TEXT TO SPECIES SCREEN
                         5  -  RETURN TO SPECIES SCREEN
                         6  -  ADD A GROUP TO THE CURRENT SPECIES
                         7  -  DELETE A GROUP
                         8  -  CHANGE A GROUP
                         9  -  CHANGE NUMBER OF ROTORS
                         ?  -  HELP
                              enter option : { Q }
```

EDITING A RADICAL:

```
Thermo estimation for radical
PHENYL                                                    C6H5
RADICAL BASED UPON PARENT  BZ
                                              C6H6
PARENT SYMMETRY         12

  Gr # -  GROUP ID  - Quantity
     1 - CB/H          -  6
     2 - BZ            -  1

UNITS:  Hf{Kcal/mol}   S & Cp {cal/mol K}
   Hf      S     Cp 300   400      500      600      800     1000     1500
  81.35   68.92   18.57   25.03    30.54    35.09    41.83    46.62    53.88
     CPINF    61.60
Symmetry number used to correct S value is          2
R ln(2) has been added to S to account for unpaired electron
bond dissociation energy used=      113.650000
  EDIT THESE RESULTS?    "Y" OR {"N"}
Y   <---------------------------------  to edit enter "Y" at this prompt.

======================= next screen ========================================

                                   EDIT SPECIES MENU
                            _________________________________
(this menu for
radicals only)          1  -  CHANGE SYMMETRY NUMBER
                        2  -  CHANGE/MODIFY THERMO
                        3  -  CHANGE SPECIES ID / FORMULA
                        4  -  ADD TEXT TO SPECIES SCREEN
                        5  -  RETURN TO SPECIES SCREEN
                        6  -  CHANGE BOND DISSOCIATION ENERGY
                        7  -  RETURN TO PREVIOUS PARENT MOLECULE
                        8  -  CHANGE NUMBER OF ROTORS
                        ?  -  HELP
                           enter option : { Q }
```

THE FOLLOWING PAGES GIVE BRIEF DESCRIPTIONS OF EACH OPTION

EDIT OPTIONS EXPLAINED:

OPTION 1:

1 - CHANGE SYMMETRY NUMBER

THIS OPTION ALLOWS YOU TO CORRECT/MODIFY THE SYMMETRY NUMBER WHICH YOU PREVIOUSLY ENTERED. UNLIKE IN THE ESTIMATION PROCEDURE A BLANK IS INTERPRETED AS NO CHANGE.

OPTION 2:

2 - CHANGE/MODIFY THERMO

THIS OPTION ENABLES YOU TO MAKE ANY SMALL ADJUSTMENTS TO THE CALCULATED THERMO VALUES. YOU CAN EITHER ADD A CORRECTION { (-)FOR NEGATIVE CORRECTIONS } OR REPLACE THE VALUE.

YOU CAN CHANGE Hf,S, OR A Cp VALUE (ONE AT A TIME).

ANY CHANGES YOU MAKE WILL BE RECORDED IN THE DOCUMENTATION FILE UNDER THE NAME "DELTA EDIT". THIS INSURES THAT LATER USE OF AUTOMATIC UPDATING PROCEDURES WILL CONSIDER THESE SMALL ADJUSTMENTS. IF AFTER CHANGING AND SAVING YOUR RESULTS, YOU DECIDE TO REMOVE THE "DELTA EDIT" YOU MUST DO SO OFF-LINE USING ANY CONVENIENT WORD PROCESSOR. YOU MUST EDIT THE DOCUMENTATION FILE AS AN ASCII FILE (no formatting) AND DELETE THE "DELTA EDIT" REFERENCE AND GROUP.

OPTION 3:

3 - CHANGE SPECIES FORMULA / ID

THIS OPTION ALLOWS YOU TO CHANGE THE FORMULA AND/OR SPECIES ID.

THIS IS USEFUL IF YOU CHANGE A GROUP TO FORM A NEW SPECIES RATHER THAN RE-ENTERING ALL GROUPS IN THAT SPECIES. IT IS ALSO USEFUL IF YOU MADE AN ERROR IN THE SPECIES NAME OR FORMULA. IF YOU WANT TO CHANGE THE SPECIES ID WITHOUT CHANGING THE FORMULA, ENTER A BLANK WHEN ASKED FOR THE NEW FORMULA (THE FORMULA WILL REMAIN UNCHANGED AND YOU WILL ADVANCE TO THE CHANGE SPECIES NAME PROMPT).

CHANGING A GROUP IS NOT ALLOWED FOR RADICALS ESTIMATED USING BOND INCREMENTS (BD groups). FOR MORE INFORMATION ON THIS OPTION REQUEST HELP ON OPTIONS 6, 7, OR 8.

OPTION 4:

4 - ADD TEXT TO SPECIES SCREEN

THIS OPTION ALLOWS YOU TO ENTER UP TO 50 LINES OF TEXT TO THE SPECIES SCREEN (DATA WILL SCROLL OFF THE SCREEN UNLESS YOU USE "Ctrl S" TO STOP SCROLL).

Text is stored in a DOC file.

TEXT IS USEFUL FOR ADDING ANY DOCUMENTATION YOU LIKE TO THE DOCUMENTATION FILE (EX: COMMON NAME).

OPTION 5: DEFAULT OPTION !

5 - RETURN TO THE SPECIES SCREEN:

"QUIT" RETURN TO THE PREVIOUS MENU
This is the default option.

OPTION 6:

6 - ADD A GROUP TO THE CURRENT SPECIES

THIS OPTION IS USEFUL IF YOU FORGOT A GROUP OR IF YOU ARE CREATING A NEW SPECIES BY EDITING THE GROUPS OF THE CURRENT SPECIES. THIS OPTION IS ONLY ENABLED FOR STABLE MOLECULES AND RADICALS WHICH WERE CALCULATED USING RADICAL GROUPS FROM "CDOT".

OPTION 7:

7 - DELETE A GROUP FROM THE CURRENT SPECIES

THIS OPTION IS USEFUL IF YOU ARE CREATING A NEW SPECIES BY EDITING THE CURRENT GROUPS. THIS OPTION IS ONLY ENABLED FOR STABLE MOLECULES AND RADICALS WHICH WERE CALCULATED USING RADICAL GROUPS "CDOT"; *not enabled for any BD increments.*

OPTION 8:

8 - CHANGE A GROUP

THIS OPTION ALLOWS YOU TO CREATE A NEW SPECIES BY EDITING THE GROUPS FROM THE PREVIOUS SPECIES. THIS IS THE MOST USEFUL OPTION OF THE THREE (6, 7, 8). YOU CAN CHANGE THE GROUP ID AND/OR QUANTITY.

EXAMPLE:
STARTING WITH BENZENE: "CB/H,6"; EDIT/ESTIMATE CHLOROBENZENE.
YOU CAN CHANGE THIS FROM "CB/H,6" TO "CB/H,5"
THEN YOU CAN SELECT OPTION "6" ADD A GROUP; AND ADD "CB/CL".
NOW YOU HAVE THE GROUPS FOR CHLOROBENZENE. ALL YOU HAVE TO DO IS CHANGE THE SYMMETRY NUMBER, FORMULA, AND SPECIES ID.

THIS MIGHT NOT SEEM LIKE IT SAVES WORK; BUT IF YOU HAVE A SPECIES WITH 10 OR 15 DIFFERENT GROUPS, AND ALL YOU NEED TO CHANGE IS ONE TO CREATE YOUR NEW SPECIES, THERE IS A SIGNIFICANT SAVINGS IN TIME OVER ENTERING ALL 10 - 15 GROUPS.

OPTION 9:

9 - Change the number of rotors in the species

This allows one to change the number of rotors which are considered to be free rotors at high temperature (5000 K).

Molecules with attached tops that have large barriers to internal rotation (V > 10 kcal/mol) cannot be considered free rotations even at high temperature.

The number of free rotors changes the limiting heat capacity Cp infinity, since each vibrational mode contributes 1 heat capacity unit while a free rotation only contributes 1/2 a heat capacity unit and a vibrational mode is lost when a free rotor exists.

OPTIONS FOR RADICAL SPECIES

OPTION 6 - RADICAL SPECIES:

6 - CHANGE BOND DISSOCIATION ENERGY FOR A RADICAL

THIS OPTION ALLOWS YOU TO ALTER THE BOND ENERGY FROM ONE OF THE BD GROUPS TO MEET YOUR NEEDS. A NEW GROUP CALLED "DELTA BOND ENERGY" IS CREATED AND ADDED TO THE DOCUMENTATION FILE.

AN EXAMPLE IS THE ESTIMATION OF THE CHLOROPHENYL RADICAL.

THE BOND ENERGY USED FOR LOSS OF H ATOM FROM BENZENE IS 113.5 Kcal. THAT FOR LOSS OF H ATOM FROM CHLOROBENZENE IS LOWER BY ABOUT 1 Kcal.

TO CREATE CHLOROPHENYL RADICAL YOU CAN APPLY THE "BZ" BOND INCREMENT GROUP TO CHLOROBENZENE AND THEN USE THIS OPTION TO CORRECT THE BOND ENERGY USED IN THE CALCULATION OF CHLOROPHENYL TO 112.5 Kcal.

OPTION 7 - RADICAL SPECIES:

7 - RETURN TO PREVIOUS PARENT MOLECULE

THIS OPTION IS THE SAME AS OPTION "M" FROM THE SPECIES ESTIMATION OPTIONS MENU. IT IS INCLUDED HERE FOR CONVENIENCE ONLY.

WHEN THIS OPTION IS CHOSEN ALL DATA ON THE RADICAL IS DISCARDED.

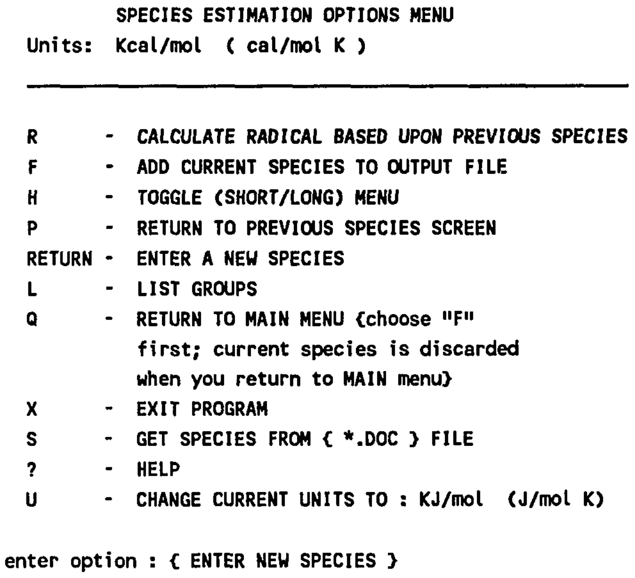
OPTION 8:

8 - Change the number of rotors in the species

This allows one to change the number of rotors which are considered to be free rotors at high temperature (5000 K).

Molecules with attached tops that have large barriers to internal rotation (V > 10 kcal/mol) cannot be considered free rotations even at high temperature.

The number of free rotors changes the limiting heat capacity Cp infinity, since each vibrational mode contributes 1 heat capacity unit while a free rotation only contributes 1/2 a heat capacity unit and a vibrational mode is lost when a free rotor exists.



```
        SPECIES ESTIMATION OPTIONS MENU
 Units:  Kcal/mol  ( cal/mol K )
 ____________________________________________

 R      -  CALCULATE RADICAL BASED UPON PREVIOUS SPECIES
 F      -  ADD CURRENT SPECIES TO OUTPUT FILE
 H      -  TOGGLE (SHORT/LONG) MENU
 P      -  RETURN TO PREVIOUS SPECIES SCREEN
 RETURN -  ENTER A NEW SPECIES
 L      -  LIST GROUPS
 Q      -  RETURN TO MAIN MENU {choose "F"
           first; current species is discarded
           when you return to MAIN menu}
 X      -  EXIT PROGRAM
 S      -  GET SPECIES FROM { *.DOC } FILE
 ?      -  HELP
 U      -  CHANGE CURRENT UNITS TO : KJ/mol  (J/mol K)

enter option : { ENTER NEW SPECIES }
```

OPTION R:

R - CALCULATE RADICAL

THIS OPTION ENABLES YOU TO ESTIMATE PROPERTIES OF A RADICAL SPECIES BY ADDING AN APPROPRIATE BOND DISSOCIATION INCREMENT TO THE CURRENT SPECIES.

BOND DISSOCIATION INCREMENTS ASSUME LOSS OF A HYDROGEN ATOM FROM THE PARENT SPECIES. TO ESTIMATE PHENYL RADICAL WE FIRST ESTIMATE BENZENE AND THEN APPLY THE "BZ" BOND INCREMENT TO THIS SPECIES.

R*ln(2) IS ADDED TO THE SPECIES ENTROPY TO ACCOUNT FOR THE FREE ELECTRON. BOND INCREMENT GROUPS ALSO INCLUDE ANY INTERNAL ENTROPY CHANGE. THE USER MUST SUPPLY THE SYMMETRY NUMBER FOR THE RADICAL, IN ADDITION TO IT'S NAME(ID) AND ELEMENTAL FORMULA.

OPTION F:

F - ADD THE CURRENT SPECIES TO A FILE

THIS OPTION SAVES THE CURRENT SPECIES IN THE "LST" FORMAT. IN ADDITION, THE DEFAULT IS TO CREATE A DOCUMENTATION FILE WHICH IS NECESSARY FOR AUTOMATIC UPDATE/RECALCULATION PROCEDURES AVAILABLE FROM THE MAIN MENU.

OPTION H:

H - TOGGLE SHORT/LONG MENU

TRY THIS OPTION TO SEE WHAT HAPPENS. CHOOSING "H" A SECOND TIME WILL RESTORE THE CURRENT MENU FORMAT.

OPTION P:

P - RETURN TO THE LAST SPECIES SCREEN

This option returns the user to the previous species screen.

OPTION M:

M - RETURN TO THE PREVIOUS PARENT MOLECULE

THIS OPTION IS ONLY ACTIVE IF THE LAST SPECIES SCREEN CONTAINED A RADICAL SPECIES. CHOOSING THIS OPTION RESTORES THE PARENT IF AND ONLY IF BOND INCREMENT GROUPS WERE USED TO CALCULATE THE RADICAL.

NOTE: CURRENT RADICAL SPECIES DATA IS DISCARDED WHEN THIS OPTION IS CHOSEN. THEREFORE, YOU SHOULD FILE "F" THIS SPECIES BEFORE CHOOSING OPTION "M".

END

OPTION L:

L - LIST GROUPS:

THIS OPTION ENABLES YOU TO REVIEW THE CURRENT GROUP FILE DATABASE.

OPTION Q:

Q - RETURN TO THE MAIN MENU

WHEN RETURNING TO THE MAIN MENU, ALL CURRENT SPECIES DATA ARE LOST. THEREFORE, ONE SHOULD FILE "F" THE CURRENT SPECIES BEFORE MAKING THIS CHOICE.

OPTION S:

S - GET SPECIES FROM { *.DOC } FILE

SAME FUNCTION AS OPTION "6" FROM THE MAIN MENU.

THIS OPTION ENABLES YOU TO RECALL/RECALCULATE A SPECIES BASED UPON INFORMATION CONTAINED IN THE DOCUMENTATION FILE.

THIS OPTION DISCARDS CURRENT SPECIES DATA. FILE "F" SHOULD BE CHOSEN FIRST TO SAVE THE CURRENT DATA.

END

```
                    RADICAL CALCULATION MENU
                 (radical by loss of H from parent)
              ____________________________________

              Q        -   RETURN TO PREVIOUS MENU
              ?        -   HELP

   THE FOLLOWING BOND INCREMENT GROUPS ARE DEFINED:

  GR#    GROUP ID          GR#    GROUP ID          GR#    GROUP ID
  ________________________________________________________________

  199  -  VIN              200  -  CHD              201  -  CHD14
  202  -  CYPD             203  -  CYPENE           204  -  CLBZ
  205  -  CHENE            206  -  CHENEA           207  -  BZ
  208  -  P                209  -  S                210  -  T
  211  -  ALPEROX          212  -  ALKOXY           213  -  PHENOXY
  214  -  ALLYLP           215  -  ALLYLS           216  -  ALLYLT

    ENTER BOND INCREMENT NAME ONLY
         enter option : { Q }
```

NOTE: These are the bond increments included with this package. If desired, more can be added to the file BD.

All bond increments must be included in this file.

Bond increments consist of a bond energy (assuming loss of H atom from the parent molecule), S and Cp increments to account for respective changes due to loss of H atom from the parent. These include loss of vibrational modes as well as adjustments for changes in barriers to internal rotation.

BOND_DISSOCIATION "BD" INCREMENTS EXPLAINED:

OPTION VIN:

VIN - VINYL RADICAL

BOND DISSOCIATION ENERGY (assuming loss of H atom) = 105.5 Kcal/mol

based upon loss of H atom from ethylene:

C2H4 ---> C2H3 + H

OPTION CHD:

CHD - CYCLOHEXA-1,3-DIENYL RADICAL

BOND DISSOCIATION ENERGY (assuming loss of H atom) = 76.6 Kcal/mol

based upon loss of H atom from CY13HD ---> CYC6H7 + H

OPTION CHD14:

CHD - CYCLOHEXA-1,4-DIENYL RADICAL

BOND DISSOCIATION ENERGY (assuming loss of H atom) = 75.8 Kcal/mol

based upon loss of H atom from CY14HD ---> CYC6H7 + H

OPTION BZ:

BZ - PHENYL RADICAL

BOND DISSOCIATION ENERGY (assuming loss of H atom) = 113.65 Kcal/mol

based upon loss of H atom from CYC6H6 ---> CYC6H5 + H

OPTION CYPD:

CYPD - CYCLOPENTADIENYL RADICAL

BOND DISSOCIATION ENERGY (assuming loss of H atom) = 74.0 Kcal/mol

based upon loss of H atom from CYPD ---> CYPD. + H

OPTION CYPENE:

CYPENE - ALLYLIC CYCOLOPENTENYL RADICAL

BOND DISSOCIATION ENERGY (assuming loss of H atom) = 84.2 Kcal/mol

based upon loss of H atom from CYC5H8 ---> CYC5H7. + H

OPTION CLBZ:

CLBZ - CHLOROPHENYL RADICAL

BOND DISSOCIATION ENERGY (assuming loss of H atom) = 112.5 Kcal/mol

based upon loss of H atom from C6H5CL ---> C6H4CL + H

OPTION CHENE:

CHENE - CYCOLHEXENENYL RADICAL { NON-ALLYLIC }

BOND DISSOCIATION ENERGY (assuming loss of H atom) = 98.98 Kcal/mol

based upon loss of H atom from CYC6H9 ---> CYC6H9 + H

OPTION CHENEA:

CHENEA - CYCOLHEXENENYL RADICAL { ALLYLIC }

BOND DISSOCIATION ENERGY (assuming loss of H atom) = 83.8 Kcal/mol

based upon loss of H atom from CYC6H9 ---> CYC6H9A + H

OPTION P:

P - PRIMARY RADICAL

BOND DISSOCIATION ENERGY (assuming loss of H atom) = 100.6 Kcal/mol

based upon loss of H atom from C2H6 ---> C2H5 + H

Barrier to internal rotation is lower for the CH2 rotor in the product than it is for the CH3 rotor in the parent.

OPTION S:

S - SECONDARY RADICAL

BOND DISSOCIATION ENERGY (assuming loss of H atom) = 98.2 Kcal/mol

based upon loss of H atom from CCC ---> CC.C + H (GUTMAN)

OPTION T:

T - Tertiary Radical

BOND DISSOCIATION ENERGY (assuming loss of H atom) = 95.2 Kcal/mol

based upon loss of H atom from C3C ---> C3C. + H (W. Tsang)

OPTION ALPEROX:

ALPEROX - ALKYL PEROXY RADICAL:

BOND DISSOCIATION ENERGY (assuming loss of H atom) = 88.0 Kcal/mol

based upon loss of H atom from ROOH ---> ROO. + H

OPTION ALKOXY:

ALKOXY - ALKOXY RADICAL

BOND DISSOCIATION ENERGY (assuming loss of H atom) = 104.1 Kcal/mol

based upon loss of H atom from CH3OH ---> CH3O. + H
Entropy loss an average of 2.6 for loss of rotor.

OPTION PHENOXY:

PHENOXY - PHENOXY RADICAL

BOND DISSOCIATION ENERGY (assuming loss of H atom) = 86.5 Kcal/mol

based upon loss of H atom from C6H5OH ---> CYC6H5O. + H

OPTION ALLYLP:

ALLYLP - primary allylic

Bond dissociation energy (assuming loss of H atom) = 88.2 Kcal/mol

average for delta H for following two reactions:

C*CC --> C*CC. + H
CC*CC --> CC*CC. + H

OPTION ALLYLS:

ALLYLS - secondary allylic

Bond dissociation energy (assuming loss of H atom) = 84.4 Kcal/mol

based upon loss of H atom from C*CCC --> C*CC.C + H

OPTION ALLYLT:

ALLYLT - tertiary allylic

Bond dissociation energy estimated as 81.8 Kcal/mol
Bozzelli : estimate

EXAMPLES: SAMPLE SESSION

```
                              THERM


              THermo Estimation for Radicals & Molecules



                            written by

                    E. R. Ritter & J. W. Bozzelli
              Department of Chemical Engineering and Chemistry
                    New Jersey Institute of Technology



                        revision 4.0 : 2/11/89

                **** Loading Configuration File ****
                 ***** LOADING GROUP DATAFILES *****
                     336  Groups in the dataBase.

                      { hit return to continue }
```

```
======================== next screen ====================================

                              THERM MAIN MENU

      Units:  Kcal/mol  ( cal/mol K )                  3/ 5/89
   ____________________________________________________________________

   1 - ENTER/ESTIMATE SPECIES      5 - REVIEW (LST FORMAT FILE)
   2 - VIEW GROUPS                 6 - GET SPECIES FROM FILE {*.DOC}
   3 - SELECT SHORT MENU           7 - ReWrite THERMO FILES {*.DOC, *.LST}
   4 - SELECT LIST FILE {*.LST}    8 - RUN THERMFIT create NASA {*.DAT}
             9 - RUN THERMRXN (calculate dU, dH, dS, dG for reaction)
            10 - Sort & ReWrite THERMO FILES { *.DOC, *.LST }
            11 - Sort/Write *.LST file { no *.DOC is needed }
            12 - RUN THERMLST (LIST NASA FORMAT FILE) /create *.LST
            13 - DOS utilities
             U - CHANGE CURRENT UNITS TO : KJ/mol  (J/mol K)
                     *  -  CLOSE ALL OPENED FILES
                     X  -  EXIT PROGRAM
                     ?  -  HELP

                      enter option : { 1 }  <------------  default option
                                                          (press return)
```

```
========================= next screen =====================================



    Enter species name (ID)
      9 characters maximum
{ "Q" return to opening menu }
BIPHENYL    <------------------------ enter the species name  and
                                      press return.

=========================== next screen ================================




enter the elemental formula for this species
{ example: for PHENOL enter " C6H5OH " }
{          for TOLUENE enter " C6H5CH3 " }
{          for CHLOROPHENOL enter " C6H4CLOH " }



       4 ELEMENTS MAXIMUM : 70 CHARACTERS MAX

  THE FOLLOWING ELEMENTS ARE DEFINED:
  C   H   O   CL  S   N   F   P   I   BR
  B

 ALL OTHER ELEMENTS OR NON-NUMERIC CHARACTERS
            WILL BE IGNORED !

 enter formula here: {Q=QUIT}
 C6H5C6H5   <-------------------------------- enter elemental formula
                                              C6H5C6H5  or C12H10


=========================== next screen ================================

FORMULA ENTERED :
C6H5C6H5


  ELEMENTAL COMPOSITION:

C            12
H            10            <--------------- this message is disabled if
              0                             "ERRORCK=OFF" appears
              0                             in the config file
                                            (see THERM.CFG)

 number of Atoms for This Species =   22

    IF THIS IS INCORRECT ENTER " Q "
           TO BEGIN AGAIN
  OTHERWISE HIT RETURN TO CONTINUE    <------------------ press return
```

```
======================== next screen ====================================
                                                    ____
   Radicals are estimated either by entering a          |
   radical {CDOT} group such as " CB/. " on this        |
   screen; or by first estimating the properties        |
   for the stable molecule here { ex: benzene } and     |
   then applying a bond increment {BD} group to         |
   account for loss of H atom from the molecule.        |
                                                        | This message can
   Bond increments cannot be entered here.  They        | be disabled if
   are entered at the RADICAL menu { available          | "ERRORCK=OFF" is
   after the molecule's properties are estimated }.     | set in THERM.CFG.
                                                        |
   You must be sure that groups you enter adequately    |
   define a molecule (no unbonded electrons).           |
   THERM makes no attempt to check the context          |
   of the groups entered (only whether they are defined).|
                                                    ____|

       Enter the Number of Groups in this species
 (each Central Atom or Ring Correction counts as 1 group)
12  <----------------------------------------------------- enter 12

How many different groups are to be entered?
1  <------------------------------------------------------- enter 1
```

NOTE: To enter literature data one would answer "0" to the prompt "Enter the number of Groups in this species". If one chooses to enter literature data, they are prompted for different information than that which follows. Literature data remains unchanged when automatic recalculation is performed.

```
======================== next screen ====================================

how  many  rotors  in this species {0}    enter the number of rotors in
1    <--------------------------------    this species and press return
```

```
========================= next screen =====================================


        336 groups in the database
                                                   __
You must enter a total of   2 different groups    |   one may review the group
  {enter " D " to re-enter the last group}        |   database "L", start a new
  {enter " Q " to start a new species}            |   species "Q", or correct/
  {enter " L " to list group database}            |   re-enter the last group
                                                __|   "D" by entering one of
                                                      these options instead of
                                                      a group id.



 ENTER: "GROUP ID,QUANTITY"   <------- a group ID and quantity  are separated
                                       by  either a space or a comma.   If  no
                                       quantity is entered, "1" is assumed.
ENTER GROUP #   1
CB/H 10           <--------------------- enter the group id : CB/H,10
ENTER GROUP #   2
CB/CB 2           <--------------------- enter the group id : CB/CB,2
     Enter symmetry # for molecule
8    <---------------------------------- enter symmetry number for
                                          biphenyl  :  8
                                         IF no symmetry number is entered,
                                         "1" is assumed.

========================== next screen ============================


Thermo estimation for molecule
BIPHENYL                                                  C6H5C6H5

  Gr # -   GROUP ID  - Quantity
     1 - CB/H           - 10
     2 - CB/CB          -  2

UNITS:  Hf{Kcal/mol}    S & Cp {cal/mol K}
   Hf       S     Cp 300    400       500      600      800     1000     1500
  42.92   93.89   39.06   52.84   64.38   73.54   86.92   96.00  109.40
     CPINF    127.17
Symmetry number used to correct S value is          8
  EDIT THESE RESULTS?     "Y" OR {"N"}       <------------- press return
                                                            to accept no
```

```
======================== next screen ====================================

                SPECIES ESTIMATION OPTIONS MENU
        Units:  Kcal/mol  ( cal/mol K )
        _______________________________________________

        R      -  CALCULATE RADICAL BASED UPON PREVIOUS SPECIES
        F      -  ADD CURRENT SPECIES TO OUTPUT FILE
        H      -  TOGGLE (SHORT/LONG) MENU
        P      -  RETURN TO PREVIOUS SPECIES SCREEN
        RETURN -  ENTER A NEW SPECIES
        L      -  LIST GROUPS
        Q      -  RETURN TO MAIN MENU {choose "F"
                  first; current species is discarded
                  when you return to MAIN menu}
        X      -  EXIT PROGRAM
        S      -  GET SPECIES FROM { *.DOC } FILE
        ?      -  HELP
        U      -  CHANGE CURRENT UNITS TO : KJ/mol  (J/mol K)

      enter option : { ENTER NEW SPECIES }
 F   <----------------------------------------------- choose "F" to file
                                                       (press return)

========================== next screen ================================


Enter output filename:
{ no file extension is needed }:
THERMO        <----------------------- enter a filename for output

   CREATE DOCUMENT FILE [ Y or N ]

   enter choice: { Y }   <-------------- press return

 choose output units for LST file
 1- Kcal/mol (cal/mol K)   2- KJ/mol (J/mol K )
     enter option { 1 }
 2  <-------------------------- enter "2" for output in KJ

    enter title  for *.LST file output { blank or Q= no title }
EXAMPLE USE OF THERM   <------------- enter a file heading if desired
```

NOTE: If the file which is specified already exists, the user is asked if it should be overwritten, appended, or if a different filename is desired.

```
======================== next screen =====================================

                SPECIES ESTIMATION OPTIONS MENU
        Units:  Kcal/mol  ( cal/mol K )
        ____________________________________________

        R       -  CALCULATE RADICAL BASED UPON PREVIOUS SPECIES
        F       -  ADD CURRENT SPECIES TO OUTPUT FILE
        H       -  TOGGLE (SHORT/LONG) MENU
        P       -  RETURN TO PREVIOUS SPECIES SCREEN
        RETURN  -  ENTER A NEW SPECIES
        L       -  LIST GROUPS
        Q       -  RETURN TO MAIN MENU {choose "F"
                   first; current species is discarded
                   when you return to MAIN menu}
        X       -  EXIT PROGRAM
        S       -  GET SPECIES FROM { *.DOC } FILE
        ?       -  HELP
        U       -  CHANGE CURRENT UNITS TO : KJ/mol  (J/mol K)


 LAST SPECIES FILED:  BIPHENYL    <------------------- biphenyl has been
                                                        filed
       enter option : { ENTER NEW SPECIES }
R   <---------------------------------------------  enter "R" to calculate
                                                      a radical
```

```
========================= next screen =====================================

                         RADICAL CALCULATION MENU
                      {radical by loss of H from parent}
                 _________________________________________

                  Q        -   RETURN TO PREVIOUS MENU
                  ?        -   HELP

 THE FOLLOWING BOND INCREMENT GROUPS ARE DEFINED:

GR#     GROUP ID          GR#     GROUP ID          GR#     GROUP ID
____________________________________________________________________

199  -  VIN               200  -  CHD               201  -  CHD14
202  -  CYPD              203  -  CYPENE            204  -  CLBZ
205  -  CHENE             206  -  CHENEA            207  -  BZ
208  -  P                 209  -  S                 210  -  T
211  -  ALPEROX           212  -  ALKOXY            213  -  PHENOXY
214  -  ALLYLP            215  -  ALLYLS            216  -  ALLYLT

  ENTER BOND INCREMENT NAME ONLY
       enter option : { Q }

BZ  <-----------------  choose the "BZ" bond increment or "?" to review
                           bond increment (BD) definitions.  The BD group
                           name is required here. DO NOT enter the number.
```

```
========================= next screen ======================================


PARENT =  C6H5C6H5

BOND TYPE =  BZ

enter the symmetry number for the radical
2  <---------------------------------------- symmetry number for ortho or
                                              meta - biphenyl rdaical is "2"

Enter radical name(ID)
C6H5C6H4.   <------------------------- give the radical a name (9 char max)

enter the elemental formula for this radical
{ example: for PHENYL enter " C6H5 " }
{          for BENZYL enter " C6H5CH2 " }
{          FOR VINYL  enter " C2H3 " }


              4 ELEMENTS MAXIMUM
C6H5C6H4   <------------------------  enter elemental formula for the radical


FORMULA ENTERED :
C6H5C6H4


  ELEMENTAL COMPOSITION:

C             12
H              9
               0    <------------------- this message can be disabled
               0                         see ERRORCK in configuration
                                             file


 number of Atoms for This Species =         21

    IF THIS IS INCORRECT ENTER " Q "
           TO BEGIN AGAIN
  OTHERWISE HIT RETURN TO CONTINUE   <----------- hit return to continue


========================= next screen ======================================

how many rotors in this species (0)     enter the number of rotors in
1    <----------------------------------   this species and press return
```

```
======================== next screen ====================================



Thermo estimation for radical
C6H5C6H4.                                                C6H5C6H4
RADICAL BASED UPON PARENT  BIPHENYL
                       C6H5C6H5
PARENT SYMMETRY          8

  Gr # -  GROUP ID  - Quantity
     1 - CB/H           - 10
     2 - CB/CB          -  2
     3 - BZ             -  1

UNITS:  Hf{Kcal/mol}   S & Cp {cal/mol K}
   Hf       S     Cp 300   400      500     600     800     1000    1500
 104.47   97.76   38.19   51.23   62.16   70.83   83.51   92.16  104.90
     CPINF   121.21
Symmetry number used to correct S value is           2
R ln(2) has been added to S to account for unpaired electron
bond dissociation energy used=      113.65
 EDIT THESE RESULTS?     "Y" OR {"N"}   <-----------  press  return
```

```
========================= next screen =====================================

                SPECIES ESTIMATION OPTIONS MENU
        Units:  Kcal/mol  ( cal/mol K )
        _________________________________________________________

        R      -  CALCULATE RADICAL BASED UPON PREVIOUS SPECIES
        F      -  ADD CURRENT SPECIES TO OUTPUT FILE
        H      -  TOGGLE (SHORT/LONG) MENU
        P      -  RETURN TO PREVIOUS SPECIES SCREEN
        M      -  RETURN TO PREVIOUS PARENT MOLECULE
        RETURN -  ENTER A NEW SPECIES
        L      -  LIST GROUPS
        Q      -  RETURN TO MAIN MENU {choose "F"
                  first; current species is discarded
                  when you return to MAIN menu}
        X      -  EXIT PROGRAM
        S      -  GET SPECIES FROM { *.DOC } FILE
        ?      -  HELP
        U      -  CHANGE CURRENT UNITS TO : KJ/mol  (J/mol K)


 LAST SPECIES FILED:  BIPHENYL

      enter option : { ENTER NEW SPECIES }
F   <----------------------------------------------  file this entry.
                                                      enter "F" and press
                                                      return
```

NOTE: Since an output file has already been connected, the user is not prompted for any file information. To change the file, one must return to the MAIN menu and choose option "*" to close all opened files. Then the next time option "F" is chosen, the user will again be prompted for filename, etc.

```
======================== next screen ====================================


                SPECIES ESTIMATION OPTIONS MENU
        Units:  Kcal/mol  ( cal/mol K )
        _______________________________________________

        R      -  CALCULATE RADICAL BASED UPON PREVIOUS SPECIES
        F      -  ADD CURRENT SPECIES TO OUTPUT FILE
        H      -  TOGGLE (SHORT/LONG) MENU
        P      -  RETURN TO PREVIOUS SPECIES SCREEN
        M      -  RETURN TO PREVIOUS PARENT MOLECULE
        RETURN -  ENTER A NEW SPECIES
        L      -  LIST GROUPS
        Q      -  RETURN TO MAIN MENU {choose "F"
                  first; current species is discarded
                  when you return to MAIN menu}
        X      -  EXIT PROGRAM
        S      -  GET SPECIES FROM { *.DOC } FILE
        ?      -  HELP
        U      -  CHANGE CURRENT UNITS TO : KJ/mol  (J/mol K)


LAST SPECIES FILED:  C6H5C6H4.

       enter option : { ENTER NEW SPECIES }
L     <---------------------------------------- enter "L" to list groups
```

```
                          LIST GROUPS MENU
                     ____________________________

                     1  -            HC
                     2  -          CYCH
                     3  -          CDOT
                     4  -            BD
                     5  -           CLC
                     6  -           HCO
                     7  -           HCN
                     A  -  All Groups in DataBase
                     Q  -  QUIT { return to previous menu}

                       enter option : { Q }

 4  <--------------  choose option "4"  to review bond increment (BD) groups

========================== next screen ======================================

GR#     GROUP ID          GR#     GROUP ID          GR#     GROUP ID
____________________________________________________________________________

199  -  VIN               200  -  CHD               201  -  CHD14
202  -  CYPD              203  -  CYPENE            204  -  CLBZ
205  -  CHENE             206  -  CHENEA            207  -  BZ
208  -  P                 209  -  S                 210  -  T
211  -  ALPEROX           212  -  ALKOXY            213  -  PHENOXY
214  -  ALLYLP            215  -  ALLYLS            216  -  ALLYLT
HIT RETURN TO CONTINUE  { "?" FOR HELP }
?    <------------------------------------- enter "?" for help

============================ next screen ================================

        LIST DATABASE OPTIONS
        _____________________

   Q -  QUIT return to previous menu
  RETURN - TO CONTINUE
   R -  REPEAT listing (from beginning)
   E -  EXTENDED listing (Hf S Cp )
  REF-  EXTENDED listing with references

       enter option {continue}
REF   <-------------------------  enter "REF" to review group data with
                                   references
 ENTER RANGE OF GROUPS TO VIEW
199 216   <---------------------  specify the range to display
                                  ( this will display groups
                                  199 {VIN} to 216 {ALLYLT}
                                  as shown above.  Group names could be
                                  entered  instead ( VIN ALLYLT )  to  specify
                                  the range.
```

```
==================  on-line review of group values and references ===========

   Hf { Kcal/mol }    S & Cp  { cal/mol K }
GROUP ID          Hf       S      Cp 300    400     500     600      800     1000
VIN             105.54   1.06    -.16     -.67   -1.17   -1.65   -2.55   -3.31
   Stein , 22nd Symp (International) on Combustion : Seattle 1988
CHD              76.56    .48   -1.87    -2.26   -2.63   -2.96   -3.56   -4.11
    W. Tsang   J Phys Chem 1986, 90,  1152
CHD14            75.80   1.21   -1.83    -2.09   -2.37   -2.67   -3.30   -3.93
    W. Tsang   J Phys Chem 1986, 90,  1152
CYPD             74.00   4.62    -.40     -.44    -.61    -.95   -1.83   -2.68
CYPENE           84.17   4.13    -.38     -.42    -.59    -.96   -1.90   -2.80
CLBZ            112.12   1.12    -.87    -1.61   -2.22   -2.70   -3.41   -3.84
    bd lowered by 1 kcal over benzene (bz) : Louw, Mulder
CHENE            98.98   5.78    -.12     -.71   -1.25   -1.74   -2.57   -3.22
CHENEA           83.83   4.71    -.13     -.74   -1.29   -1.78   -2.61   -3.28
BZ              113.65   1.12    -.87    -1.61   -2.22   -2.71   -3.41   -3.84
    W. Tsang   J Phys Chem 1986, 90,  1152
P               100.64   2.30    -.78    -1.19   -1.62   -2.09   -3.01   -3.80
    W. Tsang   I J Chem Kin 1987, 10, 821.  JACS  1985, 107, 2822
S                98.10   4.50   -1.50    -2.33   -3.10   -3.39   -3.75   -4.45
    W. Tsang   I J Chem Kin 1987, 10, 821.  JACS  1985, 107, 2822
T                95.87   6.48   -5.22    -5.52   -5.71   -5.77   -5.69   -5.54
    Russell and Gutman   JACS  1988,  110,  3084 & 3092
  { HIT RETURN TO CONTINUE } <------------------------------  press return
   Hf { Kcal/mol }    S & Cp  { cal/mol K }
GROUP ID          Hf       S      Cp 300    400     500     600      800     1000
ALPEROX          88.50  -2.10   -2.00    -2.65   -3.10   -3.40   -3.64   -3.60
ALKOXY          104.06  -2.60   -1.47    -1.82   -2.12   -2.36   -2.72   -2.97
PHENOXY          86.50  -2.60   -1.38    -2.02   -2.53   -2.91   -3.44   -3.81
    Burcat, Zelznik, McBride  nasa tech memo #83800 January 1985
ALLYLP           88.20  -1.60    -.51     -.72    -.99   -1.33   -2.08   -2.81
    avg {( C*CC --> C*CC. + H ) + ( CC*CC --> CC*CC. + H)}/2
ALLYLS           84.40   -.62    -.78    -1.19   -1.62   -2.09   -3.01   -3.78
    C*CCC --> C*CC.C + H
ALLYLT           81.80    .88   -1.50    -2.36   -3.10   -3.49   -3.95   -4.45
    Bozzelli : calc from alkyl bond trends & data in McMillian/Golden (1983)
  { HIT RETURN TO CONTINUE }
```

NOTE: reference appears on the line following the group data
       if there is no reference, none is listed. Most groups
       with no reference are from Benson.

```
       LIST DATABASE OPTIONS
       _________________
  Q -  QUIT return to previous menu
 RETURN - TO CONTINUE
  R -  REPEAT listing (from beginning)
  E -  EXTENDED listing (Hf S Cp )
 REF-  EXTENDED listing with references

      enter option {continue}
Q  <------------------------------------- return to previous menu

=========================== next screen ================================

                           LIST GROUPS MENU
                    ________________________________

                    1  -            HC
                    2  -          CYCH
                    3  -          CDOT
                    4  -            BD
                    5  -           CLC
                    6  -           HCO
                    7  -           HCN
                    A  -  All Groups in DataBase
                    Q  -  QUIT { return to previous menu}

                      enter option : { Q }
Q  <------------ "quit" back to previous menu

=========================== next screen ================================

                    SPECIES ESTIMATION OPTIONS MENU
            Units:  Kcal/mol  ( cal/mol K )
            ________________________________________________

            R      -  CALCULATE RADICAL BASED UPON PREVIOUS SPECIES
            F      -  ADD CURRENT SPECIES TO OUTPUT FILE
            H      -  TOGGLE (SHORT/LONG) MENU
            P      -  RETURN TO PREVIOUS SPECIES SCREEN
            M      -  RETURN TO PREVIOUS PARENT MOLECULE
            RETURN -  ENTER A NEW SPECIES
            L      -  LIST GROUPS
            Q      -  RETURN TO MAIN MENU {choose "F"
                      first; current species is discarded
                      when you return to MAIN menu}
            X      -  EXIT PROGRAM
            S      -  GET SPECIES FROM { *.DOC } FILE
            ?      -  HELP
            U      -  CHANGE CURRENT UNITS TO : KJ/mol  (J/mol K)

LAST SPECIES FILED:  C6H5C6H4.

         enter option : { ENTER NEW SPECIES }
Q  <---------- "quit" back to MAIN menu
```

```
========================= next screen ======================================

                              THERM MAIN MENU

    Units:  Kcal/mol  ( cal/mol K )                    3/ 5/89
   ________________________________________________________________

   1 - ENTER/ESTIMATE SPECIES     5 - REVIEW (LST FORMAT FILE)
   2 - VIEW GROUPS                6 - GET SPECIES FROM FILE {*.DOC}
   3 - SELECT SHORT MENU          7 - ReWrite THERMO FILES {*.DOC, *.LST}
   4 - SELECT LIST FILE {*.LST}   8 - RUN THERMFIT create NASA {*.DAT}
             9 - RUN THERMRXN (calculate dU, dH, dS, dG for reaction)
            10 - Sort & ReWrite THERMO FILES { *.DOC, *.LST }
            11 - Sort/Write *.LST file { no *.DOC is needed }
            12 - RUN THERMLST (LIST NASA FORMAT FILE) /create *.LST
            13 - DOS utilities
             U - CHANGE CURRENT UNITS TO : KJ/mol  (J/mol K)
                       *  -  CLOSE ALL OPENED FILES
                       X  -  EXIT PROGRAM
                       ?  -  HELP

                         enter option : { 1 }
6    <------------  choose "6" to get species data from a "DOC" file
```

```
======================== next screen ===================================

              enter document filename

              for species input
TEST  <--------------------------- enter filename for input "TEST.DOC"
                                    the file extension ".DOC" is understood.


                    INITIALIZING
              THERMO DOCUMENTATION FILE



                     PLEASE WAIT



  1  FBZ            C    6  H    5  F    1        0
  2  BZ             C    6  H    6       0        0
  3  CLBZ           CL   1  C    6  H    5        0
  4  C6H5CH3        C    7  H    8       0        0
  5  IBZ            C    6  H    5  I    1        0
  6  BIPHENYL       C   12  H   10       0        0
  7  CLBIPHENY      CL   1  C   12  H    9        0
  8  PHENOL         C    6  H    6  O    1        0
  9  PHENOXY        C    6  H    5  O    1        0
 10  C6F6           C    6  F    6       0        0
 11  PHENYL         C    6  H    5       0        0
 12  C6H5.          C    6  H    5       0        0
 13  CH3OCH3        C    2  H    6  O    1        0
 14  C2H6           C    2  H    6       0        0
 15  C2H5           C    2  H    5       0        0
 16  C2H5CL         CL   1  C    2  H    5        0
 17  C2H4           C    2  H    4       0        0
 18  C2H3           C    2  H    3       0        0
 19  C2H3CL         CL   1  C    2  H    3        0
 20  PHPH           C   12  H   10       0        0
 21  PHPH.          C   12  H    9       0        0
 22  CY13HD         C    6  H    8       0        0
 23  CYC5E          C    5  H    8       0        0
 24  C5DE14         C    5  H    8       0        0
 25  CYPD           C    5  H    6       0        0
 26  C5DE23         C    5  H    8       0        0
 27  ANILINE        C    6  H    7  N    1        0
 28  C2H4O          C    2  H    4  O    1        0
 29  C20H42         C   20  H   42       0        0
 30  C2H5F          C    2  H    5  F    1        0
 31  CH2CF2         C    2  H    2  F    2        0
```

```
========================= next screen =====================================
  1  FBZ            2  BZ             3  CLBZ           4  C6H5CH3
  5  IBZ            6  BIPHENYL       7  CLBIPHENY      8  PHENOL
  9  PHENOXY       10  C6F6          11  PHENYL        12  C6H5.
 13  CH3OCH3       14  C2H6          15  C2H5          16  C2H5CL
 17  C2H4          18  C2H3          19  C2H3CL        20  PHPH
 21  PHPH.         22  CY13HD        23  CYC5E         24  C5DE14
 25  CYPD          26  C5DE23        27  ANILINE       28  C2H4O
 29  C2OH42        30  C2H5F         31  CH2CF2        32
 enter "D" to delete; "U" to undelete; "V" to view; or hit return to continue
D   <---------  enter "D" to delete a species

DELETE which species { 0 to abort }
    or enter species number to delete}
 1   <---------------------------------- choose to delete FBZ, enter "1"
species #          1
has been deleted

=========================== next screen ===============================

  1                 2  BZ             3  CLBZ           4  C6H5CH3
  5  IBZ            6  BIPHENYL       7  CLBIPHENY      8  PHENOL
  9  PHENOXY       10  C6F6          11  PHENYL        12  C6H5.
 13  CH3OCH3       14  C2H6          15  C2H5          16  C2H5CL
 17  C2H4          18  C2H3          19  C2H3CL        20  PHPH
 21  PHPH.         22  CY13HD        23  CYC5E         24  C5DE14
 25  CYPD          26  C5DE23        27  ANILINE       28  C2H4O
 29  C2OH42        30  C2H5F         31  CH2CF2        32
 enter "D" to delete; "U" to undelete; "V" to view; or hit return to continue
U   <--------------  choose to replace FBZ by entering "U"

=========================== next screen ===============================

  1 -  FBZ             <-------------  all species that have been deleted
                                       but are retained are listed here

 undelete which species { 0 to abort }
     or enter species number to undelete
 1   <------------------------------------  enter "1" to UNDELETE FBZ

=========================== next screen ===============================

  1  FBZ            2  BZ             3  CLBZ           4  C6H5CH3
  5  IBZ            6  BIPHENYL       7  CLBIPHENY      8  PHENOL
  9  PHENOXY       10  C6F6          11  PHENYL        12  C6H5.
 13  CH3OCH3       14  C2H6          15  C2H5          16  C2H5CL
 17  C2H4          18  C2H3          19  C2H3CL        20  PHPH
 21  PHPH.         22  CY13HD        23  CYC5E         24  C5DE14
 25  CYPD          26  C5DE23        27  ANILINE       28  C2H4O
 29  C2OH42        30  C2H5F         31  CH2CF2        32
 enter "D" to delete; "U" to undelete; "V" to view; or hit return to continue
V   <-------------  choose "V" to view data in the file

view which species { enter species number }
```

```
9   <----------  choose species #9 : phenoxy radical
========================= next screen ======================================


 SPECIES
 PHENOXY
Thermo estimation for radical
 PHENOXY                                                     C6H5O
 RADICAL BASED UPON PARENT  PHENOL
 PARENT FORMULA                        C6H5OH
PARENT SYMMETRY           2
 UNITS:KJ
 GROUPS   4
   Gr # -   GROUP ID   - Quantity
      1 - O/CB/H         -  1
      2 - CB/O           -  1
      3 - CB/H           -  5
      4 - PHENOXY        -  1
   Hf      S     Cp 300     400      500      600      800     1000     1500
    50.63  303.64  96.32 125.02 149.75  169.12  197.40  216.52      .00
           CPINF =  282.66
 SYMMETRY       2
R ln(2) has been added to S to account
       for unpaired electron
 BOND  361.92
  ENDSPECIES



{ hit return to continue }
```

NOTE: *this is only an image of the data recorded in the "DOC" file.*
This option allows one to view species data before choosing to load
the data into memory.

```
========================= next screen ======================================

  1  FBZ             2  BZ             3  CLBZ           4  C6H5CH3
  5  IBZ             6  BIPHENYL       7  CLBIPHENY      8  PHENOL
  9  PHENOXY        10  C6F6          11  PHENYL        12  C6H5.
 13  CH3OCH3        14  C2H6          15  C2H5          16  C2H5CL
 17  C2H4           18  C2H3          19  C2H3CL        20  PHPH
 21  PHPH.          22  CY13HD        23  CYC5E         24  C5DE14
 25  CYPD           26  C5DE23        27  ANILINE       28  C2H4O
 29  C20H42         30  C2H5F         31  CH2CF2        32
 enter "D" to delete; "U" to undelete; "V" to view; or hit return to continue

              <---------------- press return to continue


         enter species id ( "?" for options)
CLBZ   <---------------- enter "CLBZ" and groups and other data are loaded
                               into memory for recalculation/editing


{ hit return to continue : ? for options }

=========================== next screen =================================

Thermo estimation for molecule
CLBZ                                                       C6H5CL

  Gr # -  GROUP ID  - Quantity
     1 - CB/H          -  5
     2 - CB/CL         -  1

UNITS:  Hf{Kcal/mol}   S & Cp {cal/mol K}
   Hf      S     Cp 300   400     500     600     800    1000    1500
  12.39   74.82   23.37   30.55   36.46   41.21   48.04   52.73
     CPINF    67.56
Symmetry number used to correct S value is          2
  EDIT THESE RESULTS?   { "Y" OR "N"}
```

NOTE: Properties have actually been recalculated from group database and data is in memory as if the user had just typed it in.

Literature data is not recalculated. The keyword "NOCALC" appears with every literature entry.

USING THERMFIT

Cp EXTRAPOLATION AND POLYNOMIAL FITTING PROGRAM

This stand alone program may be run from the THERM main menu or may be run independent from THERM. Input file is LST format and output is DAT format (thermo property polynomials). To run independent enter "THERMFIT" at the DOS prompt and press return.

```
THERMFIT : Polynomial Fitting & Cp Extrapolation Method


ENTER FILENAME FOR INPUT (.LST)
EXAMPLE     <------------------------ enter input filename (extension "LST"  is
                                        assumed.
ENTER FILENAME FOR OUTPUT (.DAT)
EXAMPLE     <------------------------ enter output filename (extension "DAT"  is
                                        assumed.

   GENERATING NASA FORMAT DATABASE FROM
    TEST.LST

             please wait
C6H5F
C6H6
C6H5CL
C6F6
C6CL6
C6H5CH3      <---------------- species names are listed as each is
C6H5I                          processed.
C6H5C6H5
C6H5OH
C6H5O.
```

```
EXAMPLE LIST FILE ( LST )
SPECIES       Hf       S    Cp 3(0    400     500     600     800    1000    1500     DATE          ELEMENTS
C6H5F     -26.30   72.37   22.50   29.80   35.80   40.60   47.50   52.25     .00   1/23/89        C   6 H   5 F   1     0 G 0
C6H6       19.80   64.24   19.44   26.64   32.76   37.80   45.24   50.46   58.38   1/23/89        C   6 H   6     0     0 G 0
C6H5CL     12.70   75.17   24.60   29.60   37.00   40.70   47.90   52.45     .00   1/23/89        CL  1 C   6 H   5     0 G 0
C6F6     -228.64   91.59   37.56   43.88   48.78   52.55   57.62   60.63                    STULL C   6 F   6     0     0 G 0
C6CL6      -8.10  105.45   41.36   46.00   50.52   53.00   57.74   60.28                    THERM CL  6 C   6     0     0 G 0
C6H5CH3    11.81   76.81   25.06   33.18   40.38   46.44   55.68   62.26   72.21   1/23/89        C   7 H   8     0     0 G 1
*C6H5I     39.00   79.97   24.20   31.10   36.90   41.40   48.00   52.55     .00   1/23/89        C   6 H   5 I   1     0 G 0
C6H5C6H5   42.92   93.89   39.06   52.84   64.38   73.54   86.92   96.00  109.40   1/23/89        C  12 H  10     0     0 G 1
PHENOL    -22.30   75.17   24.40   31.90   38.32   43.33   50.62   55.56     .00   1/23/89        C   6 H   6 O   1     0 G 1
PHENOXY    12.10   72.57   23.02   29.88   35.79   40.42   47.18   51.75     .00   1/23/89        C   6 H   5 O   1     0 G 0
```

NOTE: this is a sample LST format file (*.LST) as created by the group additivity code. LST files produced by THERMLST have essentially the same format, except Cp 1500 K is never blank or zero.

- If Cp 1500 K is blank or zero that means one or more groups considered had no data available at this temperature. Once data has been fit using this program, Cp data will be calculated to 5000 K unless an asterisk appears in column 1. An asterisk in column 1 "*" indicates that the desired temperature range is limited to 2000 K or that extrpolation was not successful.

- The last column contains the number of rotors which can be considered free rotors at high temperature. This is used to adjust the high temp limiting heat capacity (Cp infinity).

```
--------      THIS IS THE FORMAT OF OUTPUT FROM THERMFIT    ----------
THERMO                             <---------- first two lines are padded with
   300.000  1500.000  5000.000                  blanks to column 80
FBZ          1/23/89        C   6H   5F   1    0G   300.000  5000.000 1394.000     01
 1.61833718E+01 1.55046197E-02-5.43776755E-06 8.58686527E-10-5.04169920E-14     2
-2.11896966E+04-6.54224811E+01-4.64588189E+00 6.87643930E-02-5.85165373E-05     3
 2.52183330E-08-4.35564823E-12-1.44380166E+04 4.47766797E+01                   4
BZ           1/23/89        C   6H   6    0    0G   300.000  5000.000 1387.000     01
 1.39935036E+01 1.74971856E-02-6.15476220E-06 9.73716969E-10-5.72415007E-14     2
 2.47175791E+03-5.77602198E+01-5.39447911E+00 6.37500472E-02-4.90152689E-05     3
 1.93648825E-08-3.13307408E-12 9.13590251E+03 4.60723136E+01                   4
CLBZ         1/23/89        CL  1C   6H   5    0G   300.000  5000.000 1402.000     01
 1.76097340E+01 1.39051666E-02-4.81202456E-06 7.54213602E-10-4.40866883E-14     2
-2.09013038E+03-7.20465247E+01-2.22953876E-01 4.92544675E-02-2.68122926E-05     3
 4.13248078E-09 5.27159680E-13 4.49728146E+03 2.55701514E+01                   4
C6F6ST       STULL          C   6F   6    0    0G   300.000  5000.000 1399.000     01
 2.58396239E+01 7.52573639E-03-2.73763134E-06 4.42881600E-10-2.64426394E-14     2
-1.25330418E+05-1.08117770E+02 3.78000514E+00 6.79621910E-02-6.65244670E-05     3
 3.09697946E-08-5.56965091E-12-1.18686491E+05 6.98893678E+00                   4
C6CL6TH      THERM          CL  6C   6    0    0G   300.000  5000.000 1388.000     01
 2.55178960E+01 7.73146882E-03-2.79189838E-06 4.49470408E-10-2.67455853E-14     2
-1.38549596E+04-9.84421553E+01 9.57209477E+00 4.87056064E-02-4.35234301E-05     3
 1.89791732E-08-3.26102369E-12-8.74666935E+03-1.42163133E+01                   4
C6H5CH3      1/23/89        C   7H   8    0    0G   300.000  5000.000 1382.000     11
 1.61733987E+01 2.32878469E-02-8.15197962E-06 1.28545400E-09-7.53928226E-14     2
-2.87513958E+03-6.63277432E+01-3.90646567E+00 6.70276172E-02-4.42253282E-05     3
 1.48118855E-08-2.04209181E-12 4.49160027E+03 4.27682967E+01                   4
IBZ          1/23/89        C   6H   5I   1    0G   300.000  5000.000 1397.000     01
 1.65325962E+01 1.49823586E-02-5.20668476E-06 8.17232192E-10-4.77828786E-14     2
 1.18034749E+04-6.30109665E+01-3.13809856E+00 6.61721608E-02-5.70980103E-05     3
 2.49774647E-08-4.36424464E-12 1.80792156E+04 4.07227701E+01                   4
BIPHENYL     1/23/89        C  12H  10    0    0G   300.000  5000.000 1393.000     11
 2.78999627E+01 3.12250127E-02-1.09076740E-05 1.71779966E-09-1.00667229E-13     2
 7.36037385E+03-1.30290245E+02-1.05041920E+01 1.29491922E-01-1.09365524E-04     3
 4.73171544E-08-8.25035721E-12 1.98533421E+04 7.29519852E+01                   4
PHENOL       1/23/89        C   6H   6O   1    0G   300.000  5000.000 1399.000     11
 1.65690590E+01 1.72953159E-02-5.95971999E-06 9.30134800E-10-5.41699078E-14     2
-1.93122929E+04-6.66265382E+01-4.30712135E+00 7.12378295E-02-6.01302959E-05     3
 2.58824065E-08-4.46320704E-12-1.26228735E+04 4.35843352E+01                   4
PHENOXY      1/23/89        C   6H   5O   1    0G   300.000  5000.000 1399.000     01
 1.58794328E+01 1.54008223E-02-5.32171061E-06 8.32198383E-10-4.85354318E-14     2
-1.58383272E+03-6.31930634E+01-3.64980357E+00 6.52692783E-02-5.45970495E-05     3
 2.31094723E-08-3.91027027E-12 4.71529505E+03 4.00876924E+01                   4
END   <---------- padded with blanks to column 80.
```

NOTE: This is a sample thermo property polynomial file created by THERMFIT. Some slight differences exist between this file format and that supplied with the CHEMKIN code.

Perhaps the most significant difference is the break point (or common) temperature. This fitting method determines the best position for the breakpoint rather than holding it constant at 1000 K.

The number of rotors appears in column 79 (unused by CHEMKIN). These differences are transparent to the CHEMKIN code

USING THERMRXN

THERMODYNAMIC ANALYSIS FOR REACTION

This stand alone program may be run from the THERM main menu or may be run independent of THERM. Thermo property polynomials are required input (DAT format). Reactions may either be entered interactively or entire CHEMKIN inout files may be processed. To run this program independent of THERM, enter "THERMRXN" at the DOS prompt and press return.

THERMRXN:

Thermodynamic Property Analysis for Reactions

```
   ENTER FILENAME FOR NASA FORMAT COEFFICIENTS   { *.DAT }

                         enter filename containing thermodynamic
   EXAMPLE    <---------- property data in polynomial format.
                         ( file extension DAT is assumed )
```

NOTE: If one wants to use a polynomial file which was not created by this package, it will be necessary to run the utility program "STOD.EXE" to verify the record length is 80 characters per line. THERMRXN requires the first two lines of the polynomial format file to be padded with blanks to 80 characters (required record length). STOD will do this automatically. This is called DIRECT ACCESS FILE FORMAT. STOD stands for Sequential TO Direct conversion.

Failure to pad each line to 80 characters will result in a an error message. The polynomial file supplied with Sandia's "CHEMKIN" requires this change before it can be read. This format change is transparent to the CHEMKIN code, and allows quicker operation
when used with THERM.

```
========================= next screen ===================================

 current units: Kcal/mol (cal/mol K)
 ________________________________

    To Calculate Property Change of Reaction for
    enthalpy (del H), entropy (del S), and
    gibbs free energy (del G)
    enter a Reaction expression as follows:

      " A+B=C+D " or " A+B=C " or " A=B+C "
     (ad'n/abstr)   (ad'n/recomb)  (dissoc)
     where A, B, C, etc.  are species defined
    in the current Thermo D Base {*.DAT} file
 EXAMPLE:    C6H5CL=CYC6H5+ CL  (blanks are ignored)

    NOTE:    Reaction expression MUST BALANCE
             no quote marks are needed

     enter " L " to list defined species
           " C " to process a "CHEMKIN" input file  <--- CHEMKIN input files
           " Q " to return to the MAIN MENU              can be processed.
           " ? " to supress this information
           " U " change units to Kcal/mol (cal/mol K)


C6H5CL=C6H5.+CL     <---------------- enter reaction expression (blanks are
                          ignored; H +H+M = H2+M and 2H+M=H2+M are equivalent)
```

```
======================== next screen ====================================

    THERMODYNAMIC ANALYSIS for REACTION
Rx          C6H5CL      = C6H5.       + CL
Hf (Kcal/mol)    12.390      81.350      28.900
S (cal/mol K)    74.820      68.920      39.500

  dHr    (298K) =       97.860  Kcal/mol
dU (dE) ( " )  =       97.268  Kcal/mol
  dSr    ( " )  =       33.600  cal/mol K
  dGr    ( " ) .=       87.842  Kcal/mol
 Af/Ar   ( " )  =    3.320E+02

    T (K)  dH(Kcal/mol)  dU(Kcal/mol) dS(cal/mol K)   (Af/Ar) dG(Kcal/mol)
   300.000    9.786E+01    9.726E+01    3.360E+01    3.304E+02    8.778E+01
   400.000    9.787E+01    9.708E+01    3.364E+01    2.522E+02    8.442E+01
   500.000    9.783E+01    9.683E+01    3.354E+01    1.923E+02    8.106E+01
   600.000    9.775E+01    9.656E+01    3.340E+01    1.495E+02    7.771E+01
   800.000    9.756E+01    9.597E+01    3.313E+01    9.786E+01    7.106E+01
  1000.000    9.738E+01    9.540E+01    3.293E+01    7.071E+01    6.445E+01
  1200.000    9.723E+01    9.485E+01    3.279E+01    5.495E+01    5.788E+01
  1500.000    9.703E+01    9.405E+01    3.265E+01    4.085E+01    4.806E+01
  2000.000    9.673E+01    9.276E+01    3.247E+01    2.805E+01    3.179E+01

Q (quit)   F (file)   N (new input file)   U (change units)
   enter option  { enter new reaction }
                                            <-------------- press return to enter
                                                             a new reaction


    NOTE:    OPTIONS

   Q  -   quit this program ( return to THERM MAIN menu )
   F  -   save this data to a file.
   N  -   change the thermo property polynomial data file (DAT).
   U  -    switch units on display from kcal/mol  (cal/mol  K) to
          KJ/mol (J/mol K) or visa versa.
```

```
             or enter Reaction expression here:
        C6H6 + O = C6H5O + H

       THERMODYNAMIC ANALYSIS for REACTION
  Rx           C6H6        + O          = C6H5O       + H
  Hf (Kcal/mol)   19.800      59.550      11.370      52.100
  S (cal/mol K)   64.240      38.470      72.730      27.390

    dHr    (298K) =      -15.880  Kcal/mol
  dU (dE) ( " )  =      -15.880  Kcal/mol
    dSr    ( " )  =       -2.590  cal/mol K
    dGr    ( " )  =      -15.108  Kcal/mol
   Af/Ar   ( " )  =    2.716E-01

       T (K)  dH(Kcal/mol)  dU(Kcal/mol) dS(cal/mol K)   (Af/Ar) dG(Kcal/mol)
     300.000   -1.587E+01   -1.587E+01   -2.567E+00   2.748E-01   -1.510E+01
     400.000   -1.550E+01   -1.550E+01   -1.502E+00   4.695E-01   -1.490E+01
     500.000   -1.516E+01   -1.516E+01   -7.347E-01   6.909E-01   -1.479E+01
     600.000   -1.485E+01   -1.485E+01   -1.703E-01   9.179E-01   -1.475E+01
     800.000   -1.435E+01   -1.435E+01    5.517E-01   1.320E+00   -1.479E+01
    1000.000   -1.401E+01   -1.401E+01    9.331E-01   1.599E+00   -1.495E+01
    1200.000   -1.381E+01   -1.381E+01    1.124E+00   1.761E+00   -1.516E+01
    1500.000   -1.365E+01   -1.365E+01    1.245E+00   1.871E+00   -1.551E+01
    2000.000   -1.356E+01   -1.356E+01    1.298E+00   1.922E+00   -1.615E+01

  Q (quit)   F (file)   N (new input file)   U (change units)
     enter option  ( enter new reaction ) <-----------------------press return


             enter Reaction expression here:

  C6H6 + O = C6H5. + OH

       THERMODYNAMIC ANALYSIS for REACTION
  Rx           C6H6        + O          = C6H5.       + OH
  Hf (Kcal/mol)   19.800      59.550      81.350       9.500
  S (cal/mol K)   64.240      38.470      68.920      43.880

    dHr    (298K) =       11.500  Kcal/mol
  dU (dE) ( " )  =       11.500  Kcal/mol
    dSr    ( " )  =       10.090  cal/mol K
    dGr    ( " )  =        8.492  Kcal/mol
   Af/Ar   ( " )  =    1.605E+02

       T (K)  dH(Kcal/mol)  dU(Kcal/mol) dS(cal/mol K)   (Af/Ar) dG(Kcal/mol)
     300.000    1.150E+01    1.150E+01    1.010E+01   1.611E+02    8.473E+00
     400.000    1.159E+01    1.159E+01    1.036E+01   1.834E+02    7.448E+00
     500.000    1.161E+01    1.161E+01    1.041E+01   1.888E+02    6.408E+00
     600.000    1.159E+01    1.159E+01    1.037E+01   1.845E+02    5.369E+00
     800.000    1.143E+01    1.143E+01    1.014E+01   1.643E+02    3.317E+00
    1000.000    1.116E+01    1.116E+01    9.845E+00   1.418E+02    1.318E+00
    1200.000    1.084E+01    1.084E+01    9.550E+00   1.223E+02   -6.212E-01
    1500.000    1.029E+01    1.029E+01    9.140E+00   9.946E+01   -3.423E+00
    2000.000    9.284E+00    9.284E+00    8.564E+00   7.444E+01   -7.844E+00
```

SECTION IV

FITTING / EXTRAPOLATION METHODS FOR THE HEAT CAPACITY OF AN IDEAL GAS

EXTRAPOLATION & INTERPOLATION OF Cp(T) FOR AN IDEAL GAS USING AN EXPONENTIAL FUNCTION

We have developed a linear relationship with respect to temperature which can be used to interpolate/extrapolate Cp(T) estimates obtained with group additivity to cover the range from 300 - 5000 K. Data is generated in a piecewise fashion with an maximum error near 5% when compared to literature data. Group additivity estimates between 300 and 1000 K are all that are required. These Cp estimates can then be used to generate thermodynamic property expressions for use with CHEMKIN and other codes requiring thermo property functions in NASA polynomial format.

We start with the following approximation for Cp:

$$Cp(T) = Cp_{inf} (1 - \exp (- BT)) \quad (1).$$

Where B is a parameter obtained by regression or estimated by solving for B using known Cp data at 298 or 300 K. Cp_{inf} is the high temperature limiting heat capacity. This expression may be transformed to the following linear expression in temperature:

$$Y = - \ln (1 - Cp(T)/Cp_{inf}) = BT \quad (2).$$

Equation (2) may be used to obtain rapid estimates of Cp(T) over the entire T range (100 - 5000 K) with a typical maximum error near 10%. This is the best one can obtain with a one parameter model, as will be shown. Only a single Cp estimate at one temperature (298 or 300 K) is required. Accuracy may be improved slightly if more data is available for regression (see figure 4.1), but for significant improvment another parameter is required.

Equation 1 may be thought of as a simplification of the equation of Yuan and Mok<6,7> who developed the following four parameter exponential function to describe the temperature dependence of an ideal gas:

$$Cp(T) = A + B \exp (C / T^{n}) \qquad (3)$$

where A, B, C, and n are the parameters determined by non-linear regression.

$$\text{If we define } A = -B = Cp_{inf} = \begin{cases} (3N-2)R & \text{non-linear molecule} \\ (3N-1.5)R & \text{linear molecule} \end{cases}.$$

and define n = -1 then equation (3) reduces to equation (1).

We can improve the predictive ability of equation (2) by adding a parameter. Equation (2) now becomes the following:

$$Y = - \ln (1 - Cp(T)/Cp_{inf}) = A + BT \qquad (4).$$

This expression can yield estimates within 6% over the temperature range 300 - 5000 K given only 2 reliable data points as close together as 100 K. Data must be at low temperature (300 or 400 K) for the expression to yield accurate values.

At this point it should be noted that the error between equations (3 or 4) and literature is not random. Predictions are either above or below the literature values across the entire temperature range. Since we are interested in the integral of the heat capacity for the enthalpy function (H(T)-H^{o}(T)), we must see if the integrated error is acceptable. We note that the integrated error (or the error in the enthalpy function) is a better measure of accuracy than the percent error in Cp at any given temperature. The integrated error increases the limits of uncertainty on calculated property changes for reaction at elevated temperatures.

Figures 4.2 (a - j) compare Cp, Hf, and S estimates for ethylene oxide using equation (4) and data at 300, 400, 500, 600, 800, and 1000 K that was estimated using group additivity (the THERM program). The error at 2000 K and 3000 K are -2.5 KJ/mol (1.2%) and -5.25 KJ/mol (1.5%) respectively for (H(T)-H(298K)); -1.75 J/mol K (1.4%) and -2.75 J/mol K (1.2%) respectively for (S(T)-S(298K)); and -2.5 J/mol K (1.8%) and -2.5 J/mol K (1.67%) respectively for Cp. Results obtained using the harmonic oscillator equation (HOE) are shown figures 4.1 (a-j) for comparison. This method will be discussed in detail later.

If one plots Cp data from the JANAF tables<8> in the form - ln (1 - Cp(T)/Cp_{inf}) vs. T, it can be seen that this is not really a linear relationship (see figure 4.3). Even so, a chord drawn between data points at 300 an 400 K and projected to 5000 K (equation 4) results in less than 2% error when transformed to Cp(5000 K). Equation (4) can be used to linearly interpolate smooth data at 100 K (or wider) temperature intervals with remarkable accuracy. Data points generated using Benson's group additivity method (the THERM program) may be used to generate Cp estimates in a piecewise fashion using equation (4). Typical maximum error from this method is less than 4 %.

We determined that extrapolation above 2000 K is best achieved by assuming that Cp(20 000 K) = 99.999% Cp_{inf}. This temperature (20 000 K) has no physical significance, and is only used as a tie point to improve extrapolations.

Heat capacity data are estimated at 1500 and 2000 K using an empirically derived relationship which relates the heat capacity at these temperatures to that at 1000K and Cp_{inf}. The following relationships have been determined empirically to approximate Cp at 1500 K and 2000 K, and typically result in less than 5% error:

$$Cp(1500K) = Cp(1000K) + 0.37(Cp_{inf} - Cp(1000K)) \quad (5)$$

$$Cp(2000K) = Cp(1500K) + 0.37(Cp_{inf} - Cp(1500K)) \quad (6).$$

These estimates along with the tie point at 20 000 K are used to extrapolate equation (4) to 6000 K and generate data required to create NASA polynomials.

A computer program has been developed to perform Cp extrapolation and polynomial determination using estimates obtained from THERM and the method described above. The program is called THERMFEX (THERM Fit EXponential), and a listing appears in appendix IV-C.

The method for polynomial fitting is discussed in the next section.

GENERATION OF THERMODYNAMIC PROPERTY DATA IN POLYNOMIAL FORM:

The NASA polynomial format has become a standard method for expressing thermodynamic properties. It is well known that 3rd and 4th order polynomials do a rather poor job of fitting the heat capacity function for an ideal gas over an extended temperature range (300 - 5000 K).<9> In 1967 McBride and Gordon, at NASA,<4> developed a method to fit Cp data using two 4th order polynomials which are tangent at 1000K, the break point temperature (T_{bk}). Fourteen coefficients are determined simultaneously. Ten coefficients are for Cp. Five are for the low temperature region ($T<T_{bk}$) and five for the high temperature region ($T>T_{bk}$). The remaining four coefficients are used for Hf and S over the two temperature regions. In this method H, S, and Cp data are regressed simultaneously. The regression is constrained so that Hf, S, Cp, and their first derivatives match at 1000 K, the break point temperature (T_{bk}).

A second and more efficient method was developed by Hanson and Haskell at Sandia National Laboratory in 1978 to fit thermo property data and generate polynomials in the NASA format.<2,3> The chief difference with their method is that the 10 Cp coefficients are determined first, and then the 4 remaining coefficients (integration constants for Hf and S) are determined using Hf and S data at 298 K only. This method is also a constrained linear regression which must match the Cp its first derivative at 1000 K or some other break point temperature.

A new method is developed here to fit Cp and generate thermo property polynomials in NASA format using group additivity estimates obtained from the program THERM. The methods discussed above perform constrained linear regression to simultaneously determine 14 or 10 parameters for two polynomials which are tangent at a predetermined T_{bk}. Our method, on the other hand, performs two unconstrained linear regressions to determine 5 Cp coefficients at a time and then solves for the point of tangency. In order for this method to be successful, the data for high and low temperature polynomials must overlap. We generate 45 data points between 300 and 6000 K using estimates from THERM and either an harmonic oscillator equation, developed in the next section, or an exponential function, developed in the previous section. The temperature range is divided into 3 regions. For example the temperature range may be divided into 15 points between 273 and 1000 K (group A), 15 points between 1000 and 1500 K (group B), and 15 points between 1500 and 6000 K (group C). Two unconstrained linear regressions are performed and two polynomials determined. Regression of data groups A and B gives the low temperature polynomial, while regression of data groups B and C gives the high temperature polynomial. A point of tangency is then determined in the region described by data group B (the overlap region) using the Newton/Raphson method.

The coefficients for H_f and S are determined from properties at 298 K in a fashion similar to that of Hanson.<2,3,10>

Properties at T_{bk} are matched to within 0.005% error and their temperature derivatives are matched to within an absolute tolerance of 0.005.

In the event that T_{bk} cannot be determined to satisfy the tolerance for property values and derivatives, the overlap region is moved and/or weighted differently (a different number of data points are used in that region) and the regressions are repeated. In most cases T_{bk} is determined on the first iteration making this a very effective and efficient regression method when sufficient data is available or when it can be extrapolated.

Figure 4.4 compares the polynomials with the Cp estimates used in their determination. The break point temperature is shown at a point of tangency and labeled T_{bk}. For temperatures below T_{bk} the low temperature coefficients are used and above T_{bk} the high temperature coefficients are used. Figures 4.5 (a-f) compare the polynomial fit of Cp data, enthalpy function ($H(T)-H^o(T_{ref})$), and entropy function ($S(T)-S^o(T_{ref})$) extrapolated from group additivity estimates with literature data for biphenyl radical. Figures 4.6 (a-f) are similar comparisons for phenoxy radical, while figures 4.7 (a-b) are comparisons of Cp(T) for furan. Also shown for comparison in these figures are the extrapolations obtained using the harmonic oscillator equation which will be discussed in the next section.

This method has been compared with literature for more than 40 species and in all cases agreement is excellent with typical maximum error less 5%.

We have developed a computer code which takes Hf, S, and Cp estimates from group additivity (using the computer program THERM), extrapolates them to high temperature, and then uses the method described above to generate thermo property functions in polynomial form. The extrapolation methods are developed in other sections. One computer program is called THERMFEX, and uses an exponential function to generate the required Cp data for regression (listing appears in appendix IV-C). The second program is called THERMFIT and uses an harmonic oscillator equation (HOE) to generate the required Cp data for regression (listing appears in appendix IV-D). An example of the usage of these programs appears in the THERM Users Manual in appendix III-C.

Table IV.ii shows the polynomial output in NASA compatible format phenoxy radical and dimethyl ether. Although this format can be read by CHEMKIN and other codes which allow T_{bk} to vary from one species to the next, there are several small modifications which have been incorporated for use with the programs developed here. Table IV.i explains what information is contained in these polynomials column by column for each of the four lines required per species. The chief modification to the standard format used by CHEMKIN is that the number of rotors in a molecule appears in column 79 on the first line of data for each species. In addition, the first two lines in the file must be padded with blanks to column 80 to allow direct access for use with the program THERMRXN (thermodynamic property analysis for reaction) which is described in the THERM Users Manual (see appendix III-C).

Table IV.i gives the polynomial equations for H(T), S(T), and Cp(T) with coefficients as described in table IV.ii.

HEAT CAPACITY DATA FROM STATISTICAL MECHANICS:

The heat capacity (Cp) of an ideal gas can be calculated from spectroscopic data using standard formulas from statistical mechanics.<5,11,12,13,14>

Each degree of freedom in a molecule makes a contribution to the total heat capacity (Cp_t). The translational contribution to the heat capacity at constant volume ($Cv_t(T)$) is given by the following expression:

$$C_{trans} = 5/2 \ R \tag{7}$$

where R is the universal gas constant. Rotational contributions are different for linear and non-linear molecules. The following expressions are used for rotational contributions to Cv_t:

$$C_{rot} = R \text{ (for linear molecules)} \tag{8a}$$

$$C_{rot} = (3/2)R \text{ (for non-linear molecules)} \tag{8b}$$

The temperature dependence for the heat capacity of an ideal gas arises from the vibrational contribution alone for molecules with no hindered internal rotors and negligible electronic contribution. The vibrational contribution must be summed over the number of vibrational degrees of freedom (S) in a molecule and is given by the following expressions:

$$C_{vib} = R \cdot \sum_{i}^{S} f(u_i,T)$$

$$\text{where} \quad f(u_i/T) = (u_i/T)^2 \frac{\exp\ (\ u_i/T\ )}{(\exp\ (u_i/T)\ -\ 1\ )^2} \tag{9}$$

$S = 3\ N - 6$ for non-linear molecules or

$= 3\ N - 5$ for linear molecules.

N = the number of atoms in the molecule.

$$u_i = (h\ c\ w_i\ /\ k\ ) \qquad (10)$$

where h is Plank's constant, c is the speed of light, k is Boltzman's constant, and w_i is the vibrational wavenumber (cm^{-1}) for the i^{th} vibrational mode (degree of freedom) in the molecule.

The total heat capacity at constant volume ($Cv_t(T)$) is then given as the sum of the individual contributions as follows:

$$Cv_t(T) = C_{trans} + C_{rot} + C_{vib} + C_{elec}$$

where C_{trans}, C_{rot}, C_{vib} are the translational, rotational, and vibrational contributions to Cv. The electronic contribution (C_{elec}) can typically be neglected.

The heat capacity at constant pressure can then be written as follows:

$$Cp_t(T) = Cv_t + R = R + C_{trans} + C_{rot} + C_{vib} \qquad (11)$$

$$= 4\ R + C_{vib} \quad (\text{non - linear molecule}) \quad (12a)$$

$$= (7/2)R + C_{vib} \quad (\text{linear molecule}) \quad (12b)$$

Using equation 9, one can calculate C_{vib}. Equations 11 and 12 can be used only for simple polyatomic molecules that have no contribution from free or hindered internal rotations (such as those found in dimethyl acetylene (free) or ethane (hindered)), and where electronic and anharmonic effects can be neglected.

The limit of equation 9, as temperature (T) tends to infinity is unity. Each vibrational mode, therefore, contributes a maximum of R (gas constant) to Cp_t. The high temperature limiting heat capacity (Cp_{inf}) is therefore obtained from equation 12 as follows:

$$\begin{aligned} Cp_{inf} &= R\ (\ 3\ N - 2\ ) \quad \text{(non-linear molecule)} \quad (13a) \\ &= R\ (\ 3\ N - 1.5\ ) \quad \text{(linear molecule)} \quad (13b). \end{aligned}$$

These expressions must be modified to account for free internal rotations if they need to be considered. Each free rotor (a functional group which may rotate with no potential barrier) contributes 1/2 R to Cp_t at the expense of one vibrational degree of freedom. For a molecule having one or more free rotors, equations 11 - 13 become the following:

$$\begin{aligned} Cp_t(T) &= Cv_t + R = C_{trans} + C_{rot} + C_{vib} + (N_{frot}/2)R \quad (14) \\ &= 4\ R + C_{vib} + (N_{frot}/2)R \ \text{(non-linear molecule)} \quad (15) \\ Cp_{inf} &= R\ (3N - (\ 2 + N_{frot}/2\ ))\ \text{(non-linear molecule)} \quad (16) \end{aligned}$$

Equation 9 is used to calculate the vibrational contribution to Cp_t, except that $S = 3N - (6+N_{frot}))$ where N_{frot} is equal to the number of free rotors in the molecule.

The situation is significantly complicated when hindered internal rotations need to be considered.<13> A detailed discussion of the contribution of hindered internal rotations to Cp_t is beyond the scope of this work.

The interested reader is referred to the work of Pitzer,<15> Pitzer and Gwinn,<16> Pitzer and Brewer,<17> Price,<18> and Crawford.<19> Detailed consideration of hindered internal rotations requires the solution of the Schrodinger wave equation given the appropriate potential function for internal rotation. Pitzer and Gwinn<16> have performed a detailed analysis on this problem and correlated the contributions to thermodynamic properties from hindered internal rotations with the maximum potential barrier (V/RT) and the reciprocal of the rotational partition function for free rotation ($1/Q_f$). These tables have been widely used to estimate the contribution to Cp_t from hindered internal rotors.

An example is now presented to illustrate the calculation of Cp(T) for a molecule where internal rotations need to be considered:

Hexafluoroethane has eight atoms and one hindered internal rotation. Using equation 8 and assuming that the hindered rotation acts essentially as a free rotor in the high temperature limit (this is a very good assumption) we find that Cp_{inf} = 178.7 KJ/mol. Hexafluoroethane has 17 vibrational degrees of freedom and one hindered internal rotation. The JANAF Tables<8> list the frequencies for the 17 oscillators and gives the maximum potential barrier to internal rotation (V) to be 16.57 KJ/mol. Using equations 7-16 we can obtain the vibrational contribution to Cp_t (see appendix IV-E for this calculation). A computer program named Cpideal was written to facilitate these calculations. A listing appears in appendix IV-E. To apply the method of Pitzer and Gwinn it is necessary to calculate the reduced moment of inertia for rotation about the C/C bond in this molecule. The moment of inertia for rotation of a CF_3 group about the C - C axis is given by the following expression, with bond distances and bond angle from the JANAF Tables:

$$I_{CF3} = \frac{3 \text{ x } 19. \text{ x } (1.33 \sin 68.5^{o})^{2}}{6.023 \text{ E+23 x } 1.0 \text{ E+16}}$$

$$= 144.9 \text{ E-40 gm cm}^{2}$$

The reduced moment of inertia is given by the following expression:

$$1/I_r = 1/I_a + 1/I_b \qquad (17)$$

where I_a and I_b are the moments of inertia for each group. In the case of hexafluoroethane $I_a = I_b = I_{CF3} = 1.449$ E-38 gm cm^2. Equation 11 then gives $I_r = 7.32$ E-39 gm cm^2. The partition function for free rotation is given by the following expression:

$$1/Q_f = \frac{h\,\sigma_r}{(8\,\pi^{3}\, I_r\, k\, T\,)^{1/2}} = \frac{n}{2.7935 \text{ E+19 } (I_r\, T)^{1/2}} \qquad (18)$$

where n is the number of potential maxima which occur during one rotation of the group (for CF_3/CF_3, n = 3). From equation 12 we obtain $1/Q_f = 0.07247$ at 300 K. Using the tables of Pitzer and Brewer <17> we obtain a contribution to Cp from internal rotation of the CF_3 group (at 300 K) equal to 9.414 J/mol. The Cp for C_2F_6 at 300 K is then given as the sum of contributions from each degree of freedom in the molecule or 106.7 J/mol. This is in excellent agreement with the tabulated value of 106.8 J/mol given in the JANAF Tables. The 0.1 J/mol difference is due to interpolation error for values in the Pitzer/Brewer tables. This calculation is presented in more detail with results tabulated in appendix IV-E. A similar calculation for internal rotational contribution to Cp for biphenyl is also presented in appendix IV-E. These examples are presented to illustrate the current literature method used to calculate Cp for an ideal gas.

In the next section, a model is developed which will allow accurate extrapolation of low temperature heat capacity estimates from group additivity or available experimental data (< 1000 K) to high temperatures (5000 K). While, this model is based upon the principles of statistical mechanics which have been reviewed here, it does not require input of moments of inertia or barriers to internal rotation. All that is required is low temperature Cp data and the number of rotors.

DEVELOPMENT OF A Cp EQUATION BASED UPON STATISTICAL MECHANICS:

For molecules where no internal rotations need be to considered, equations 1 - 10 show us that the only temperature dependent term for contributions to Cp is that for vibrations (C_{vib}) given by equation 3. One may attempt to fit Cp data using a single term from equation 3 and some mean vibrational frequency (w_{mean} {cm^{-1}}). In this case the summation is replaced by the constant factor S (the number of oscillators) as follows for non-linear molecules:

$$Cp(T)/R = 4 + S \cdot (u_{mean}/T)^2 \frac{\exp (u_{mean}/T)}{(\exp (u_{mean}/T) - 1)^2} \qquad (19)$$

It can be easily shown that equation 19 gives generally poor results even if u_{mean} is determined by fitting reliable Cp data. The reason for this is that the contribution to Cp from each vibrational mode is a function of frequency and temperature. The relative contribution from any one vibrational frequency varies with temperature. The mean frequency which reproduces Cp at one temperature is different from that required to fit Cp at another temperature. Therefore, it is not surprising that a single mean vibrational frequency does a poor job of representing the complex behavior of Cp as a function of temperature. Figures 4.8 and 4.9 illustrate how poorly the geometric and arithmetic mean frequencies reproduce Cp(T) using equation 19 for cyclopentadiene when compared with the data of Dorafeeva, et al.<20>

In order to improve this calculation, one may attempt to increase the number of terms (n_t) considered. Equation 19 can then be rewritten as:

$$Cp(T)/R = 4 + \frac{S}{n_t} \sum_{i}^{n_t} (u_i/T)^2 \frac{\exp(u_i/T)}{(\exp(u_i/T) - 1)^2} \tag{20}$$

It can be shown that equation 14 is only able to fit the Cp behavior in the limit as the number of terms considered (n_t) approaches the number of oscillators in the molecule. For cyclopentadiene, which has 27 vibrational modes, we may require nearly 27 parameters (or at least half that many) to fit Cp using equation 20. In this limit, equation 2.0 reduces to equation 3. Since vibrational frequency data is typically not readily available for many molecules whose Cp are required, equations 9 and 20 offer no solution to fitting or extrapolation of Cp data.

The solution to this problem is to replace the summation over the number of vibrational degrees of freedom in the molecule with a 3 term expression based upon equation 3, but with weighting factors as coefficients to these terms. These coefficients need not have integer values.

We have chosen a three term equation as follows (for a non-linear molecule):

$$Cp(T)/R = 4 + A_1 \cdot f(u_1/T) + A_2 \cdot f(u_2/T) + A_3 \cdot f(u_3/T) \tag{21}$$

where $f(u_i/T)$ is given by equation 3 with u_i and A_i determined from non-linear regression using the method of Marquardt<21>. Equation 21 is a six parameter non-linear model for Cp(T) for any non-linear molecule having no internal rotations.

Making use of the fact that the coefficients A_i must sum to S (the number of oscillators) we can reduce equation 21 to a 5 parameter expression which we call an Harmonic Oscillator Equation (HOE):

$$\boxed{Cp(T)/R = 7/2 + A_1 \cdot f(u_1/T) + A_2 \cdot f(u_2/T) + (S-(A_1+A_2)) \cdot f(u_3/T)}$$

Harmonic Oscillator Equation (HOE) (22a)
linear molecule

$$\boxed{Cp(T)/R = 4 + A_1 \cdot f(u_1/T) + A_2 \cdot f(u_2/T) + (S-(A_1+A_2)) \cdot f(u_3/T)}$$

Harmonic Oscillator Equation (HOE) (22b)
non-linear molecule

This equation is based upon the harmonic oscillator model for vibrational contribution to the heat capacity of an ideal gas. This equation has been found to give an excellent fit for Cp data across the temperature range 300 - 6000 K. Equations 22 have the fundamentally correct functional form for the heat capacity of an ideal gas. This allows Cp data to be extrapolated from low temperature to high temperature with a high degree of accuracy. Figure 4.10 illustrates how well this expression fits Cp data for cyclopentadiene. The solid line is the Cp function described by equation 22b. Only six data points between 300 and 1000 K in addition to the high temperature limiting heat capacity Cp_{inf} (described by equation 13b) are required to determine parameters for equations 22. Figures 4.11 (a-b) demonstrate how well equation 22b does in extrapolating Cp data to high temperature for benzene. Again, only data at 300, 400, 500, 600, 800, and 1000 K were used to determine the parameters of equation 22b. The data points labled ( ) in figure 4.11a are those of Burcat<1>. Those points labled (*) are the group additivity data points obtained with the THERM computer program and used to project the Cp to high temperature. There is excellent agreement between extrapolated Cp values and those from literature. The maximum error is less than 0.3% when Cp data from group additivity are used, and less than 0.05 % when Burcat's Cp data at 300, 400, 500, 600, 800, and 1000 K are used for the regression (see figure 4.11c).

Equation 22b implies that the 30 non-degenerate vibrational frequencies of benzene can be represented as the weighted average of only three frequencies. Term 1 represents the low frequency vibrations, term 2 represents the mid frequency vibrations, and term 3 represents the high frequency vibrations occurring in a molecule.

The use of equation 22b for molecules which contain free or hindered internal rotations requires only a small modification to the high temperature limiting heat capacity and the number of oscillators.

All hindered rotations are considered to become free rotors in the high temperature limit. Few exceptions to this rule have been found. Three molecules which violate assumptions of equations 22 are now presented to illustrate the maximum errors encountered when these equations are used to extrapolate Cp data.

Hexachloroethane has a severely hindered rotation with a potential barrier in excess of 42 KJ/mol.<8> Rigorous calculation of the contribution of this hindered rotation to Cp for C_2Cl_6 show a maximum in Cp(T) near 2000 K.<8> This maximum is not predicted by equation 22b, however, the maximum error in neglecting this is less than 4% (see figures 4.12 a-f). Two additional molecules which do not adhere to equations 22 are acetylene (which requires significant anharmonic corrections) and ammonia (which undergoes inversion between two pyramidal conformations). In these cases, equation 22b produces less than 5% error in high temperature Cp extrapolations (see figures 4.12-4.14). Acetylene, however (equation 22a), has larger errors due to the required corrections to the rigid rotator harmonic oscillator model for this molecule. Acetylene is a unique molecule in this respect; ethynl radical and substituted acetylenes produce no such violation of the HOE model.

Hexafluoroethane, which has a more typical barrier to internal rotation, is fit with a maximum error of 0.3% (see figures 4.15 a-f).

The required modifications to equation 16 for Cp_{inf} have been determined empirically as follows:

$$Cp_{inf} = R\ (3N - (\ 2 + N_{rot}/4\ ))\ \text{(non-linear molecule)}\quad (23)$$

$$\text{with } S = 3\ N - (6 + N_{rot}/2)\quad \text{(non-linear molecule)}\quad (24)\ .$$

Equations 23 and 24 sacrifice some realism for the sake of a more generally applicable expression. Each internal rotation is weighted as 1/2 of a vibrational degree of freedom instead of the theoretically correct value of 1 vibrational degree of freedom. Statistical mechanics tells us that Cp_{inf}/R should be reduced by 1/2 for each free rotor. Equation 23, however, yields results which are more consistent with available experimental data. It is recognized that the use of equations 23 and 24 will result in a slight overestimate (typically less than 1%) for molecules which contain truly free internal rotations.

Cp extrapolations have been compared with literature for more than 50 species using this method. Agreement is generally excellent, with maximum errors less than 0.1% when the literature Cp data are used for the non-linear regression.

As stated earlier, the motivation to develop an extrapolation method for Cp data arises from the need to determine thermodynamic properties in polynomial form which are valid outside the temperature range for which Benson's group estimates are available. Equations 22 - 24 provide such a method. In figures 4.16 - 4.18 (a-f), Cp, Hf, and S data estimated using the THERM program (extrapolated with equation 22b) are compared with data from Burcat, et al<1> for phenyl radical, phenol, and biphenyl. Here we are comparing data from two different sources, and the agreement is excellent.

An important and valuable benefit to the use of equations 22 for Cp data extrapolation is that the parameters determined from regression have physical significance. The parameters obtained from fitting Cp data can be used to estimate the vibrational distribution, geometric and arithmetic mean frequencies, the density and sum of vibrational quantum states, and the vibrational partition function for molecules. These applications will be discussed further in the next section.

A computer program called CPFIT has been developed to apply equations 22 to fit Cp data for linear as well as non-linear molecules (with or without hindered internal rotations). The code employs Marquardt's method <21> for the non-linear regression. A numerically determined Jacobian matrix has been replaced with an explicit matrix of partial derivatives to substantially improve computational efficiency (source code listing appears in appendix IV-F).

The code does not require any estimate for potential barriers or reduced moments of inertia for molecules having hindered internal rotations, however, we still need accurate low temperature data.

The chief weakness for this method is the sensitivity to the initial parameter guesses and parameter constraints. Molecules which contain only low frequency oscillators, such as CCl_4, will not be fit well if initial parameter values contain a high frequency oscillator. Regression will converge to a solution which is in significant error (as much as 25%). If calculated frequencies are too high, the error will be negative (Cp estimates will fall below Cp data). If, on the other hand, calculated frequencies are too low, the error will be positive (Cp estimates will be larger than the Cp data points). This fact is used to perform a systematic restart of the regression to insure an adequate fit. A new starting point and feasible region are determined if an adequate fit (max error < 4%) is not achieved with the first solution.

A listing of this computer program CPFIT appears in appendix IV-F. A listing of the program THERMFIT (which incorporates NASA polynomial determination) appears in appendix IV-D.

APPLICATION OF HARMONIC OSCILLATOR EQUATION PARAMETERS TO CALCULATIONS REQUIRING VIBRATIONAL FREQUENCIES

The harmonic oscillator equation (HOE) was shown to be an effective method to extrapolate heat capacity data from low temperature (< 1000 K) to high temperature (5000 K). In this section we shall demonstrate that group additivity and the HOE (equations 22) can be used to estimate properties and functions which require vibrational frequency data. We shall compare vibrational partition functions, sum of vibrational quantum states, and mean frequencies (geometric, arithmetic, and root mean square) estimated using HOE and Cp data from group additivity with those calculated from experimental frequencies available in literature. Unimolecular QRRK calculations performed with estimated geometric mean frequencies will be compared with those calculated using experimental mean frequencies as well as RRKM calculations from literature.

VIBRATIONAL PARTITION FUNCTION: We consider first the vibrational partition function which can be written as follows for a polyatomic molecule:<12,22>

$$Q_{vib}(T) = \sum_{i=1}^{S} \frac{1}{1 - \exp\{-hcw_i/kT\}} \qquad (25)$$

where h is Plank's constant, c is the speed of light, k is Boltzman's constant, T is absolute temperature, and w_i is the vibrational wavenumber for the ith vibrational degree of freedom.

Figure 4.19 compares the natural logarithm of the vibrational partition function for cyclohexa-1,3-diene estimated from Cp data with that calculated from experimental frequency data.<46> Also included for comparison, are estimates using arithmetic and geometric mean frequencies. The predictive ability of HOE is quite good. Maximum error for cyclohexa-1,3-diene is 1 unit in $\ln(Q_{vib})$ at 298 K using HOE frequencies determined from group additivity estimates for Cp. In comparison, the literature geometric mean frequency produces a maximum error of 3 units in $\ln(Q_{vib})$ at 1000 K, while the literature arithmetic mean produces a maximum error of 8 units at 6000 K.

Prediction of $\ln(Q_{vib})$ for molecules having hindered internal rotations is not necessarily improved by subtracting the contribution to Cp due to the hindered internal rotation before fitting with the HOE (equation 22b). For C_2F_6 we remove the internal rotational contribution from Cp (calculation of internal rotation contribution for C_2F_6 appears in appendix IV-E). Fitting the corrected Cp(T) we then obtain frequencies and degeneracies which produce similar error in $\ln(Q_{vib})$. Figure 4.20 demonstrates estimates of vibrational partition function for C_2F_6 with and without corrections for the hindered internal rotation are within the uncertainty for properties derived from group additivity values.

SUM AND DENSITY OF VIBRATIONAL QUANTUM STATES

The next calculation that we shall examine is the sum and density of vibrational quantum states. The sum ($W(E_{vib})$) and density ($N(E_{vib})$) of vibrational quantum states can be calculated using the Whitten-Rabinovitch modification to the Marcus-Rice semiclassical approximation.<22,23,24> This approximation can be written as follows:

$$W(E_{vib}) = \frac{(E_{vib} + a\,E_z)^{S}}{S\,!\,\prod_i hcw_i} \qquad (26)$$

and

$$N(E_{vib}) = \frac{(E_{vib} + a\,E_z)^{S-1}}{(S-1)\,!\,\prod_i hcw_i} \left[1 - B(du/dE') \right] \qquad (27)$$

where

$$E_z = \sum_{i=1}^{S} (hcw_i/2) \; ; \qquad E' = E_{vib}/E_z$$

$$a = 1 - B\,u\,(E') \; ; \qquad B = \frac{S-1}{S} \cdot \frac{<w^2>}{<w>^2}$$

$$<w^2> = (1/S)\cdot\sum_{i}^{S}(w_i{}^2) \; ; \qquad <w>^2 = \{(1/S)\cdot\sum_{i}^{S}(w_i)\}^2$$

$$u\,(E') = \begin{cases} (5.0\,E' + 2.73\,E'^{0.5} + 3.51)^{-1} & \text{for } 0.1<E'<1.0 \\ \exp\{-2.4191\,E'^{0.25}\} & \text{for } 1.0<E'<8.0 \end{cases}$$

$$du(E')/dE' = \begin{cases} -(5.00 + 1.365\ E'^{-0.5})\ u^2 & \text{for } (0.1<E'<1.0) \\ -(0.60478\ E'^{-0.75})\ u & \text{for } (1.0 < E' < 8.0) \end{cases}$$

Figure 4.21 compares $\ln\{W(E_{vib})\}$ calculated using literature frequencies with those calculated from HOE parameters obtained from fitting Cp data for cyclohexa-1,3-diene. For comparison $\ln\{W(E_{vib})\}$ estimated with geometric and arithmetic mean frequencies are also shown. As can be seen, HOE frequency parameters can be used to estimate this function with much higher accuracy than the geometric mean frequency. Hindered internal rotor effects need not be removed from Cp estimates in order to approximate these functions accurately. Figure 4.22 compares estimates based upon HOE parameters with calculations using literature frequencies for biphenyl. With or without hindered internal rotational contributions removed from group additivity Cp estimates, the HOE parameters which are obtained approximate the sum of vibrational quantum states rather well.

ESTIMATING GEOMETRIC AND ARITHMETIC MEAN VIBRATIONAL FREQUENCIES:

The geometric mean vibrational frequency is required for QRRK calculations presented throughout this thesis. An harmonic oscillator equation (equations 22) may be used to obtain an estimate of mean vibrational frequencies by fitting Cp data from THERM (group additivity). One uses the parameters of equations 22 to determine mean frequencies. The geometric (w_{gm}) and arithmetic (w_{am}) mean vibrational frequencies may be obtained using the following expressions:

$$\bar{w}_{gm} = \sqrt[S]{w_1^{A_1} \cdot w_2^{A_2} \cdot w_3^{(S-(A_1+A_2))}} \qquad (28)$$

$$\bar{w}_{am} = \frac{1}{S} \left[A_1 \cdot w_1 + A_2 \cdot w_2 + (S-(A_1+A_2)) \cdot w_3 \right] \qquad (29)$$

with parameters A_i and w_i described by equations 22. The computer program CPFIT performs these estimations (see appendix IV - F).

Table IV.iii compares estimated and literature mean frequencies for 50 species (more than 30 of which are based upon group additivity Cp estimates). Maximum error in estimating the geometric mean frequency is estimated as 50 cm^{-1} which is rarely greater than 10% error with error for most species within 2%.

Geometric mean frequencies obtained from literature range anywhere from 400 cm^{-1} for CCl_4 to near 2000 cm^{-1} for CH_4. Predictions in table IV.iii have a typical accuracy better than $\pm 25 cm^{-1}$ when compared with literature values for geometric mean frequencies. This is actually quite good. Let us consider how well literature frequency data from two sources compare. Frequency data may be obtained for furan (C_4H_4O) from Chao et.al.<25> or from Simandiras<26>. These sources give geometric mean's of

1221 and 1244.5 cm^{-1} respectively or a difference of nearly 25cm^{-1}. The estimate from THERM (group additivity) Cp data is obtained using the HOE as 1215 cm^{-1} which is closer to the value from Chao then the value from Simandiras is. This demonstrates that mean frequencies estimated in this manner have uncertainties on the same order as those derived from literature frequency data.

A comparison of mean frequencies estimated from literature Cp data with those estimated from group additivity Cp data shows that HOE parameters are only as reliable as the Cp estimates which one uses. Mean frequencies based upon group additivity data typically have larger errors than those based upon literature Cp data. Even so, group additivity Cp estimates do a remarkably good job in predicting the functions described in this section. It is our feeling that the HOE (equations 22) presents us with an easy and effective method to estimate mean vibrational frequencies from group additivity Cp data. QRRK calculations presented in other sections of this work utilized geometric mean frequencies estimated in this manner.

Unimolecular QRRK calculations using the method of Dean<27> are now presented for three species to illustrate how good of an approximation mean frequencies obtained using the HOE and Cp are.

The first case to be considered is that for unimolecular dissociation of SF_6 at 1750 K. Required input data come from Lyman<28> and appear in table IV.iv. Using Cp data from Benson<30> the geometric mean vibrational frequency is estimated from the HOE as 596/cm as compared to 591/cm given by Lyman. Figure 4.23 compares calculated fall-off behavior for this reaction using these two frequencies. Results are identical. QRRK results are shown to be in good agreement with the RRKM results of Lyman.

The second example will be unimolecular dissociation of phenyl radical (C_6H_5) at 1700 K. Required input data are taken from Rao and Skinner<29> and are presented in table IV.v. Using group additivity (THERM) and the HOE we estimate the geometric mean vibrational frequency for C_6H_5 to be 1204 cm^{-1} while that obtained from the data of Skinner is 1180 cm^{-1}. Figure 4.24, again, shows QRRK calculations using these two frequencies to be virtually identical. Skinner's RRKM calculations are also reproduced for comparison. QRRK using this estimated mean frequency matches Skinner's results rather well.

The final example presented is unimolecular dissociation of benzene (C_6H_6) to form phenyl + H atom. Group additivity and the HOE estimate the geometric mean frequency to be 1216 cm^{-1} while the data of Burcat give 1205.8 cm^{-1}. Again, required input parameters for QRRK are taken from Skinner and are presented in table IV.vi. Figure 4.25, shows once again that the estimated mean frequency reproduces fall-off results calculated with actual mean frequency. In addition, the QRRK results are in good agreement with Skinners RRKM calculations. Figures

4.23 - 4.25 not only show how well frequencies obtained from group additivity do when applied to QRRK calculations, but also how good the QRRK theory is at approximating the more rigorous RRKM calculations.

CONCLUSION: In this section we have shown that parameters obtained fór the Harmonic Oscillator Equation (HOE) by fitting estimated Cp data not only have physical significance, but can be used to estimate functions requiring vibrational frequency data with a high degree of success. The harmonic oscillator equation is a valuable tool allowing the extrapolation of low temperature (< 1000 K) Cp data to 5000 K and in addition giving frequencies and degeneracies which can be used further to approximate statistical mechanics functions for species where the vibrational frequency data are not available.

REFERENCES SECTION IV

1. Burcat, A., Zeleznik, F.J., and McBride, B.J., NASA Technical Memorandum # 83800, 1985.
2. Hanson, R. J. and K. H. Haskell, Sandia National Laboratories Report, SAND77-0552, 1978.
3. Hanson, R. J. and K. H. Haskell, Sandia National Laboratories Report, SAND78-1290, 1979.
4. McBride, B. J. and S. Gordon, NASA Technical Memo: NASA TN-D-4097, 1967.
5. Stull, D.R., Westrum, E.F.Jr., and Sinke, G.C., THE CHEMICAL THERMODYNAMICS OF ORGANIC COMPOUNDS, Robert E. Krieger Publishing Company, Malibar, Florida (1987).
6. Yuan, S. C. and Y. I. Mok, Hydrocarbon Processing, 47, 7, 1968.
7. Yuan, S. C. and Y. I. Mok, Hydrocarbon Processing, 47, 3, 1968.
8. JANAF Thermochemical Tables, J. Phys. Chem. Ref. Data, 14,1, 1985.
9. Thinh, T. P., J. L. Duran, R. S. Ramalho, and S. Kaliaguine, Hydrocarbon Processing, January 1971.
10. Kee, R.J., Rupley, F.M., and Miller, J.A., Sandia National Laboratories Report, SAND87-8215, 1987.
11. Rice, O. K., Statistical Mechanics Thermodynamics and Kinetics, W. H. Freeman and Company, San Francisco (1967).
12. Soontag, R. E. and G. J. Van Wylen, Fundamentals of Statistical Mechanics, Robert E. Krieger Publishing Company, Malabar, Florida (1985).
13. Herzberg, G., Molecular Spectra and Molecular Structure II. Infred and Raman Spectra of Polyatomic Molecules, Van Nostrand Reinhold Company, New York (1945).
14. Cox, J.D., and Pilcher, G., Thermochemistry of Organic and Organometalic Compounds, Academic Press, New York, 1970.
15. Pitzer, K. S., J. Chem. Phys., 5, pp 469-472, 473-479, 1937.
16. Pitzer, K. S. and W. D. Gwinn, J. Chem. Phys., 10, 1942.
17. Pitzer, K. S. and L. Brewer, Thermodynamics, McGraw-Hill Book Co., New York (1961).
18. Price, D., J. Chem. Phys., 9, 1941.
19. Crawford, B. L., J. Chem. Phys., 8, 1940.
20. Dorafeeva, O. V., L. V. Gurvich, and V. S. Jorish, J. Phys. Chem. Ref. Data, 15, 2, 1986.
21. Reklaitis, g.V., A. Ravindran, and K.M. Ragsdell, Engineering Optimization, John Wiley and Sons, New York (1983).
22. Troe, J., J. Phys. Chem., 83, 1, 1979.
23. Robinson, P. J. and K. A. Holbrook, Unimolecular Reactions, Wiley Interscience, New York (1972).
24. Forst, W., Theory of Unimolecular Reactions, Academic Press, New York (1973).
25. Chao, J., K. R. Hall, K. N. Marsh, and R. C. Wilholt,

J. Phys. Chem. Ref. Data, 15, 4, 1986.

26. Simandiras, E. D., N. C. Handy, and R. Amos, J. Phys. Chem., 92, 7, 1988.

27. Dean, A. M., J. Phys. Chem., 89, 21, 1985.
Dean, A. M. and P. R. Westmoreland, Int. J. Chem. Kin., 33, 1987.
Westmoreland, P. R., J. B. Howard, J. P. Longwell, and A. M. Dean, AICHE Journal, 32, 12, 1986.

28. Lyman, J. L., J. Chem. Phys., 67, 5 (1977).

29. Rao, V. S. and G. B. Skinner, J. Phys. Chem., (1988).

30. Benson, S., Thermochemical Kinetics, John Wiley and Sons, New York (1976).

APPENDIX IV - A

TABLES FOR SECTION IV

TABLE IV.i

POLYNOMIAL REPRESENTATION OF THERMODYNAMIC PROPERTY DATA

Heat capacity:

$$Cp(T) = R\ (\ a_1 + a_2T + a_3T^2 + a_4T^3 + a_5T^4\ ) \qquad T > T_{bk}$$

$$Cp(T) = R\ (\ a_8 + a_9T + a_{10}T^2 + a_{11}T^3 + a_{12}T^4\ ) \qquad T < T_{bk}$$

Enthalpy of formation:

$$T > T_{bk}$$

$$H(T) = R\ (\ a_1T + \frac{a_2}{2}\ T^2 + \frac{a_3}{3}\ T^3 + \frac{a_4}{4}\ T^4 + \frac{a_5}{5}\ T^5 + a_6\ )$$

$$T < T_{bk}$$

$$H(T) = R\ (\ a_8T + \frac{a_9}{2}\ T^2 + \frac{a_{10}}{3}\ T^3 + \frac{a_{11}}{4}\ T^4 + \frac{a_{12}}{5}\ T^5 + a_{13}\ )$$

Entropy:

$$T > T_{bk}$$

$$S(T) = R\ (\ a_1\ln(T) + a_2T + \frac{a_3}{2}\ T^2 + \frac{a_4}{3}\ T^3 + \frac{a_5}{4}\ T^4 + a_7\ )$$

$$T < T_{bk}$$

$$S(T) = R\ (\ a_8\ln(T) + a_9T + \frac{a_{10}}{2}\ T^2 + \frac{a_{11}}{3}\ T^3 + \frac{a_{12}}{4}\ T^4 + a_{14}\ )$$

TABLE IV.ii

NASA FORMAT EXPLAINED

```
THERMO
 300., 1000., 5000.
C6H5O     3/20/89 THERMC  6H   5O   1    0G   300.000  5000.000 1404.000    01
 1.57324764E+01 1.54394760E-02-5.31383756E-06 8.28591116E-10-4.82238305E-14    2
-1.76799222E+03-6.20052317E+01-3.73072513E+00 6.71840729E-02-5.88121307E-05    3
 2.61335718E-08-4.61968583E-12 4.31827564E+03 4.02213334E+01                   4
CH3OCH3   3/20/89 THERMC  2H   6O   1    0G   300.000  5000.000 1364.000    21
 8.43631162E+00 1.30235626E-02-4.45885307E-06 6.93233262E-10-4.02757540E-14    2
-2.62777965E+04-2.24530764E+01 1.53263200E+00 2.37669572E-02-8.14177550E-06    3
-5.29451929E-10 5.96605679E-13-2.32831259E+04 1.66340885E+01                   4
END
```

column numbers:

```
1        10        20        30        40        50        60        70        80
|        |         |         |         |         |         |         |         |
|23456789|123456789|123456789|123456789|123456789|123456789|123456789|123456789|

|<-name>||<-date->|<ref>|<---- elements --->G|<- Tlow >|<- Thi->|<- Tbk->|   21
|<--- a1 ---->|<---- a2 ---->|<---- a3 ---->|<---- a4 ---->|<---- a5 ---->|    2
|<--- a6 ---->|<---- a7 ---->|<---- a8 ---->|<---- a9 ---->|<---- a10 --->|    3
|<--- a11 --->|<---- a12 --->|<---- a13 --->|<---- a14 --->|                   4
```

- 4 lines per species
- number of rotors appears in column 79 of first line

line 1:

- species name column 1-9
- date : column 11-19
- reference: column 20-24
- elements : columns 26-44
- phase : column 45
- low temperature limit
- high temperature limit
- break point temperature
- number of rotors : column 79
- line number "1" for first line of species :column 80

line 2:

- coefficients 1-5 : columns 1 - 75
- line number "2" for second line of species :column 80

line 3:

- coefficients 6-10 : columns 1 - 75
- line number "3" for third line of species :column 80

line 4:

- coefficients 11-14 : columns 1 - 60
- line number "4" for fourth (last) line for this species :column 80

TABLE IV.iii

	LIT	Cpfit	LIT	Cpfit	source
SPECIES	w_{am}(cm^{-1})	w_{am}(cm^{-1})	w_{gm}(cm^{-1})	w_{gm}(cm^{-1})	Cp data
C_3H_6	1646.9	1648.4	1429.7	1430.5	(a)dorafeeva
"	"	1614.3	"	1415.1	(b)therm
CY-1,3-HD	1436.7	1442.3	1183.2	1209.7	a
	"	1449.1	"	1196.0	b
C_2F_6	745.77	725.0	637.7	668.9	c: therm (subtr. rotor)
"	"	637.0	"	601.3	d: therm (ignore rotor)
BIPHENYL	1308.5	1323.23	1038.4	1091.13	c
"	"	1266.65	"	1034.54	d
Cy-C_5H_{10}	1570.7	1540.76	1339.5	1324.5	c
"	"	1523.17	"	1294.10	d
Cy-C_4H_8O	1546.34	1524.12	1339.2	1345.6	c
"	"	1505.52	"	1317.43	d
C_6H_6	1426.2	1427.3	1205.8	1213.1	(e) Burcat
"	"	1415.48	"	1218.35	b
C_2H_4	1799.58	1798.57	1585.99	1586.1	(f) JANAF
"	"	1779.02	"	1578.3	b
C_6H_5Cl	1288.57	1291.65	1033.74	1057.72	(g) Skinner
"	"	1269.60	"	1054.32	b
Cy-C_2H_4O	1637.4	1636.7	1443.97	1443.07	f
"	"	1625.91	"	1440.03	b
Cy-C_3H_4	1573.9	1573.6	1345.2	1346.8	a
"	"	1575.4	"	1353.1	b
Cy-C_7H_{14}	1460.79	1467.87	1174.6	1215.9	a
"	"	1443.16	"	1205.26	b

SPECIES	LIT $W_{am}(cm^{-1})$	Cpfit $W_{am}(cm^{-1})$	LIT $W_{gm}(cm^{-1})$	Cpfit $W_{gm}(cm^{-1})$	source Cp data
C_6H_5Br	1280.97	1285.18	1013.53	1044.32	g
"	"	1273.72	"	1052.36	b
C_2HCl	1152.29	1154.47	798.27	823.48	f
"	"	1134.59	"	795.04	b
C_2Cl_2	660.71	611.82	441.70	401.55	f
"	"	598.82	"	352.30	b
Cy-1,3-C_8H_{12}	1404.6	1412.09	1108.14	1152.73	a
"	"	1404.37	"	1150.5	b
Cy-1,5-C_8H_{12}	1401.85	1408.27	1107.95	1149.08	a
"	"	1373.52	"	1142.84	b
Cy-C_7H_8	1405.77	1409.22	1139.56	1162.22	a
"	"	1408.66	"	1171.82	b
C_2F_2	799.86	801.42	635.32	644.34	f
Cy-C_4H_4	1455.28	1452.75	1239.8	1240.26	a
Cy-C_4H_4S	1353.7	1351.74	1131.97	1132.69	(h) Simandiras
Cy-C_4H_5N	1465.0	1462.8	1231.4	1233.0	"
C_2HF	1223.43	1226.02	865.26	883.39	f
C_2F_4	778.17	784.02	619.62	646.87	"
$SiBr_4$	255.46	256.96	199.91	202.07	"
SF_6	623.6	631.51	591.0	598.1	(i) Lyman
SiH_2Cl_2	989.78	1048.39	777.55	840.64	f
SiH_2F_2	1112.44	1116.47	953.99	970.05	"
SiH_4	1493.0	1494.1	1365.5	1366.0	"
CH_3F	1865.3	1862.2	1708.0	1700.3	"
$CHCl_3$	951.62	953.59	727.99	728.73	"

SPECIES	LIT $w_{am}(cm^{-1})$	Cpfit $w_{am}(cm^{-1})$	LIT $w_{gm}(cm^{-1})$	Cpfit $w_{gm}(cm^{-1})$	source Cp data
CHF_3	1216.0	1217.1	1051.12	1053.68	"
H_2SO_4	1107.3	1110.6	788.01	807.4	"
CH_4	2106.5	2128.7	1956.74	1958.45	"
N_2H_4	1736.25	1741.86	1387.12	1402.95	f
CCl_3F	549.09	553.08	479.5	429.6	"
CCl_4	462.7	460.5	408.2	401.9	"
CF_4	835.6	828.8	766.36	757.4	"
CH_2Cl_2	1398.0	1399.9	1116.3	1141.8	"
CH_2F_2	1564.2	1570.3	1381.78	1391.10	"
$B_3H_2O_3F$	992.38	996.51	796.96	820.04	"
$B_3O_3Cl_3$	596.52	603.93	456.34	492.25	"

TABLE IV.iv

```
 SF6 DISSOCIATION Re: LYMAN jcp 67 P1808 1977
****DISSOCIATING SPECIES****
 SF6
****PRODUCT CHANNEL 1 (LOWEST ENERGY)****
 SF5 + F
****PRODUCT CHANNEL 2 ****

****PRODUCT CHANNEL 3 ****

***GEOMETRIC MEAN FREQUENCY;    # OF OSCILLATORS (S)*****
         591.0                          15
***COLLISION COMPLEX MASS;    DIAMETER;     WELL DEPTH*****
           146.0              5.128,        222.1
***THIRD BODY (COLLISION PARTNER)*****
 AR
***THIRD BODY MASS;   DIAMETER;   WELL DEPTH;    ENERGY TRANSFERRED******
         40.0,           3.54,     93.3,              500.
****COLLISION COEFFICIENT****
       0.
****CHANNEL 1:  HIGH PRESSURE A FACTOR;  EA(KCAL/MOL)*****
                  1.2E+16               92.0,
****CHANNEL 2:  HIGH PRESSURE A FACTOR;  EA(KCAL/MOL)*****
                   0.0,                  0.0
****CHANNEL 3:  HIGH PRESSURE A FACTOR;  EA(KCAL/MOL)*****
                     0.0,                0.0
                                                                          53
NUMBER OF TEMPERATURES   T1      T2      T3     T4      T5      T6 (K)    54
4,                       1000., 1300.,  1500,  1750.,
                                                                          56
NUMBER OF PRESSURES      P1      P2      P3     P4     P5     P6 (TORR)
6,                      2.0,    10.,    100.,  1000., 10000., 100000.,

Tout (FOR DISTRIBUTION K(E,T)*Ki(E) vs. E AT THIS TEMP (K))
1750.,

Pout
760.,

OUTPUT OPTION (0,1,2,3)
1,
 NOTE:   CALCULATION/OUTPUT OPTION
    0  -  GAMMA FUNCTION : SHORT OUTPUT
    1  -  GAMMA FUNCTION : LONG OUTPUT
    2  -  FACTORIAL : SHORT OUTPUT
    3  -  FACTORIAL : LONG OUTPUT
```

TABLE IV.v

```
CYC6H5 DISSOCIATION
****DISSOCIATING SPECIES****
CYC6H5
****PRODUCT CHANNEL 1 {LOWEST ENERGY}****
 LINC6H5
****PRODUCT CHANNEL 2 ****

****PRODUCT CHANNEL 3 ****

***GEOMETRIC MEAN FREQUENCY;    # OF OSCILLATORS (S)*****
         1104.25,                          27
***COLLISION COMPLEX MASS;     DIAMETER;     WELL DEPTH*****
          77.0,            5.349,        412.30
***THIRD BODY (COLLISION PARTNER)*****
AR
***THIRD BODY MASS;  DIAMETER;  WELL DEPTH;    ENERGY TRANSFERRED******
         40.0,          3.54,     93.3,               630.
****COLLISION COEFFICIENT****
       0.028
****CHANNEL 1:  HIGH PRESSURE A FACTOR;  EA(KCAL/MOL)*****
                1.3E+15,                81.3,
****CHANNEL 2:  HIGH PRESSURE A FACTOR;  EA(KCAL/MOL)*****
                 0.0,                   0.0
****CHANNEL 3:  HIGH PRESSURE A FACTOR;  EA(KCAL/MOL)*****
                   0.0,                 0.0

NUMBER OF TEMPERATURES   T1      T2      T3      T4      T5      T6 (K)
4,                       1600.,  1700.,  1800.,  1900.,    .,       55

NUMBER OF PRESSURES      P1      P2      P3      P4      P5      P6 (TORR)
6,                      10.,   31.62,  100.,  316.23,  1000.,  10000.,

Tout (FOR DISTRIBUTION K(E,T)*Ki(E) vs. E AT THIS TEMP (K))
1700.,

Pout
760.,

OUTPUT OPTION (0,1,2,3)
1,
NOTE:   CALCULATION/OUTPUT OPTION
    0  -  GAMMA FUNCTION : SHORT OUTPUT
    1  -  GAMMA FUNCTION : LONG OUTPUT
    2  -  FACTORIAL : SHORT OUTPUT
    3  -  FACTORIAL : LONG OUTPUT
```

TABLE IV.vi

```
CYC6H6 DISSOCIATION
****DISSOCIATING SPECIES****
CYC6H6
****PRODUCT CHANNEL 1 (LOWEST ENERGY)****
 CYC6H5 + H
****PRODUCT CHANNEL 2 ****

****PRODUCT CHANNEL 3 ****

***GEOMETRIC MEAN FREQUENCY;    # OF OSCILLATORS (S)*****
         1205.8,                         30
***COLLISION COMPLEX MASS;     DIAMETER;     WELL DEPTH*****
           78.0,              5.349,        412.30
***THIRD BODY (COLLISION PARTNER)*****
AR
***THIRD BODY MASS;  DIAMETER;  WELL DEPTH;    ENERGY TRANSFERRED******
         40.0,          3.54,    93.3,              630.0
****COLLISION COEFFICIENT****
       0.0
****CHANNEL 1:  HIGH PRESSURE A FACTOR;  EA(KCAL/MOL)*****
                 1.3E+16,               113.6,
****CHANNEL 2:  HIGH PRESSURE A FACTOR;  EA(KCAL/MOL)*****
                  0.0,                  0.0
****CHANNEL 3:  HIGH PRESSURE A FACTOR;  EA(KCAL/MOL)*****
                    0.0,                0.0

NUMBER OF TEMPERATURES   T1      T2      T3      T4      T5      T6 (K)
4,                      1600.,  1700.,  1800.,  1900.,    .,

NUMBER OF PRESSURES     P1      P2      P3      P4      P5      P6 (TORR)
6,                     10.,    31.62,  100.,  316.23,  1000., 10000.,

Tout (FOR DISTRIBUTION K(E,T)*Ki(E) vs. E AT THIS TEMP (K))
1700.,

Pout
760.,

OUTPUT OPTION (0,1,2,3)
0,
NOTE:   CALCULATION/OUTPUT OPTION
    0  -  GAMMA FUNCTION : SHORT OUTPUT
    1  -  GAMMA FUNCTION : LONG OUTPUT
    2  -  FACTORIAL : SHORT OUTPUT
    3  -  FACTORIAL : LONG OUTPUT
```

APPENDIX IV - B

FIGURES FOR SECTION IV

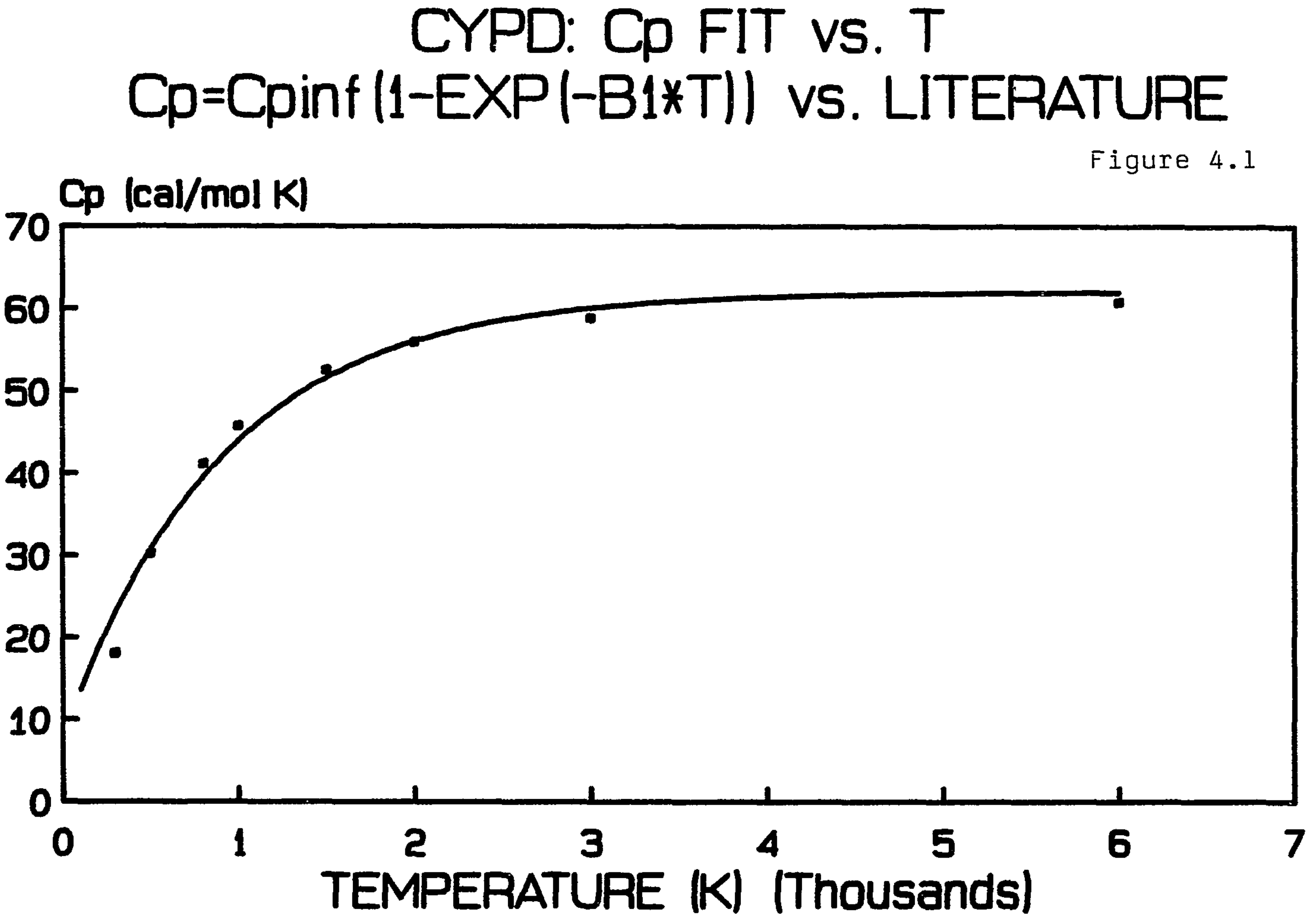
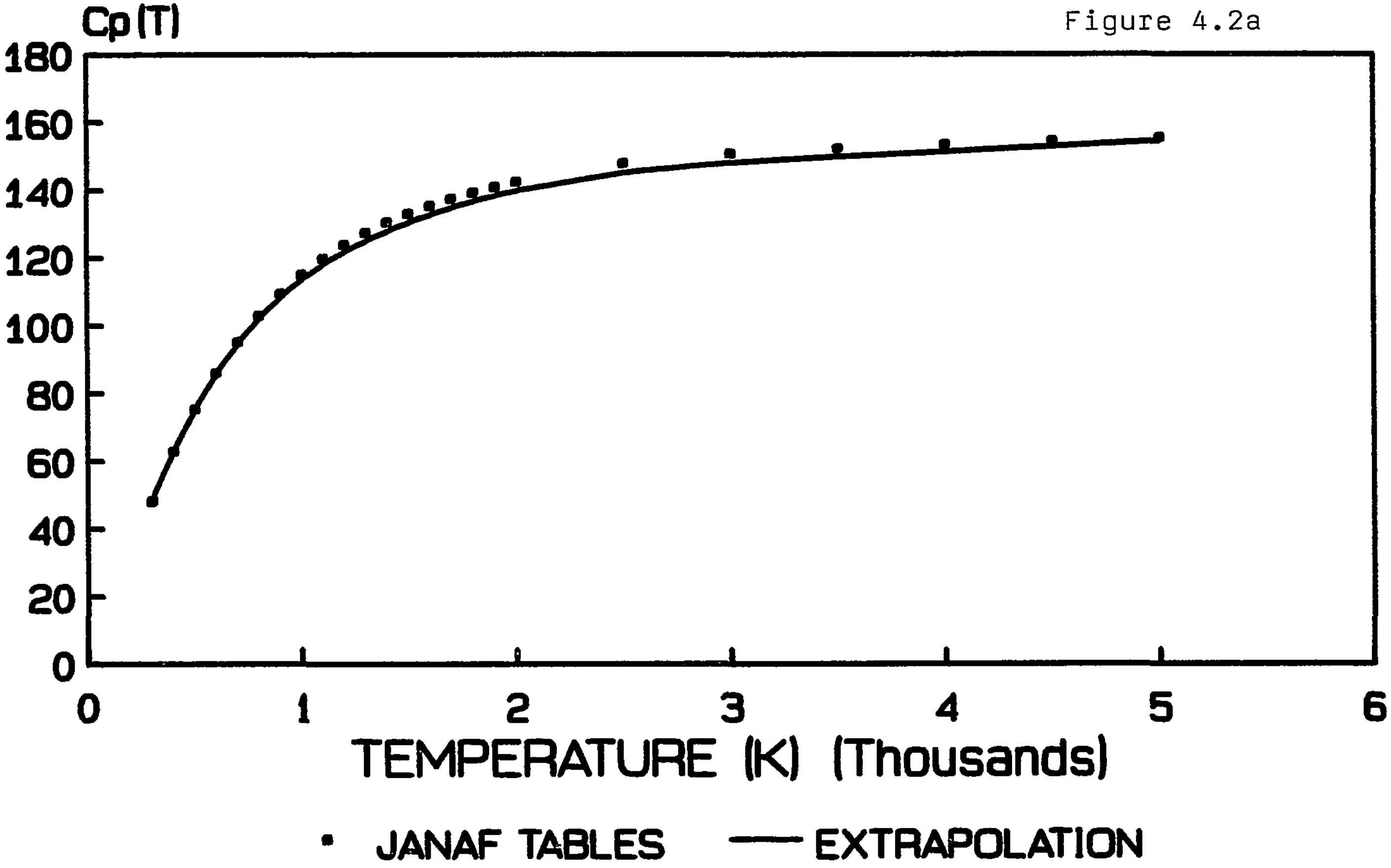

CYPD: Cp FIT vs. T
Cp=Cpinf(1-EXP(-B1*T)) vs. LITERATURE
Cp (cal/mol K)
70
60
50
40
30
20
10
0
0
1
2
3
4
5
6
7
TEMPERATURE (K) (Thousands)

Figure 4.1

4.-
a

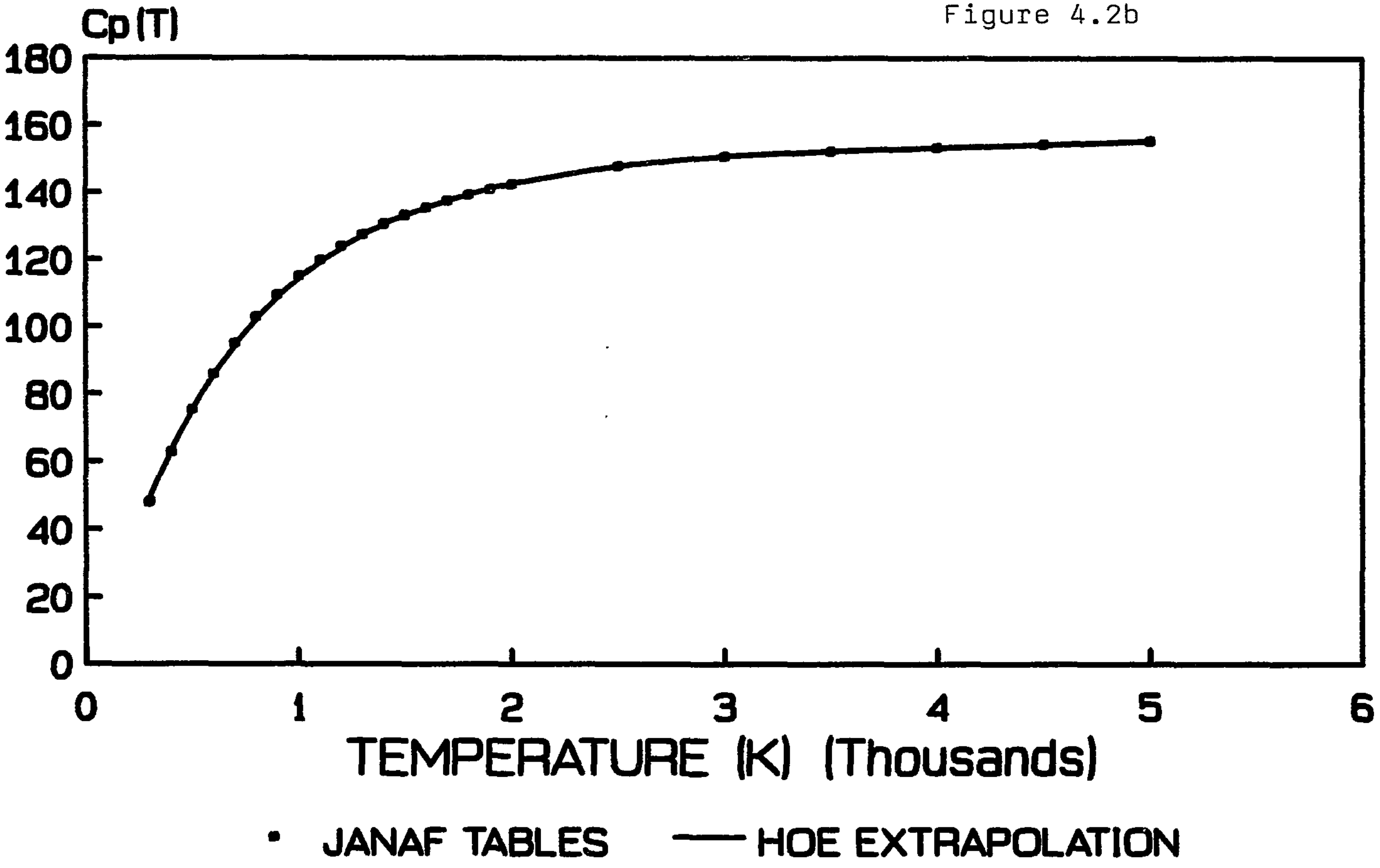
ETHYLENE OXIDE Cp(T)
FIT WITH EXPONENTIAL MODEL

Figure 4.2a



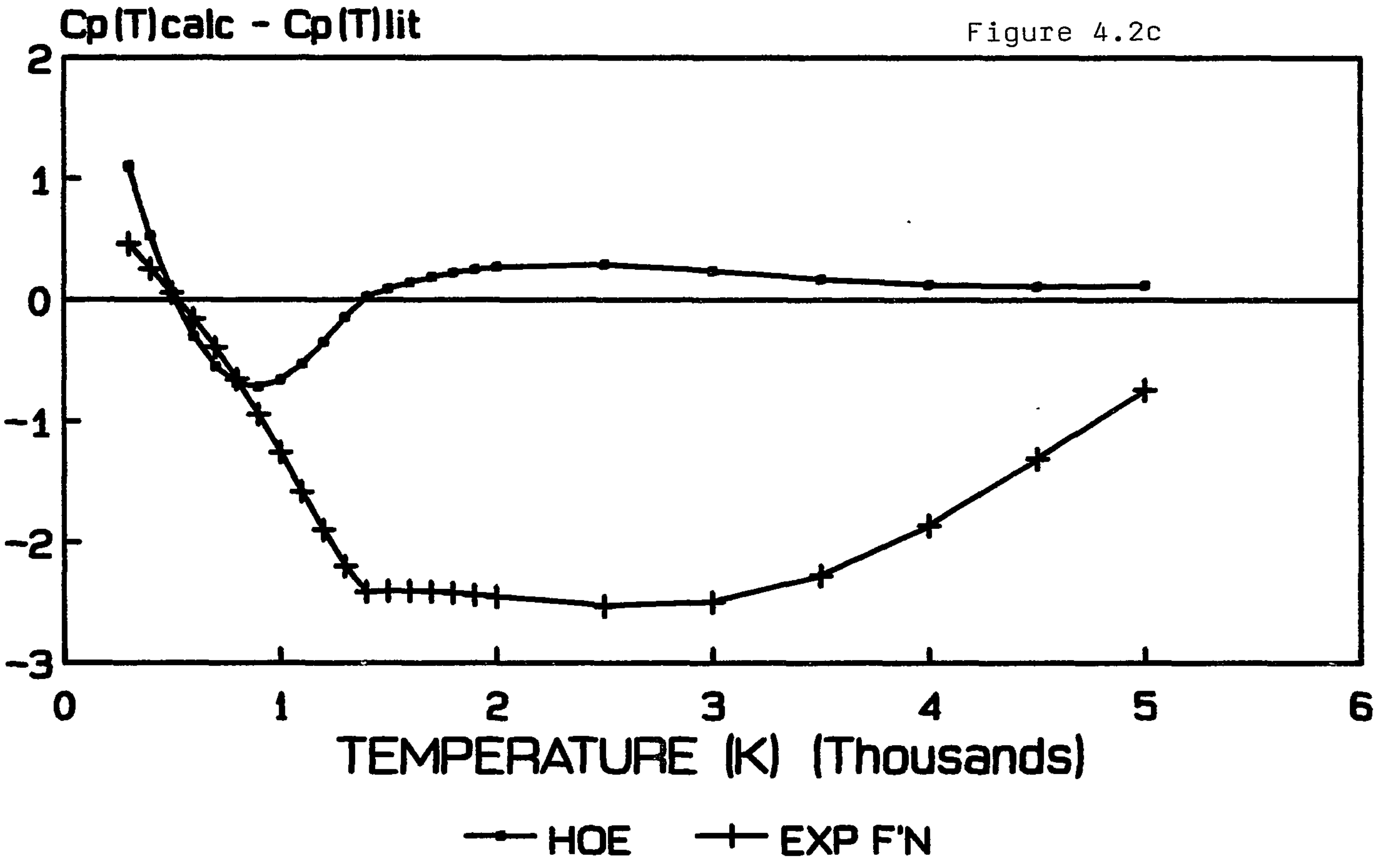
ETHYLENE OXIDE Cp (T)
FIT WITH HOE MODEL & THERM DATA

Figure 4.2b



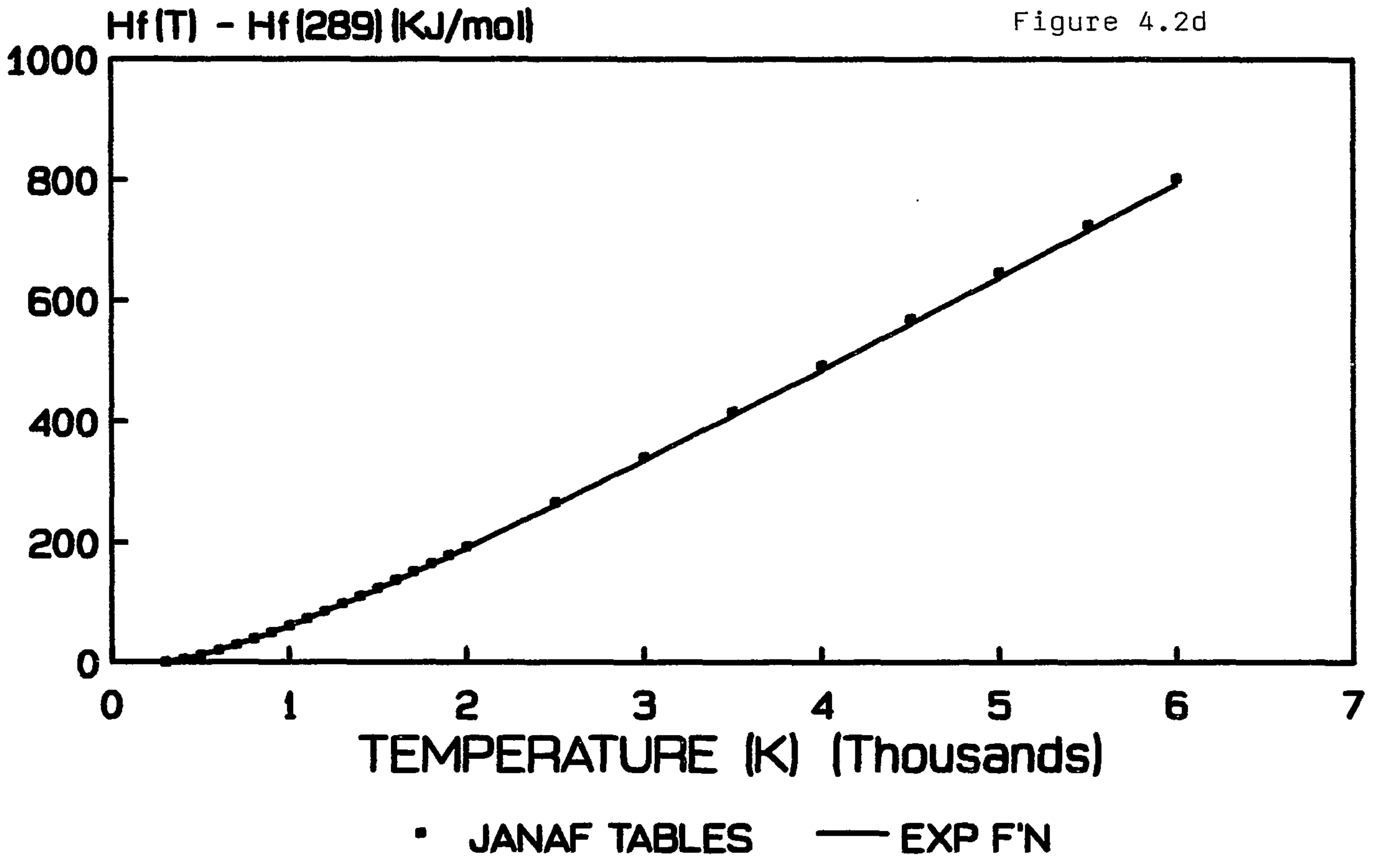
ETHYLENE OXIDE Cp(T):RESIDUALS
RESIDUALS FOR FIT OF HOE AND EXP MODELS

Figure 4.2c



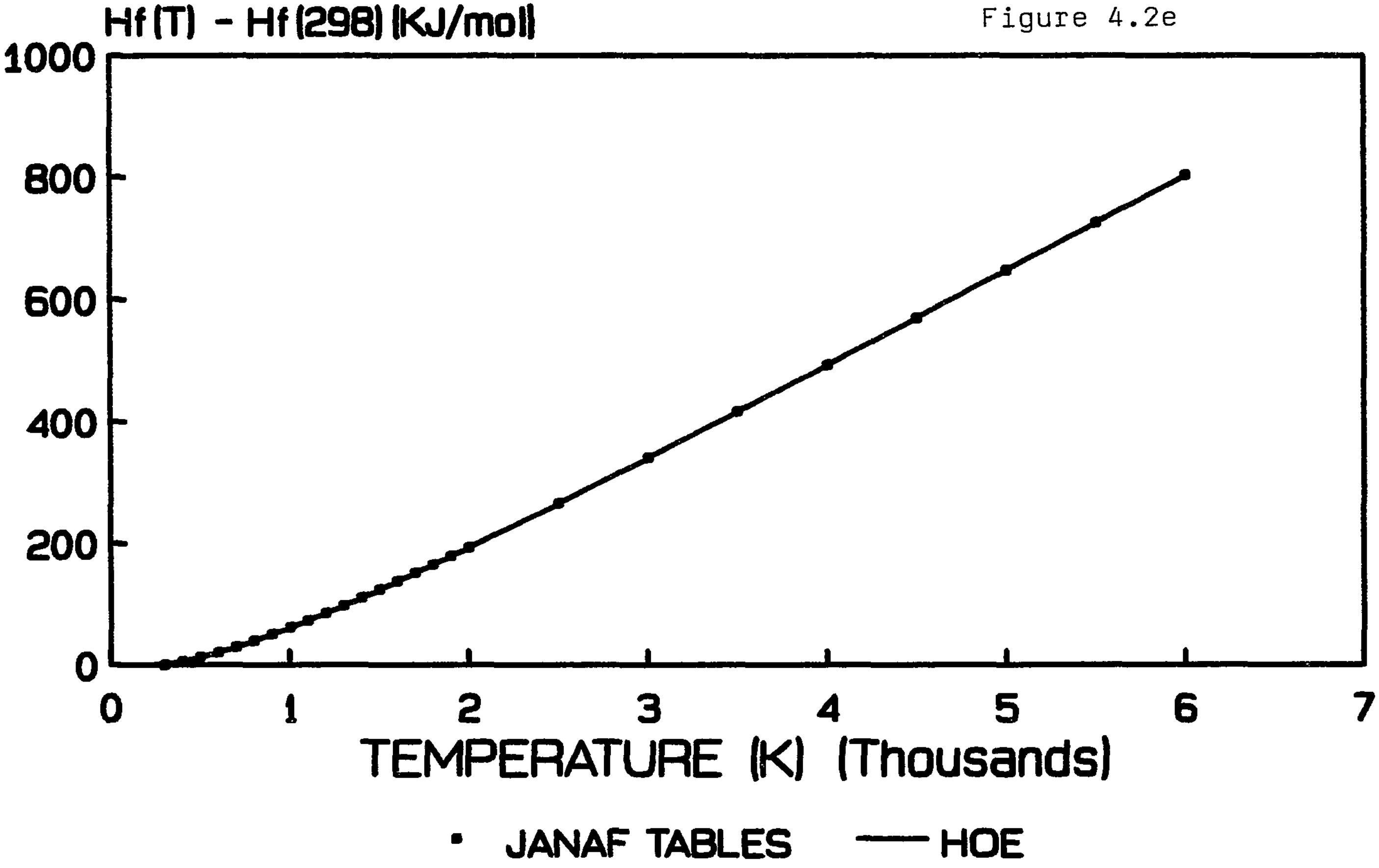
ETHYLENE OXIDE: Hf (T)-Hf (298K)
FIT FOR EXPONENTIAL EXTRAPOLATION: THER

Figure 4.2d



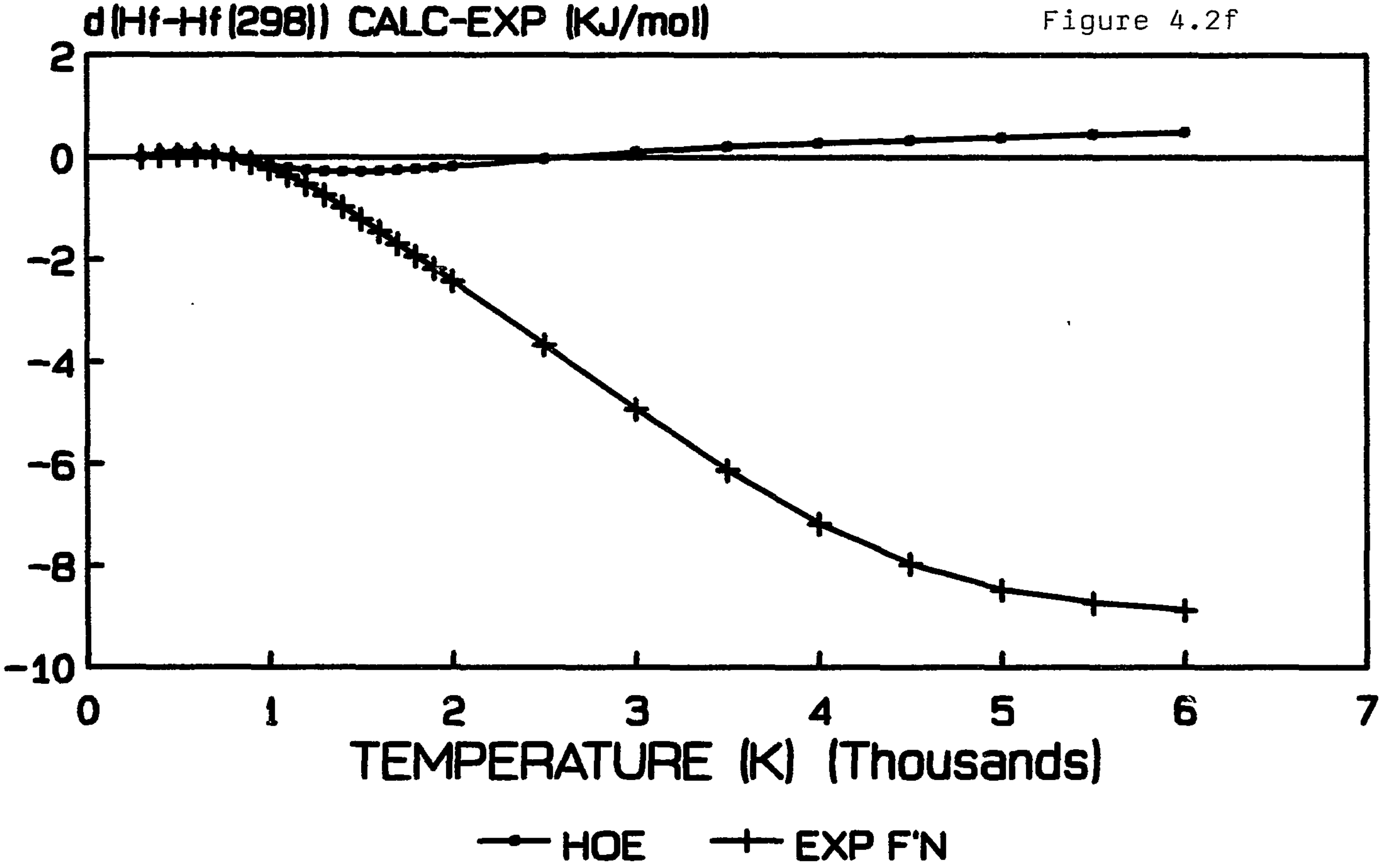
ETHYLENE OXIDE: Hf (T) -Hf (298K)
FIT FOR HOE EXTRAPOLATION: THERM

Figure 4.2e



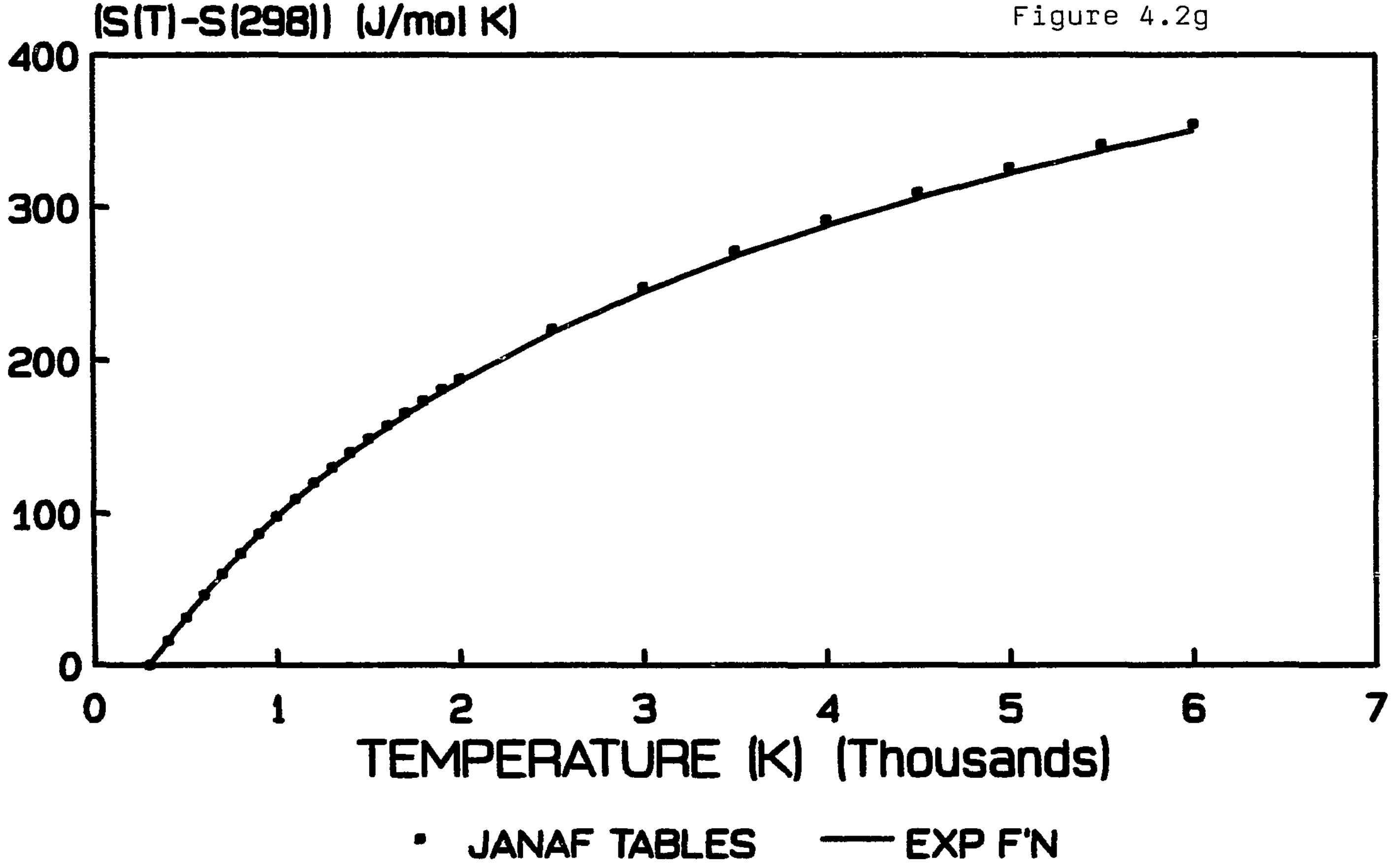
ETHYLENE OXIDE: Hf (T)-Hf (298K)
RESIDUALS FOR HOE AND EXP F'N MODELS

Figure 4.2f



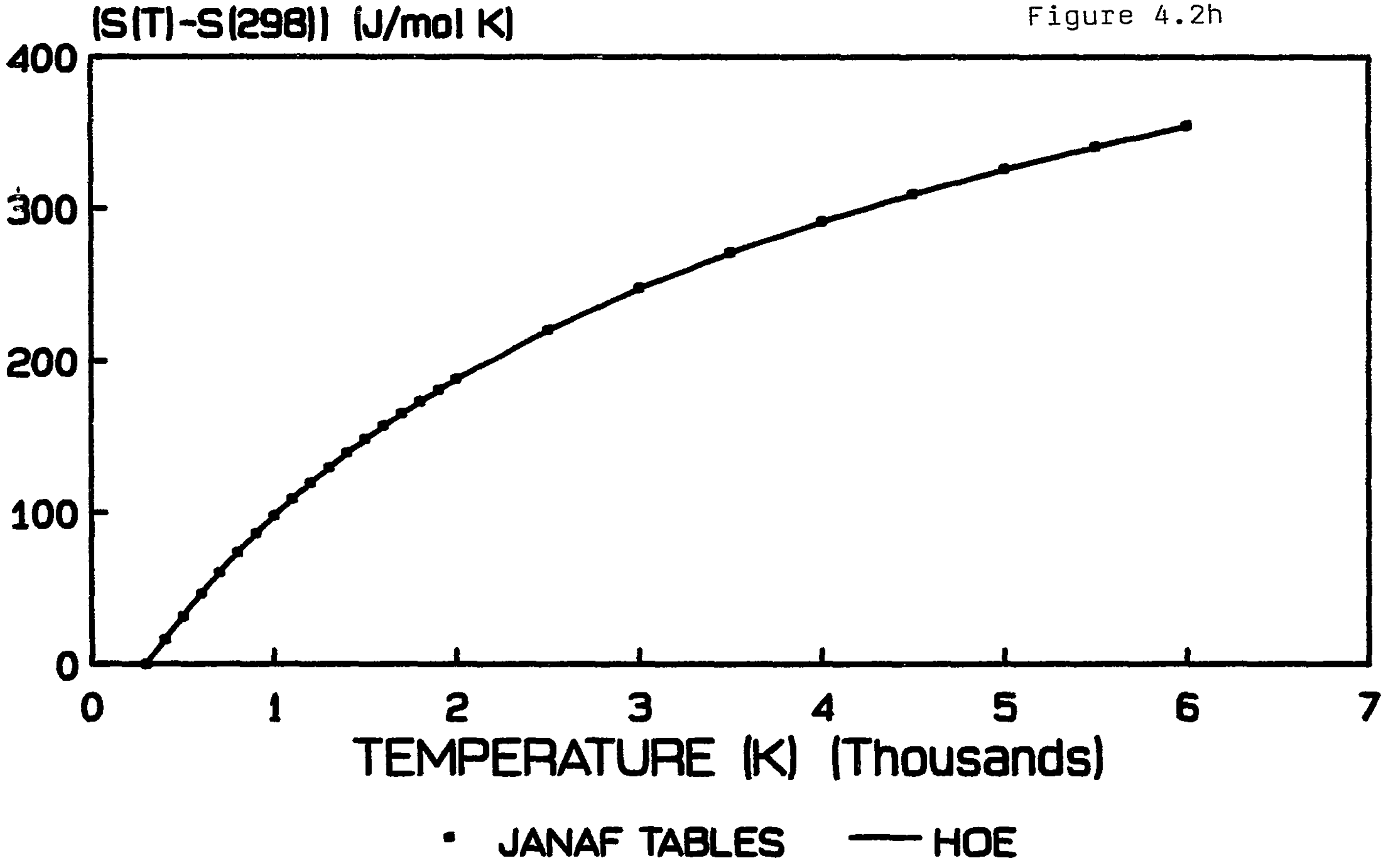
ETHYLENE OXIDE: S(T)-S(298K)
ENTROPY FUNCTION: EXP F'N MODEL

Figure 4.2g



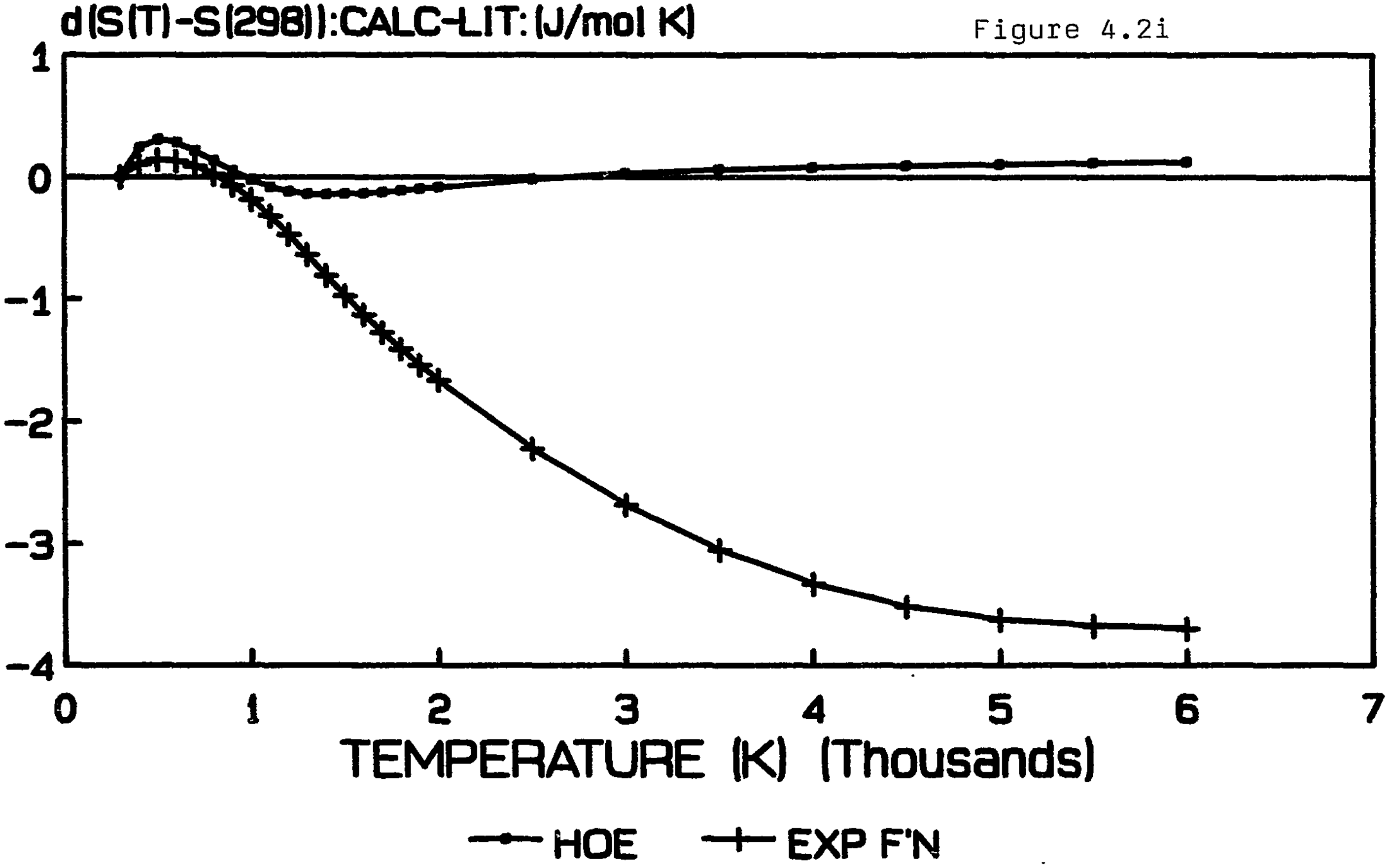
ETHYLENE OXIDE: S(T)-S(298K)
ENTROPY FUNCTION: HOE MODEL

Figure 4.2h



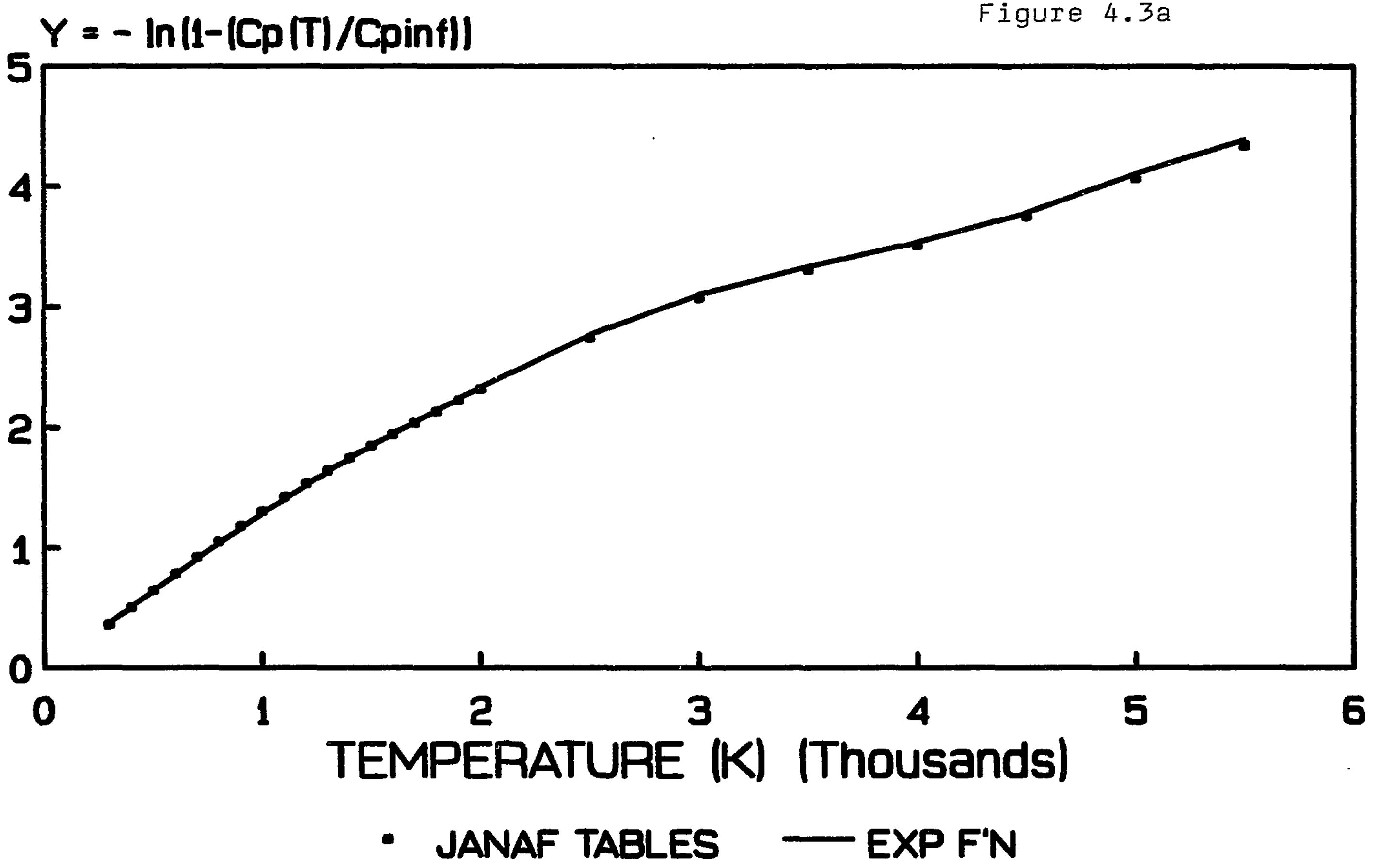
ETHYLENE OXIDE: S(T)-S(298K)
RESIDUALS ENTROPY FUNCTION

Figure 4.2i



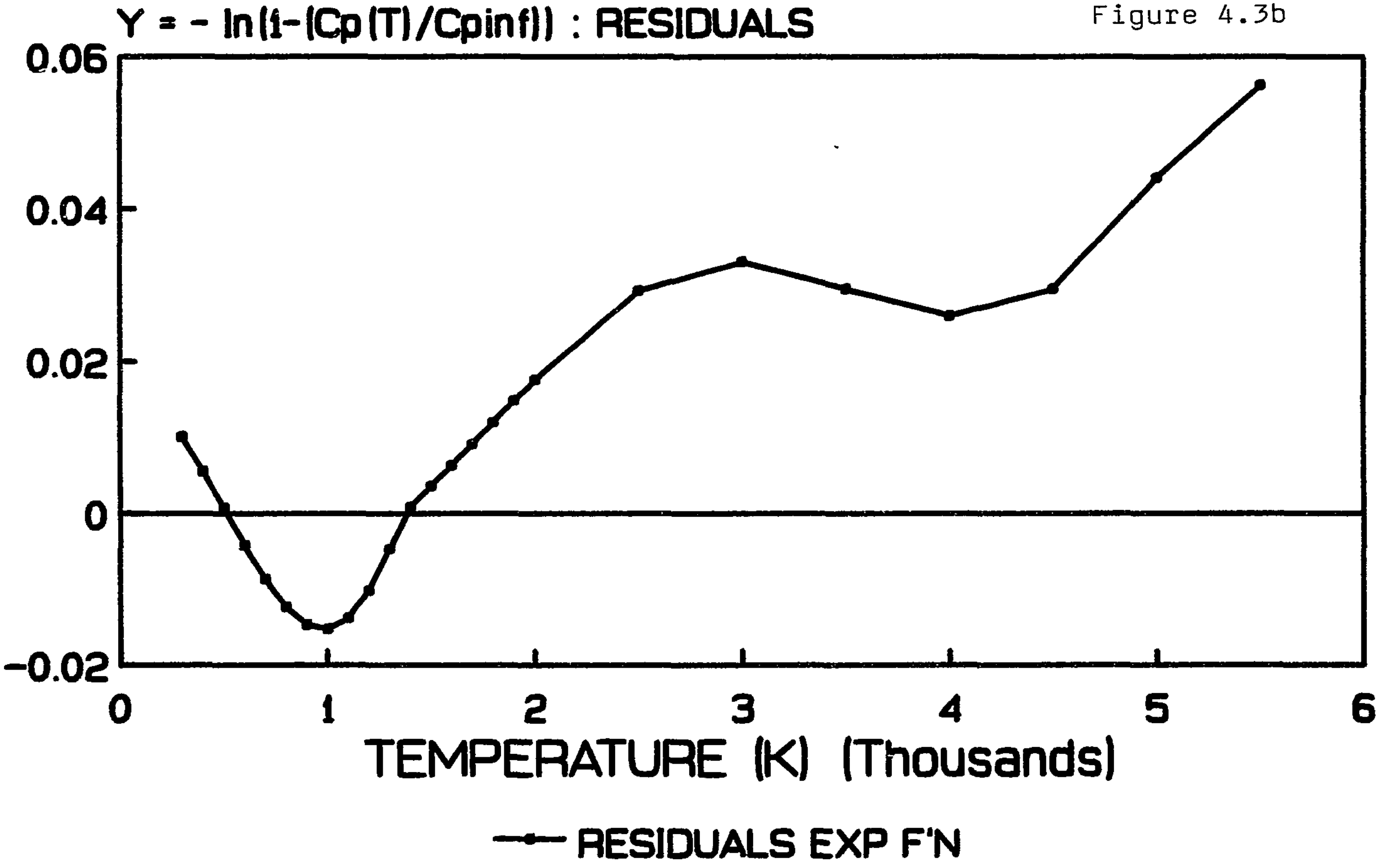
ETHYLENE OXIDE:Cp EXP F'N
Y = -ln(1-Cp/Cpinf) = A + BT

Figure 4.3a



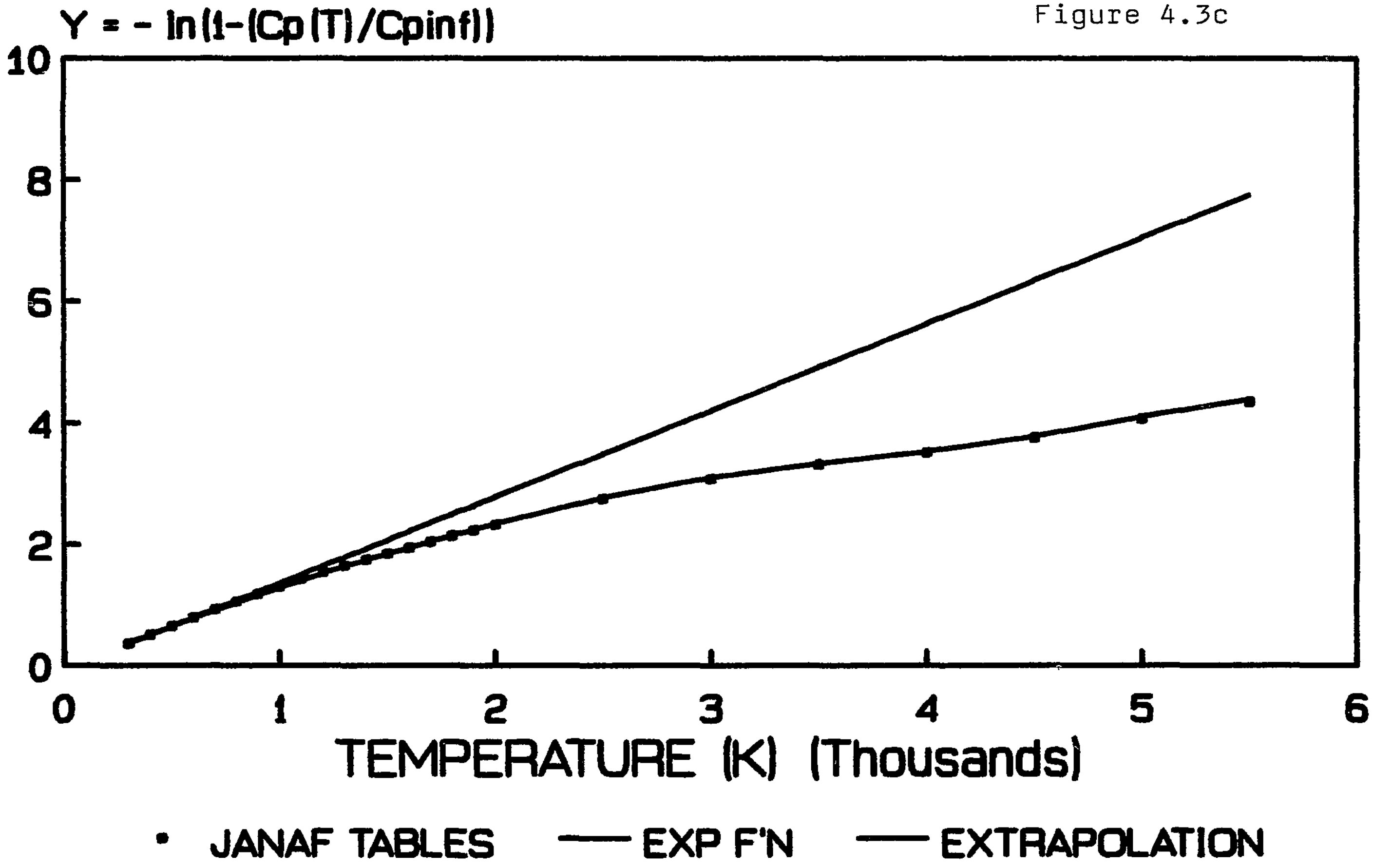
ETHYLENE OXIDE:Cp EXP F'N
Y= -ln(1-Cp/Cpinf) = A + BT : RESIDUALS

Figure 4.3b



ETHYLENE OXIDE:Cp EXP F'N
Y = -ln(1-Cp/Cpinf) = A + BT

Figure 4.3c



4.4

CYCLOHEPTATRIENE : Cp(T) polynomials low and high Temp polynomials

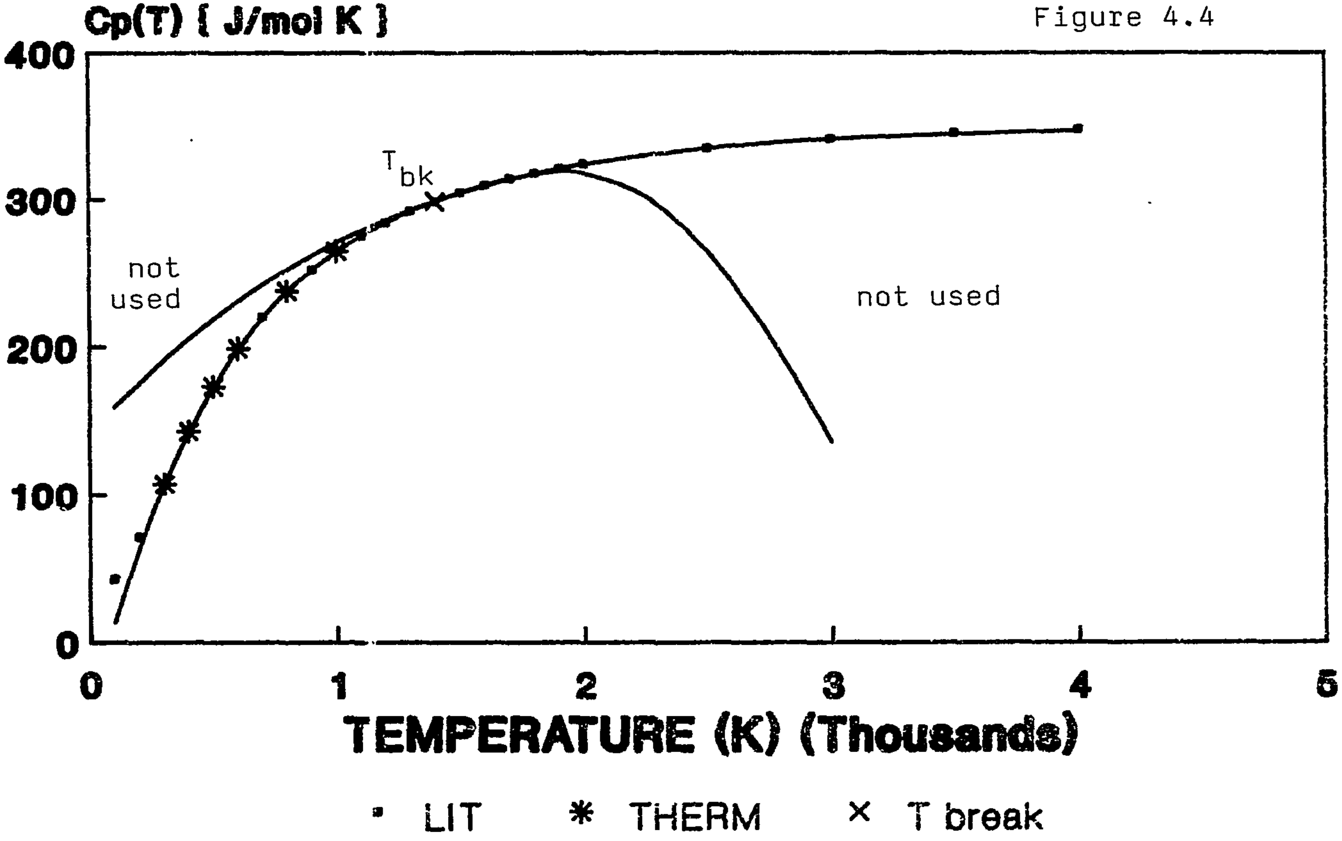
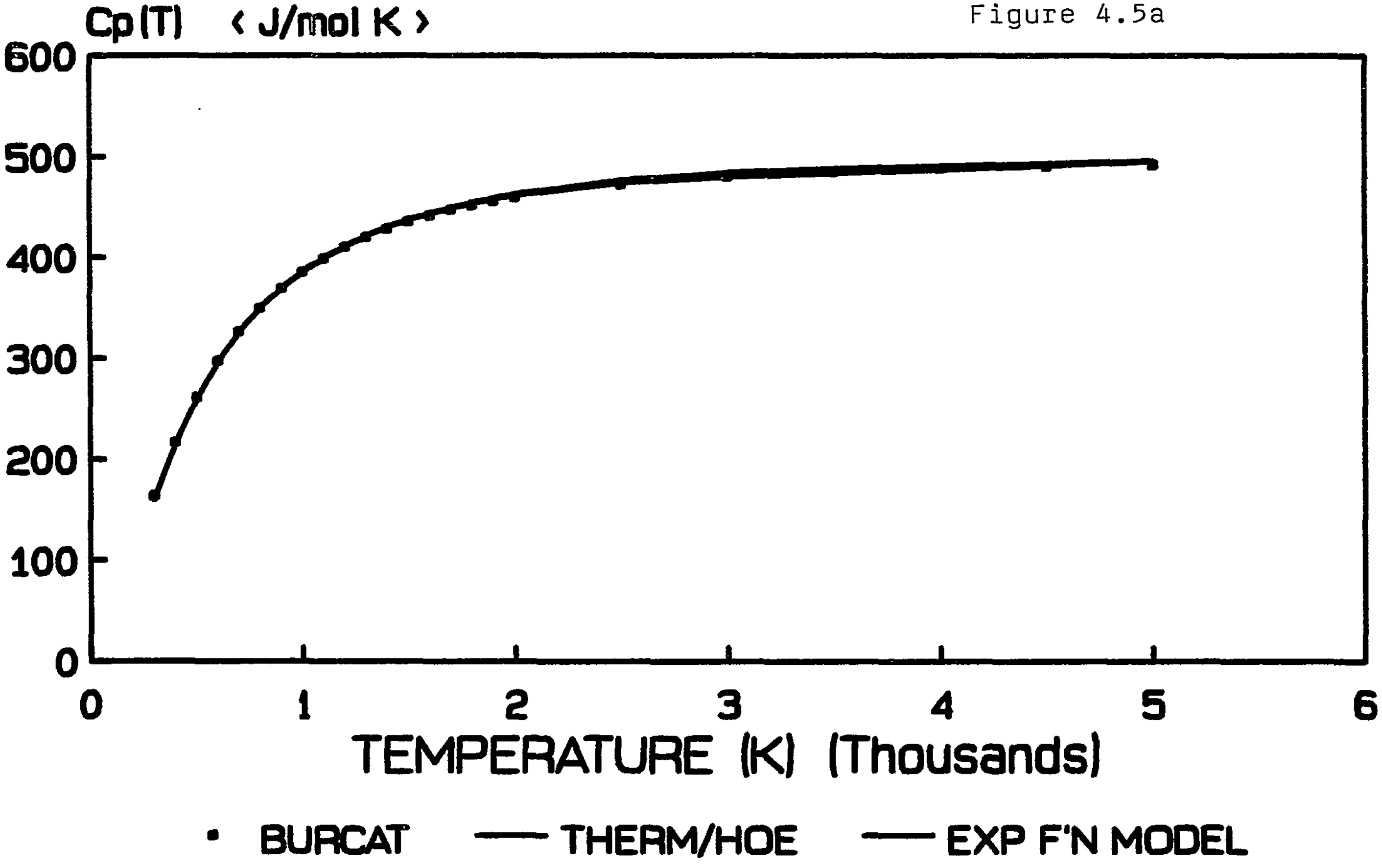


Figure 4.4

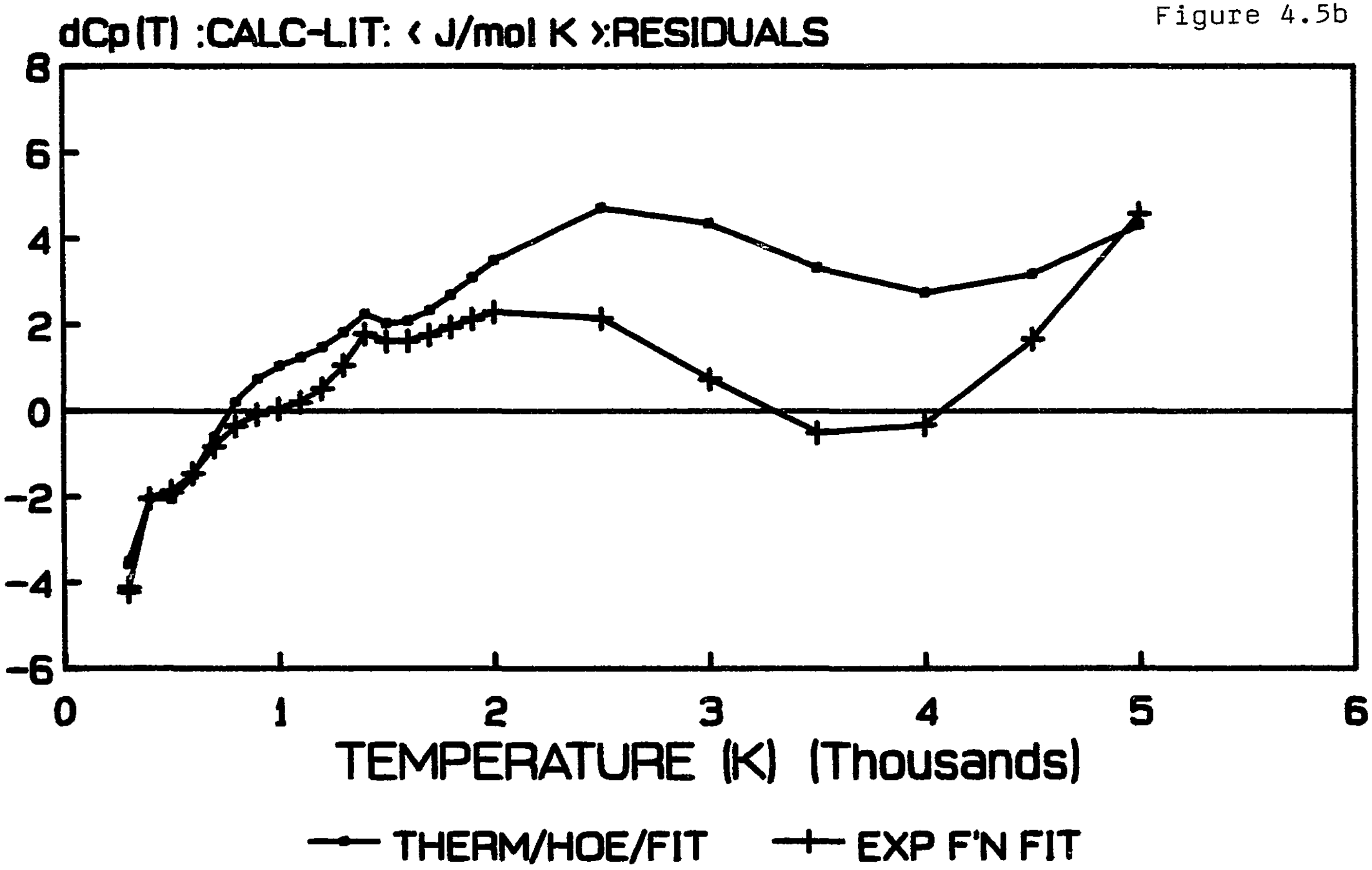
BIPHENYL RADICAL Cp(T): EXP MODEL THERM DATA vs. BURCAT

Figure 4.5a



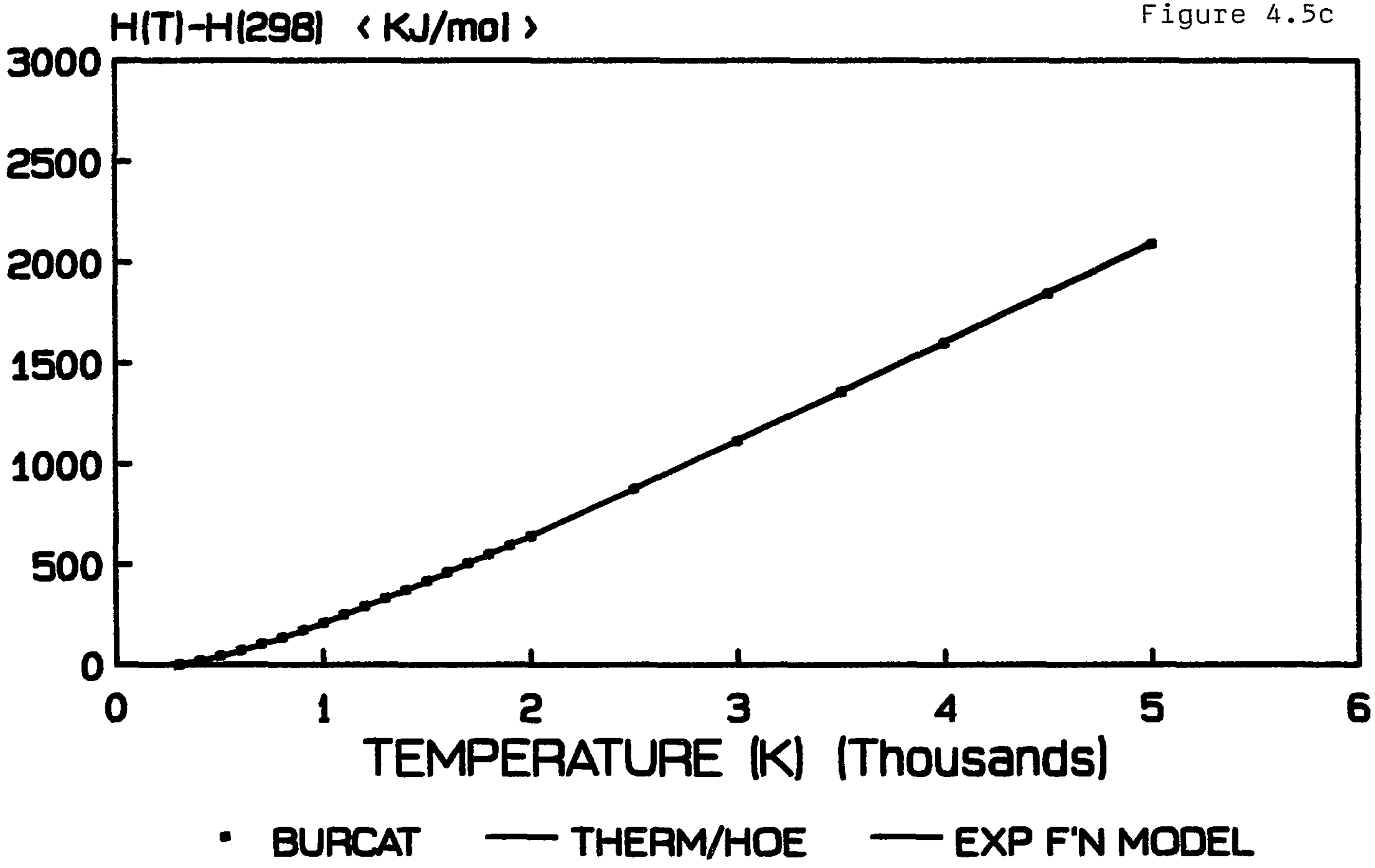
BIPHENYL RADICAL Cp (T): RESIDUALS
THERM DATA vs. BURCAT

Figure 4.5b



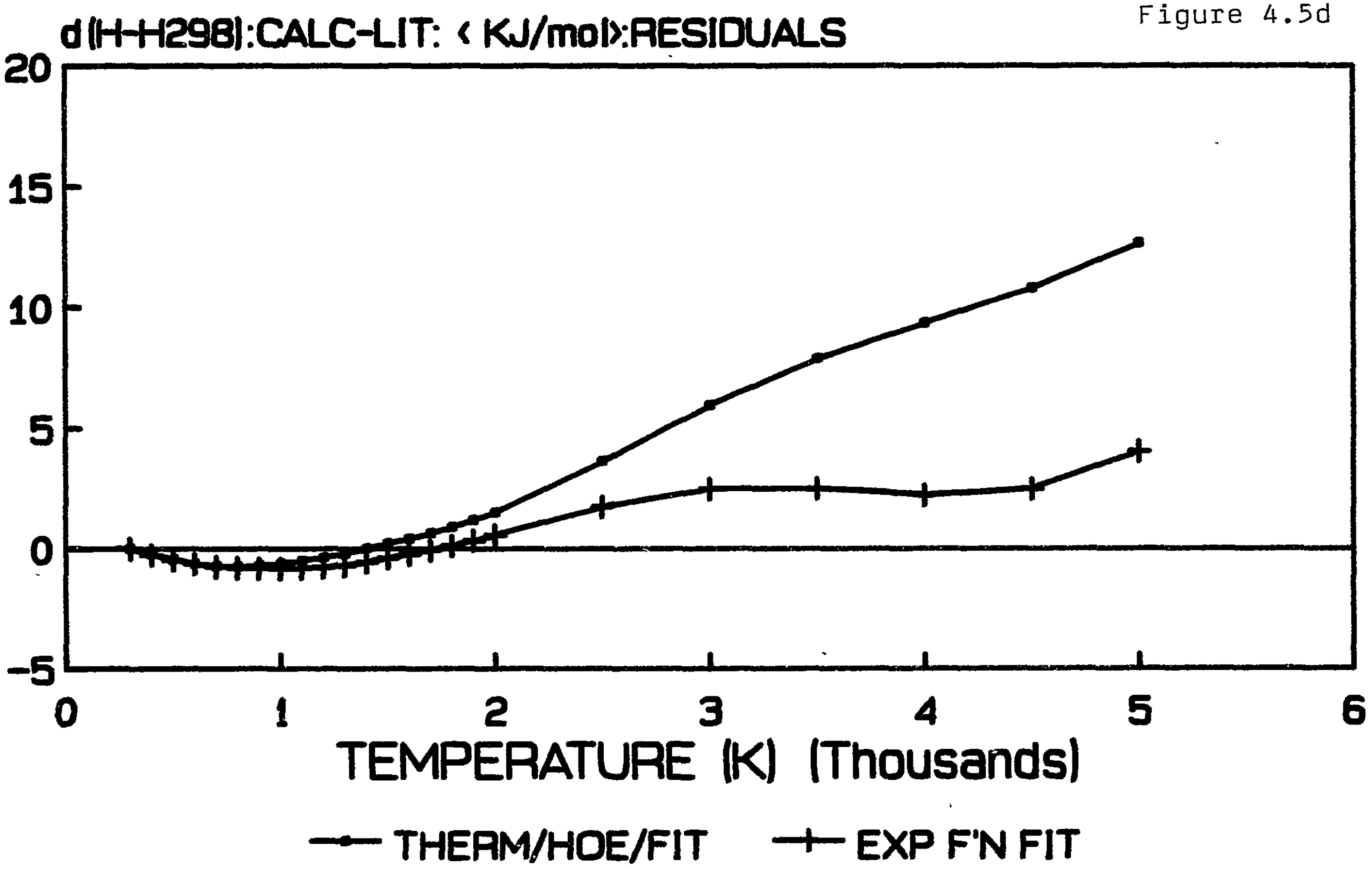
BIPHENYL RADICAL: H-H298: EXP MODEL
THERM DATA vs. BURCAT

Figure 4.5c



BIPHENYL RADICAL: H-H298: RESIDUALS
THERM DATA vs. BURCAT

Figure 4.5d



BIPHENYL RADICAL: S-S298: EXP MODEL
THERM DATA vs. BURCAT

Figure 4.5e

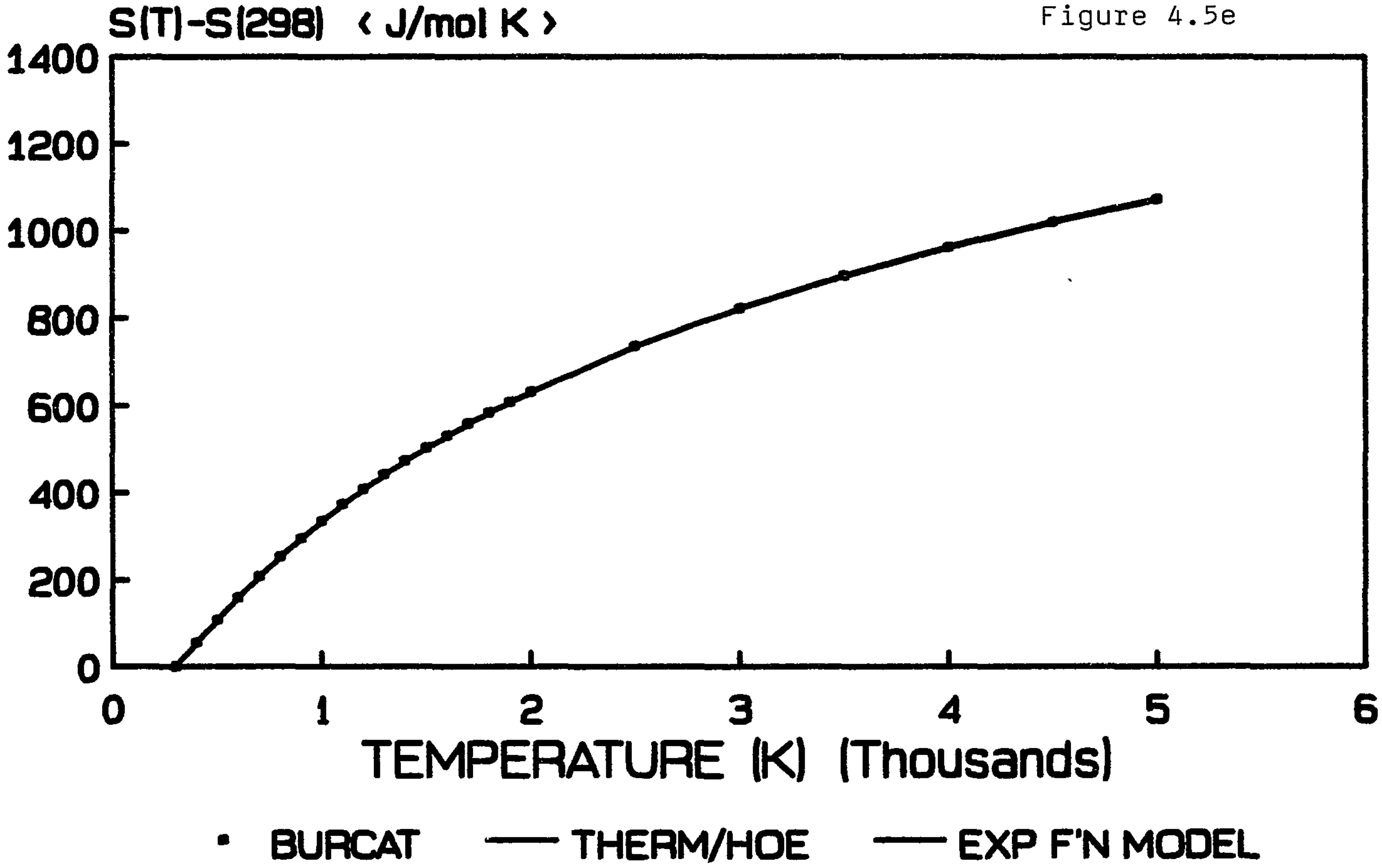
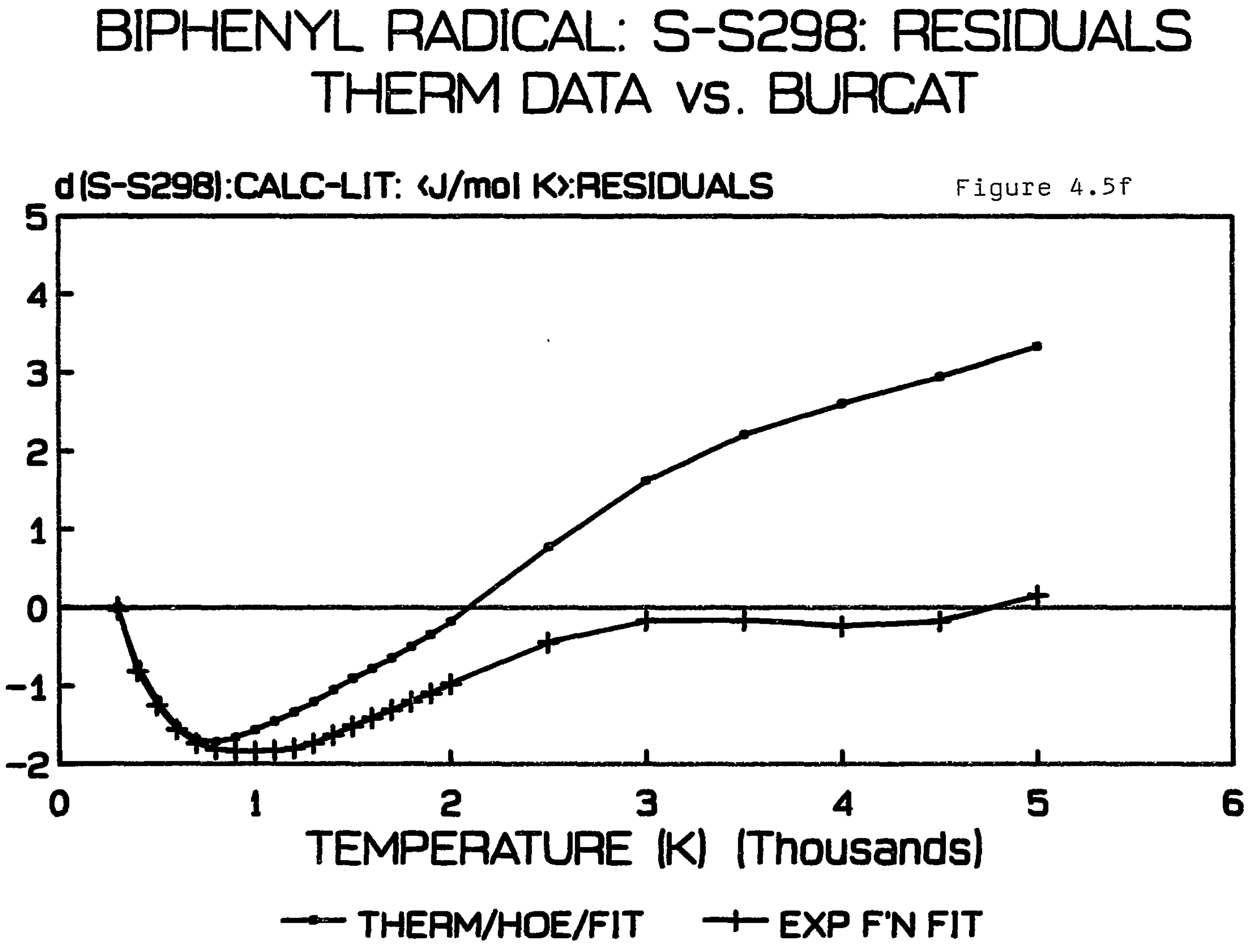
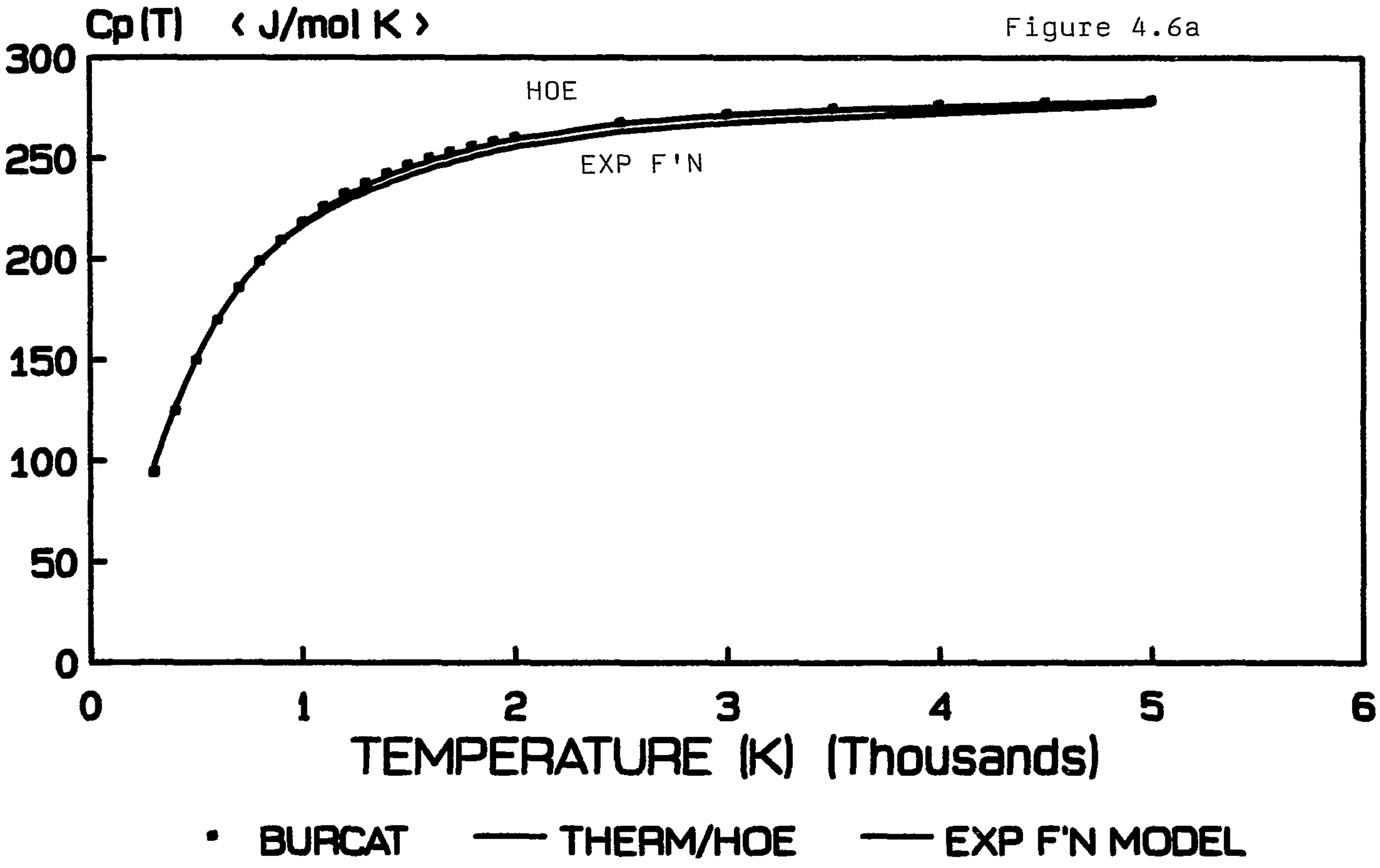


Figure 4.5f


BIPHENYL RADICAL: S-S298: RESIDUALS
THERM DATA vs. BURCAT
d(S-S298):CALC-LIT: <J/mol K>:RESIDUALS
5
4
3
2
1
0
-1
-2
0
1
2
3
4
5
6
TEMPERATURE (K) (Thousands)
THERM/HOE/FIT
EXP F'N FIT

PHENOXY RADICAL Cp(T): EXP MODEL THERM DATA vs. BURCAT

Figure 4.6a



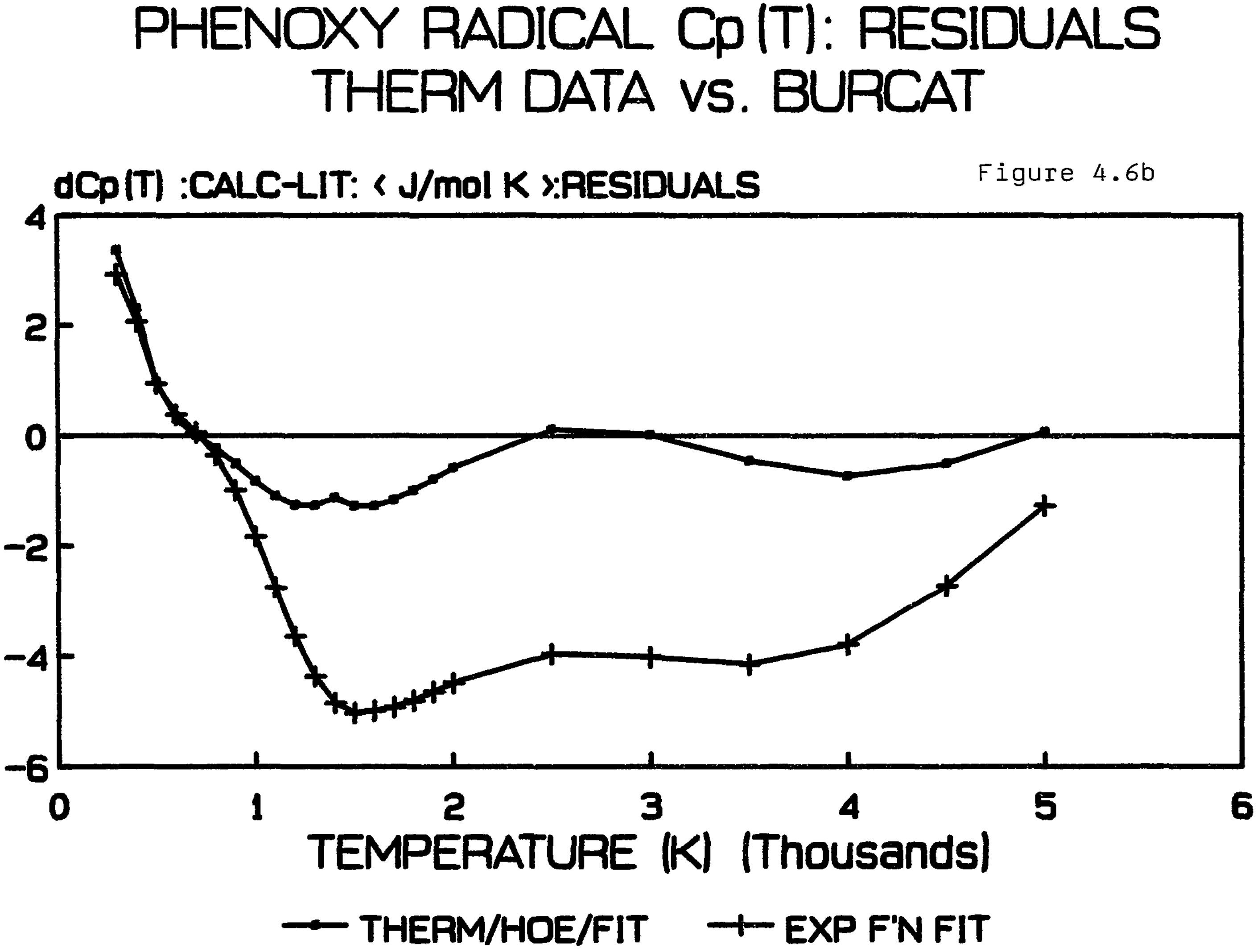
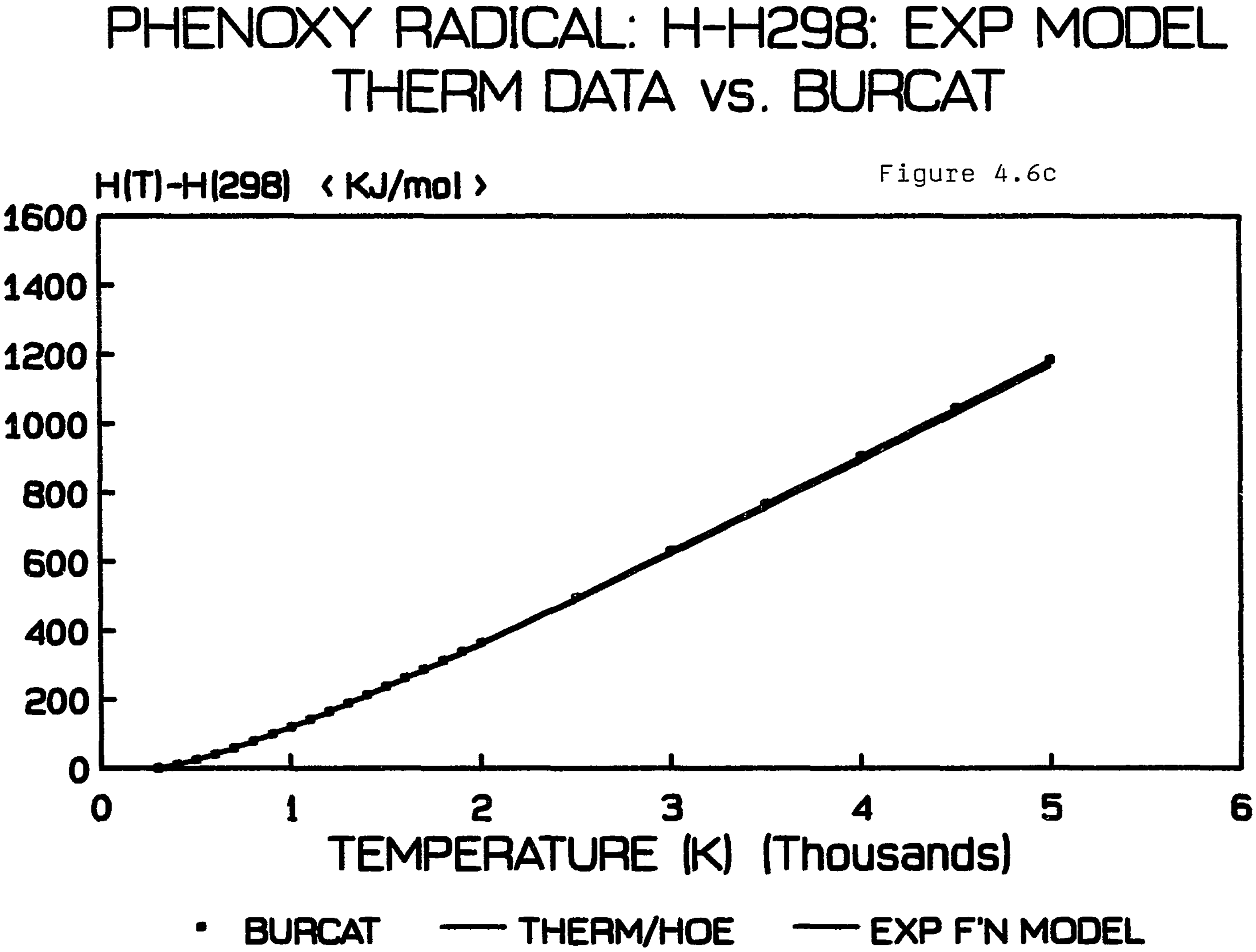

PHENOXY RADICAL Cp(T): RESIDUALS
THERM DATA vs. BURCAT
dCp(T) :CALC-LIT: < J/mol K >:RESIDUALS
4
2
0
-2
-4
-6
0
1
2
3
4
5
6
TEMPERATURE (K) (Thousands)
THERM/HOE/FIT
EXP F'N FIT

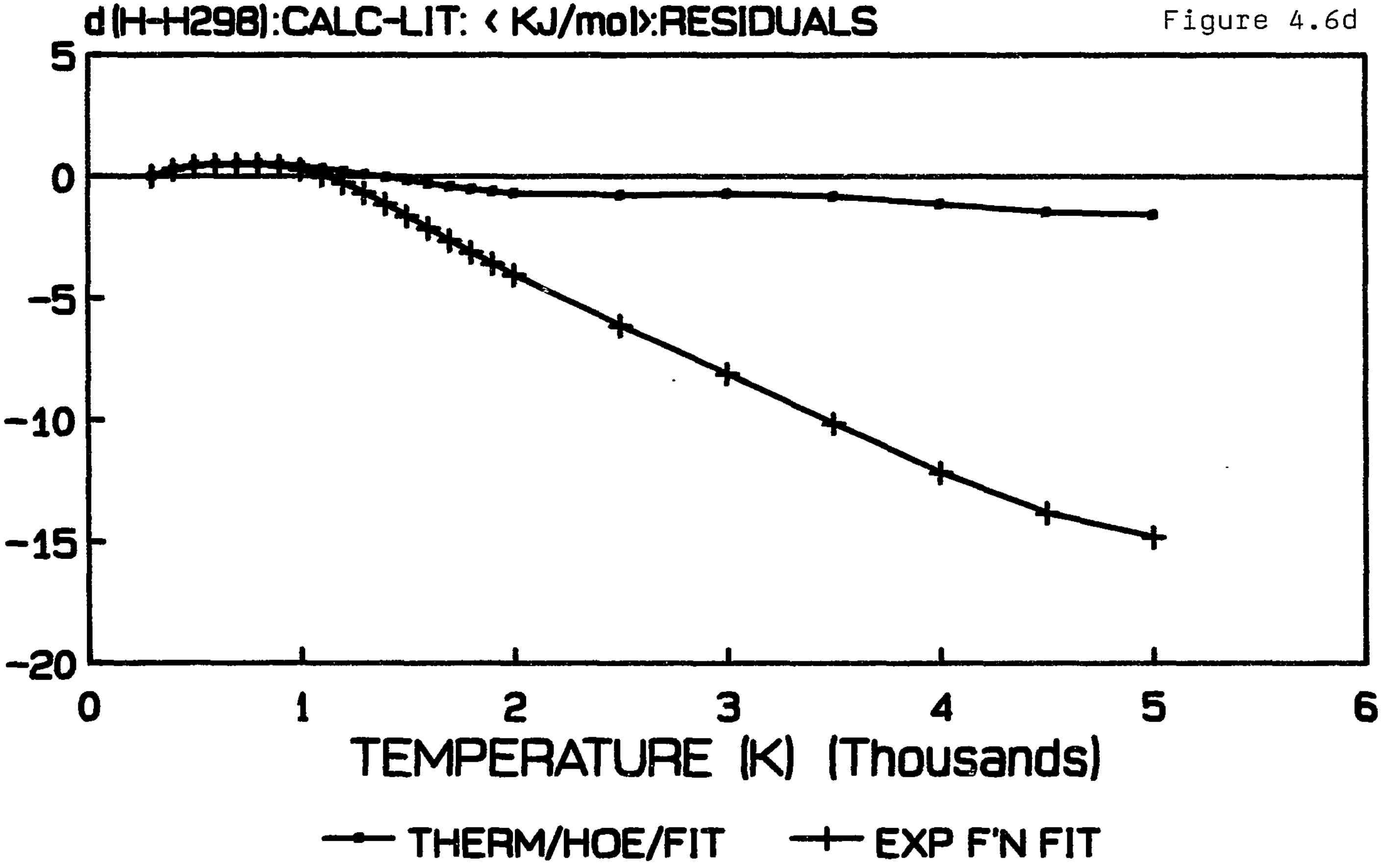
Figure 4.6b

Figure 4.6c


PHENOXY RADICAL: H-H298: EXP MODEL
THERM DATA vs. BURCAT
H(T)-H(298) ‹ KJ/mol ›
1600
1400
1200
1000
800
600
400
200
0
0
1
2
3
4
5
6
TEMPERATURE (K) (Thousands)
• BURCAT
THERM/HOE
EXP F'N MODEL

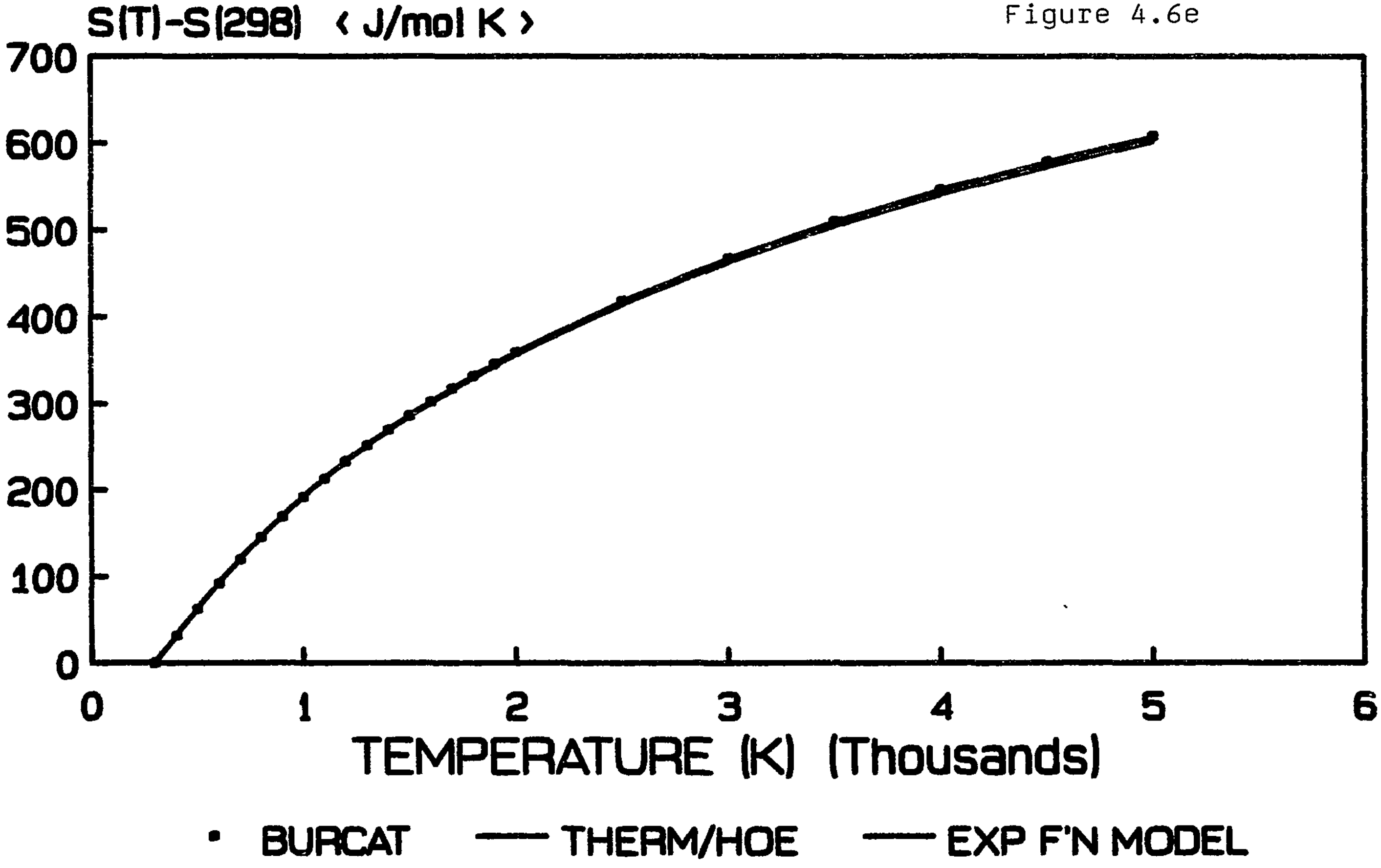
PHENOXY RADICAL: H-H298: RESIDUALS
THERM DATA vs. BURCAT

Figure 4.6d



PHENOXY RADICAL: S-S298: EXP MODEL THERM DATA vs. BURCAT

Figure 4.6e



PHENOXY RADICAL: S-S298: RESIDUALS
THERM DATA vs. BURCAT

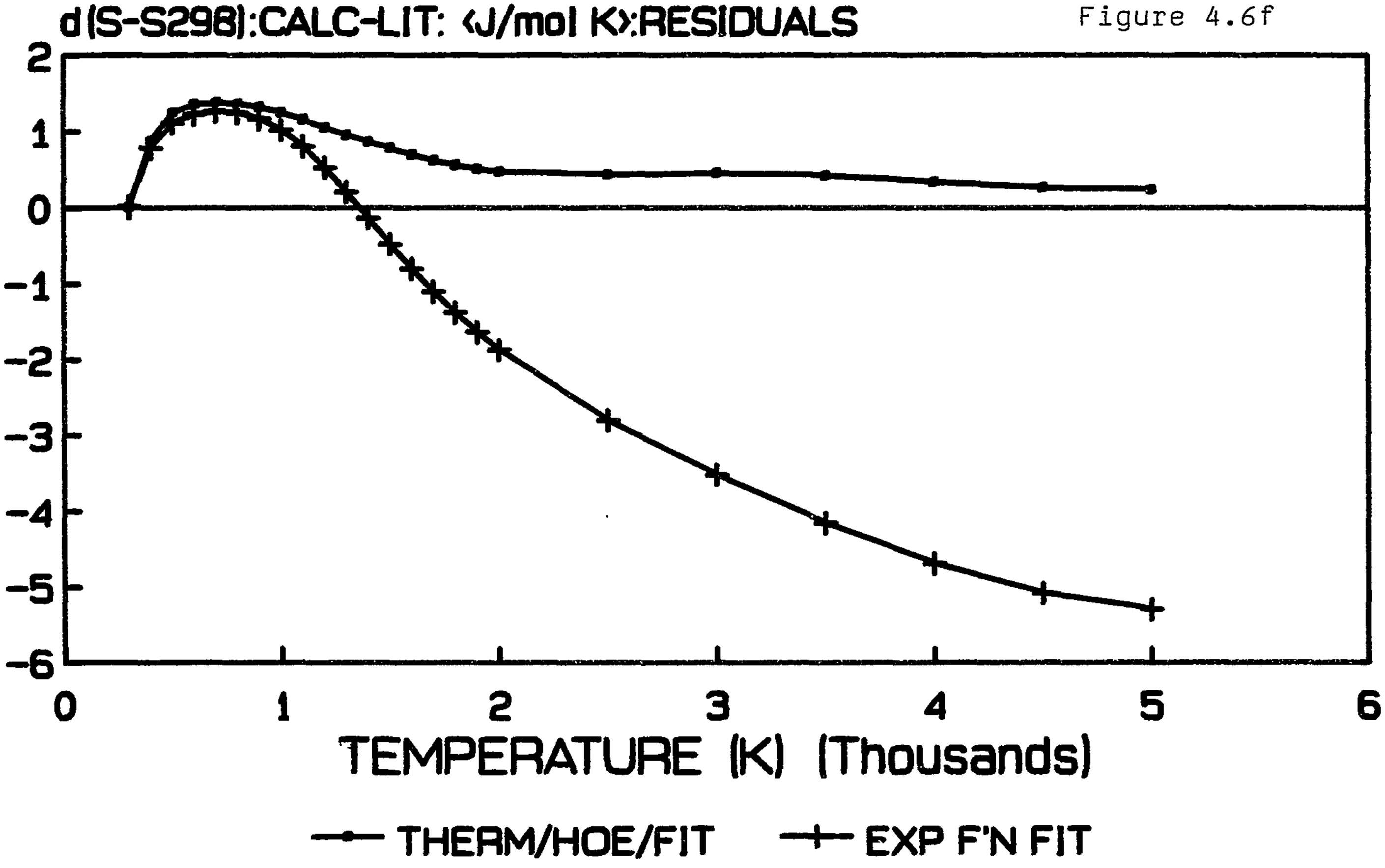
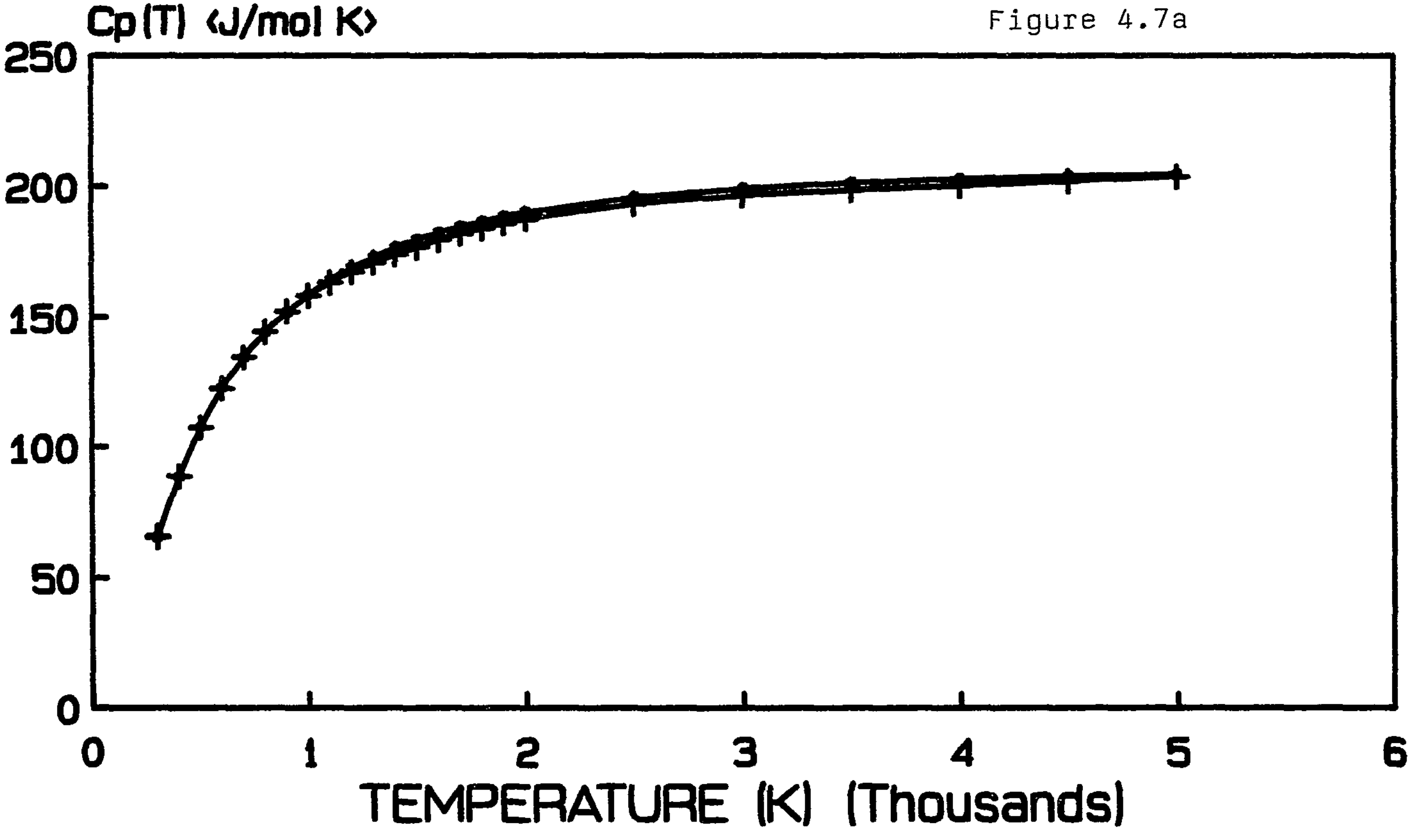


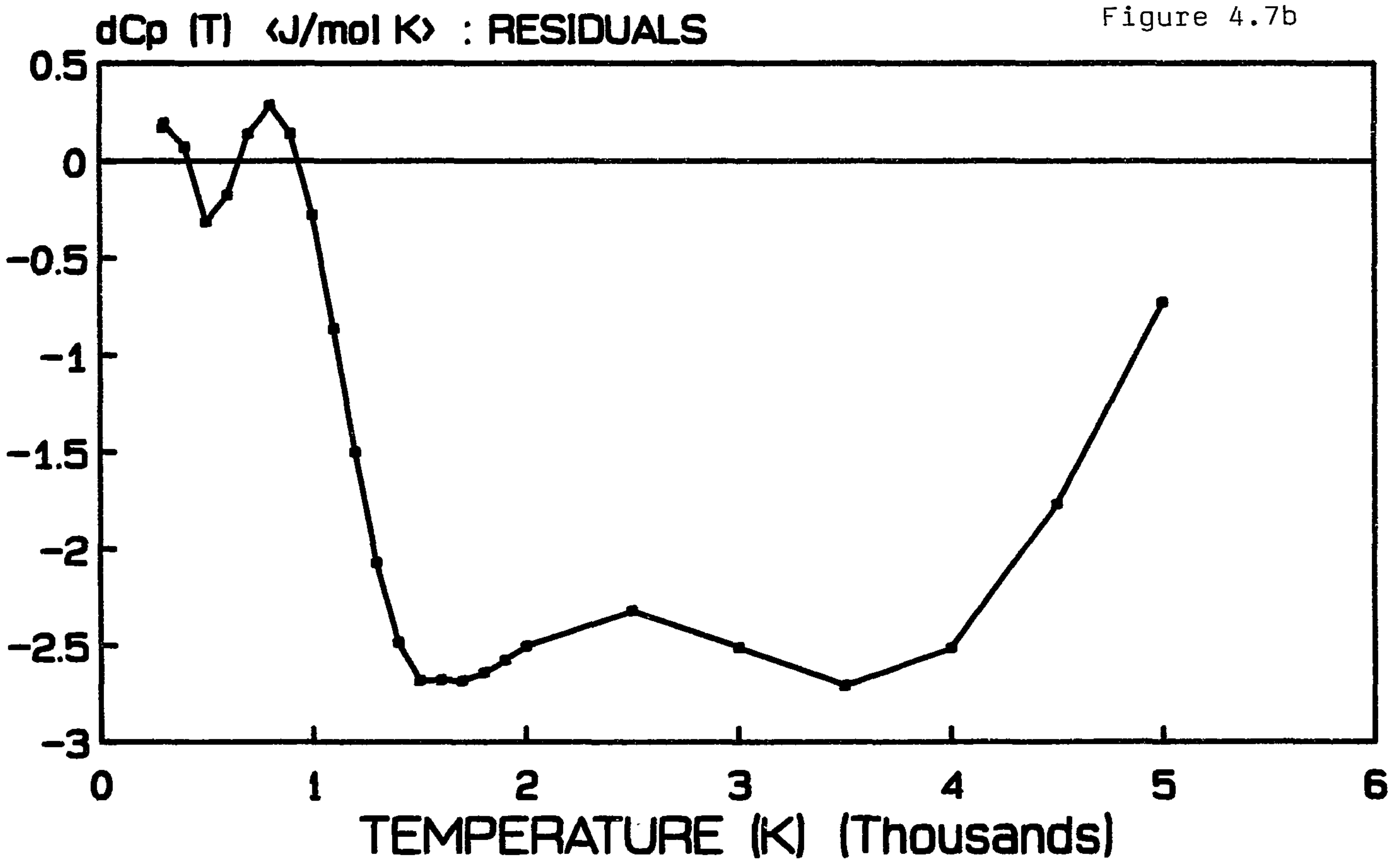
Figure 4.6f

FURAN : Cp (T)

Figure 4.7a

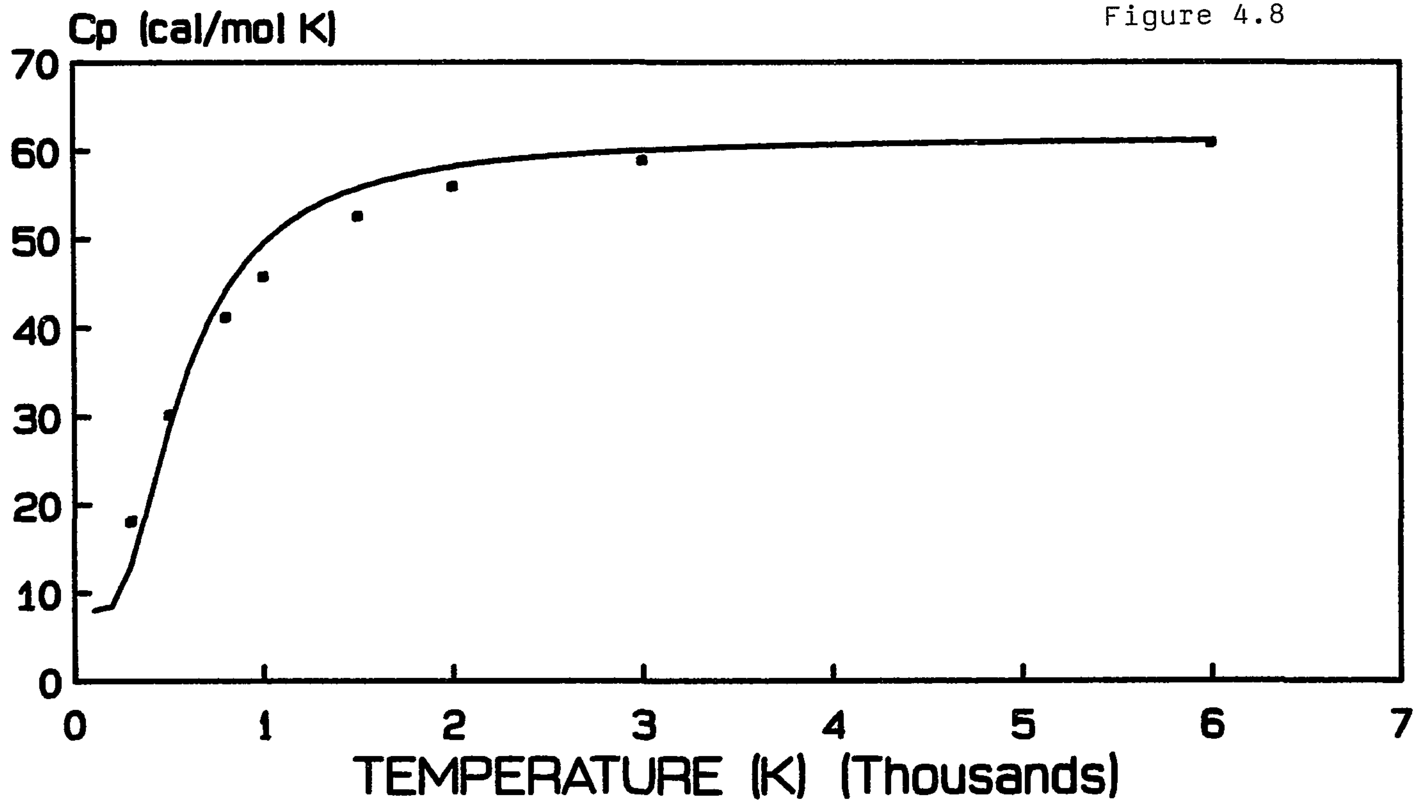


FURAN :Cp (T): RESIDUALS



CYPD: Cp FIT vs. T
GEOMETRIC MEAN APPROX vs. LITERATURE

Figure 4.8



CYPD: Cp FIT vs. T
ARITHMETIC MEAN APPROX vs. LITERATURE

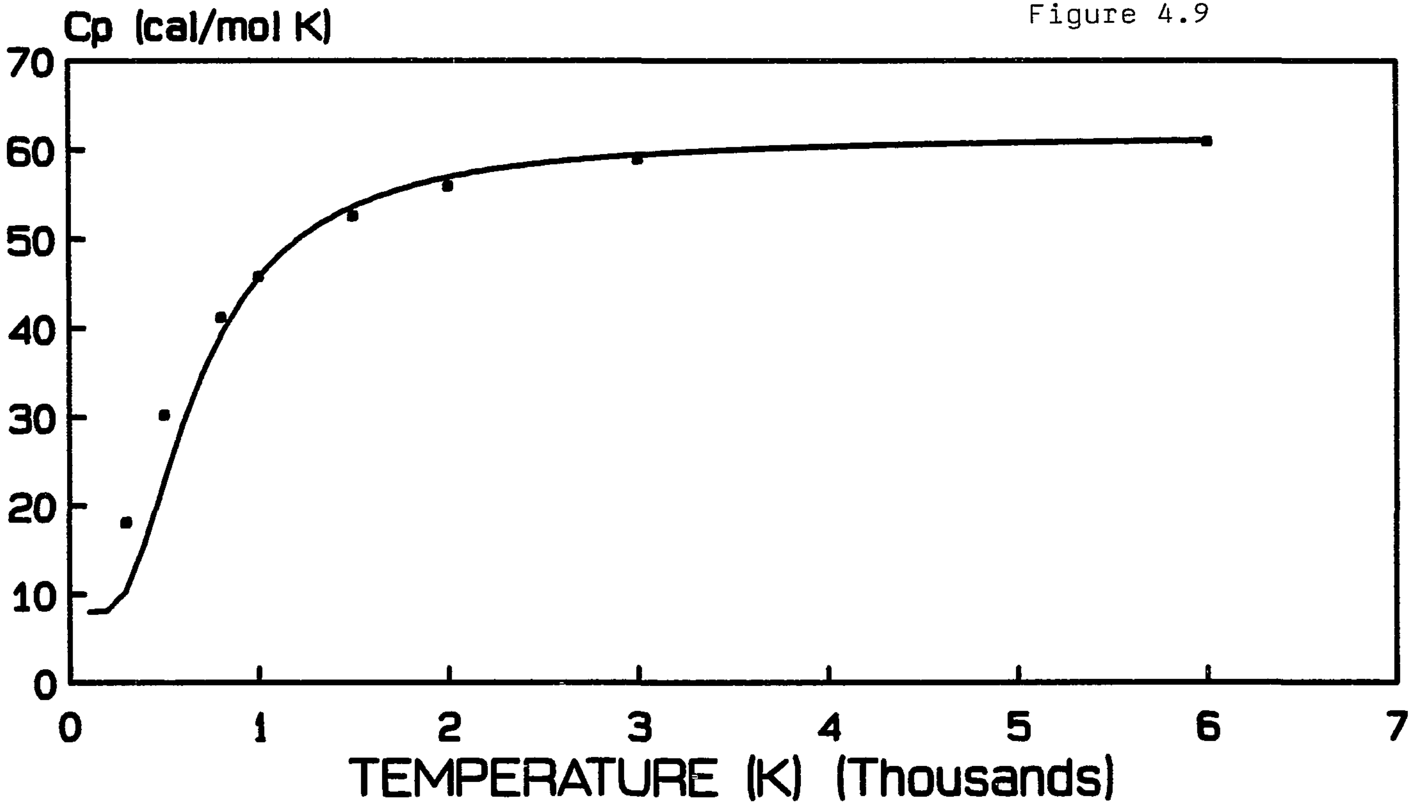
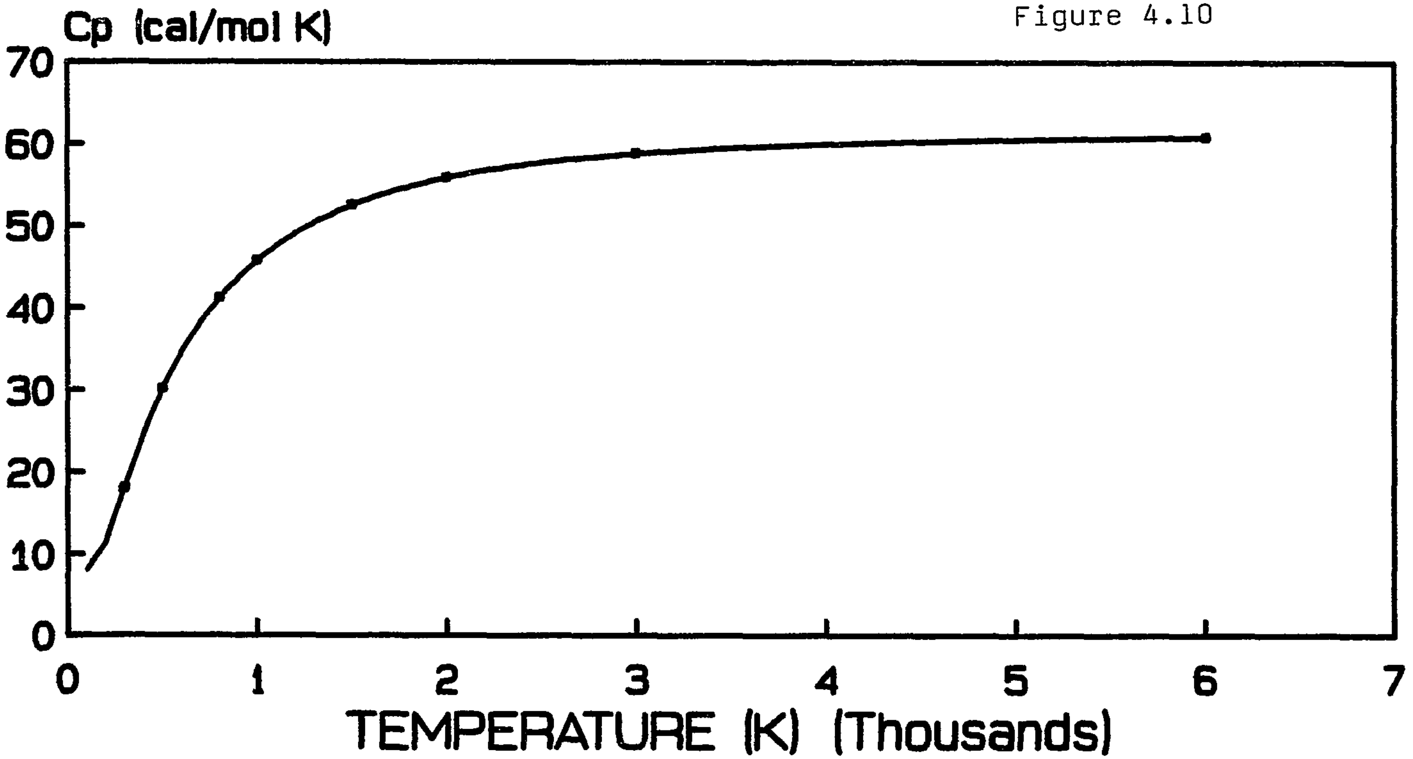


Figure 4.9

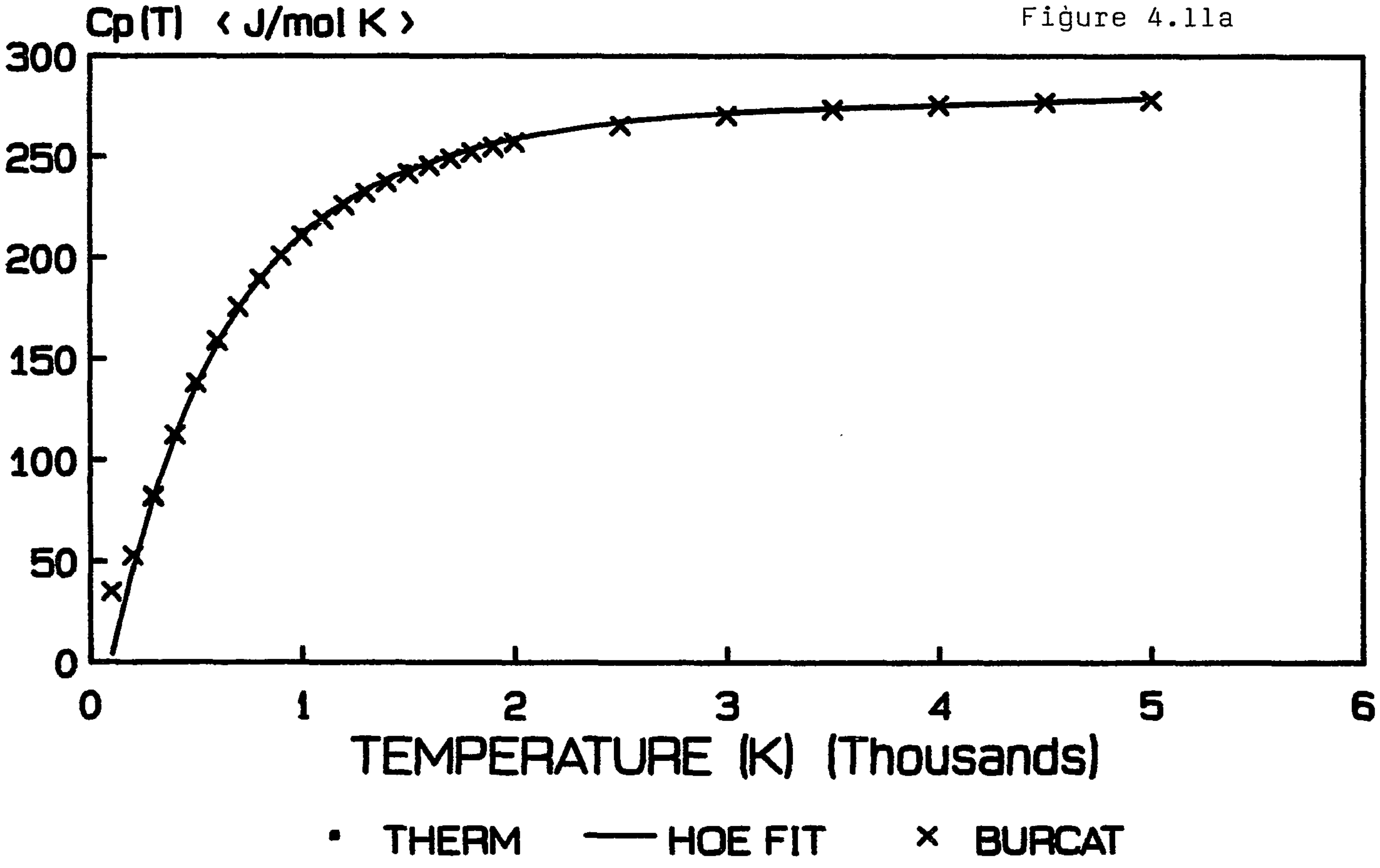
CYPD: Cp FIT vs. T
HOE vs. LITERATURE

Figure 4.10



Cp (T) : BENZENE
COMPARISON OF HOE EXTRAPOLATION W/ LIT

Figure 4.11a



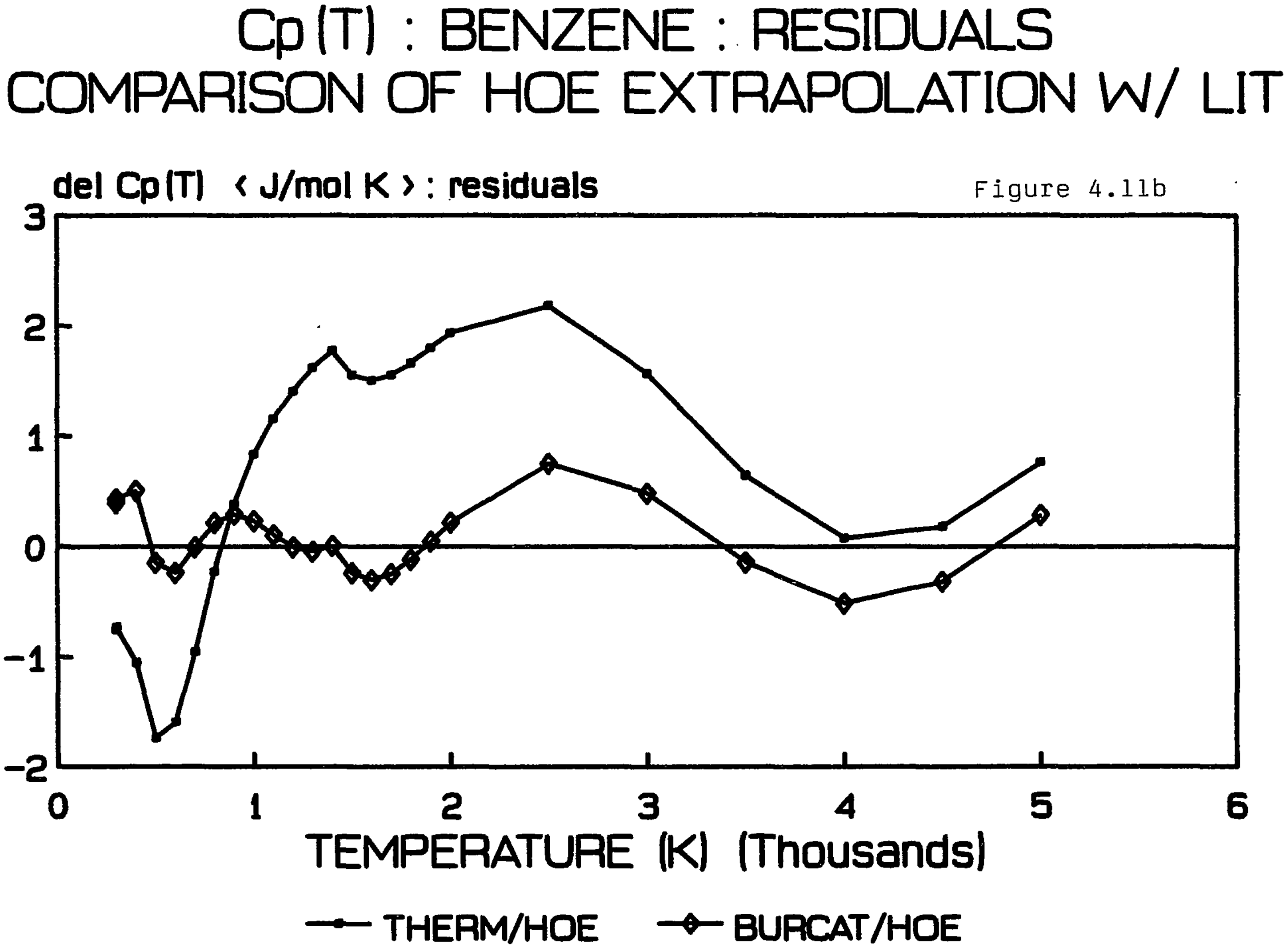

Cp(T) : BENZENE : RESIDUALS
COMPARISON OF HOE EXTRAPOLATION W/ LIT
del Cp(T) < J/mol K > : residuals
3
2
1
0
−1
−2
0
1
2
3
4
5
6
TEMPERATURE (K) (Thousands)
THERM/HOE
BURCAT/HOE

Figure 4.11b

4,12

C2Cl6 : fit to HOE

Cp(T) ASSUMING 0 AND 1 ROTOR

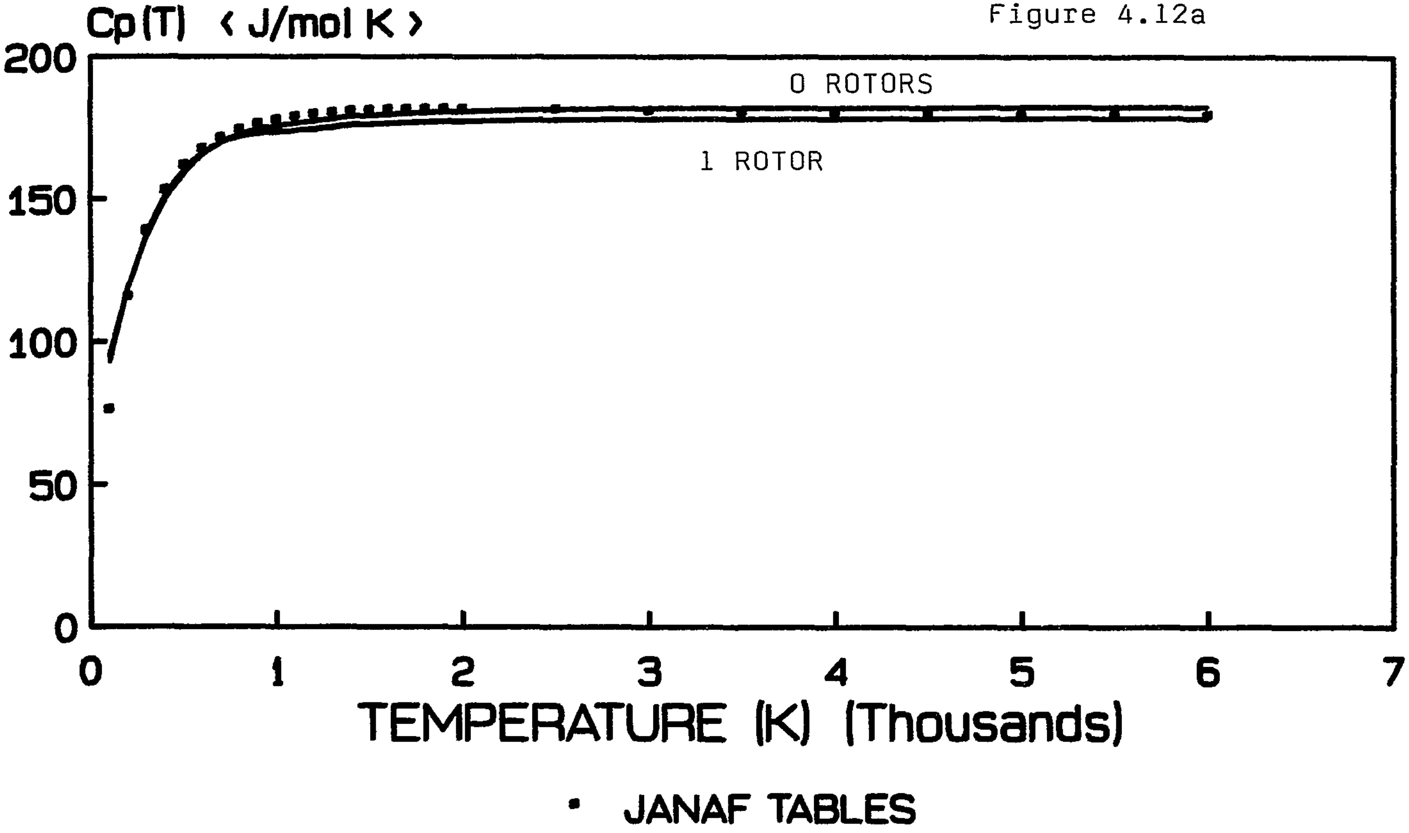


Figure 4.12a

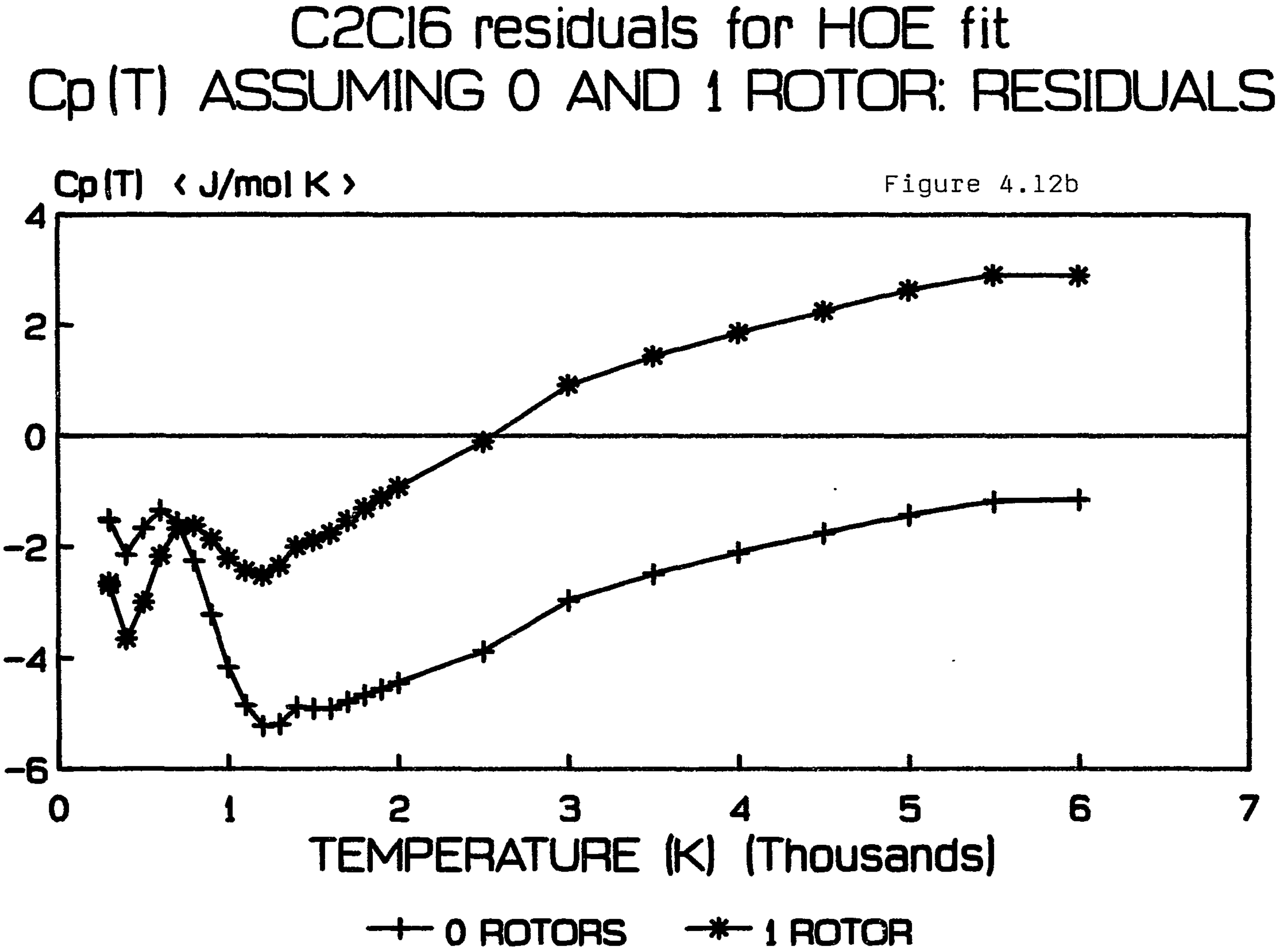

C2Cl6 residuals for HOE fit
Cp(T) ASSUMING 0 AND 1 ROTOR: RESIDUALS
Cp(T) ‹ J/mol K ›
4
2
0
-2
-4
-6
0
1
2
3
4
5
6
7
TEMPERATURE (K) (Thousands)
—+— 0 ROTORS
—✱— 1 ROTOR

Figure 4.12b

C2Cl6 :ENTHALPY FUNCTION (H(T)-H(298))
Hf(T) ASSUMING 0 AND 1 ROTOR

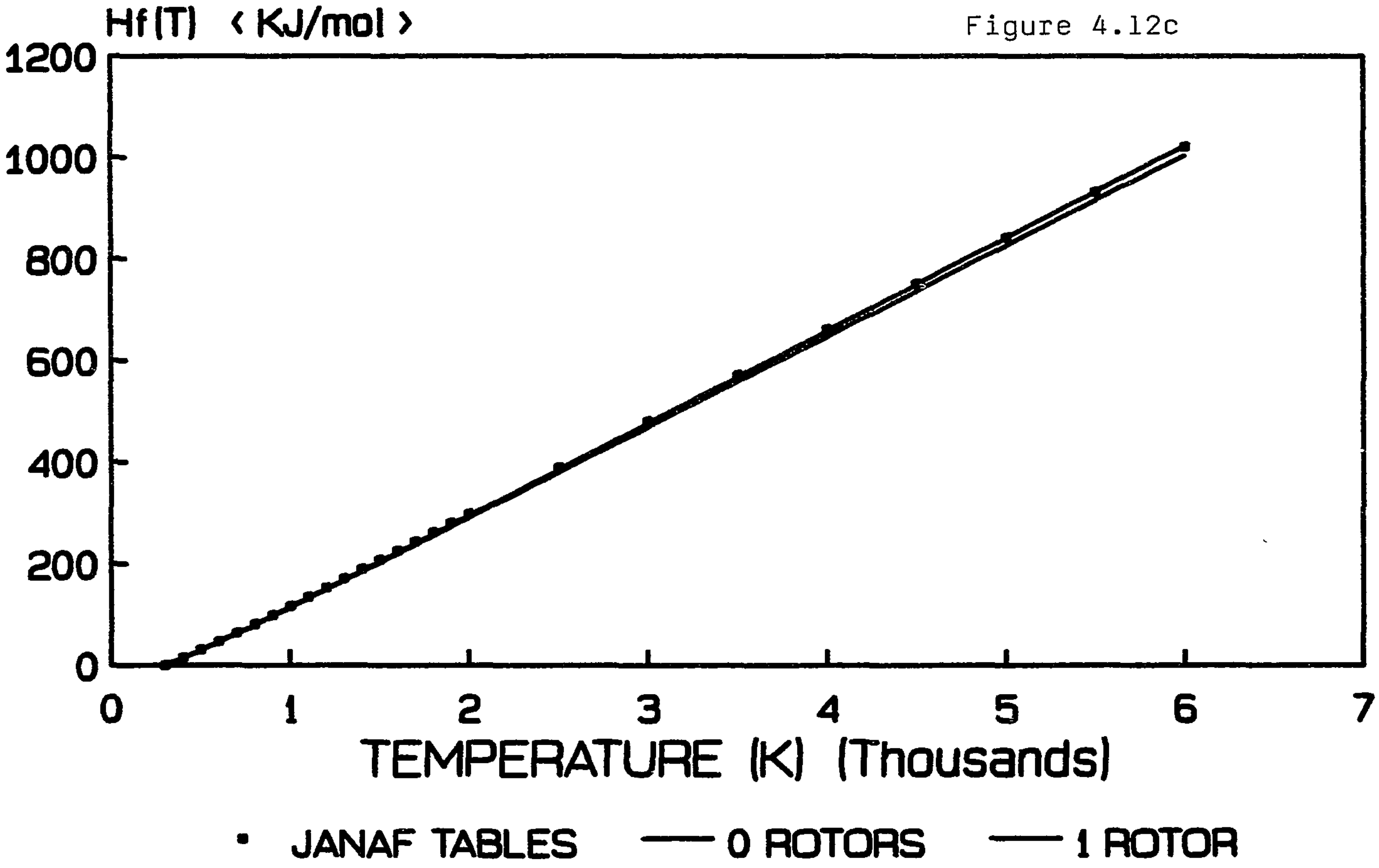
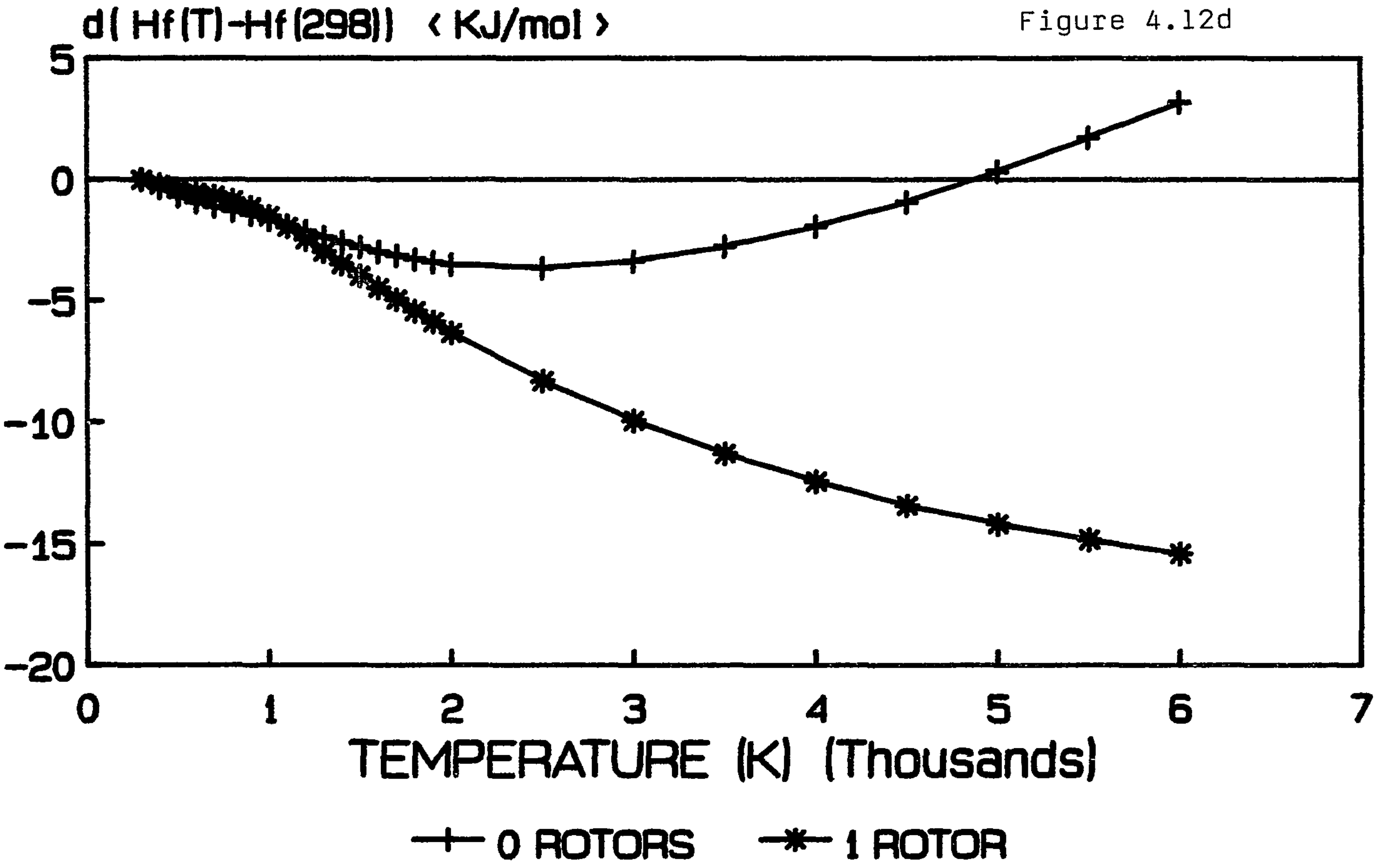


Figure 4.12c

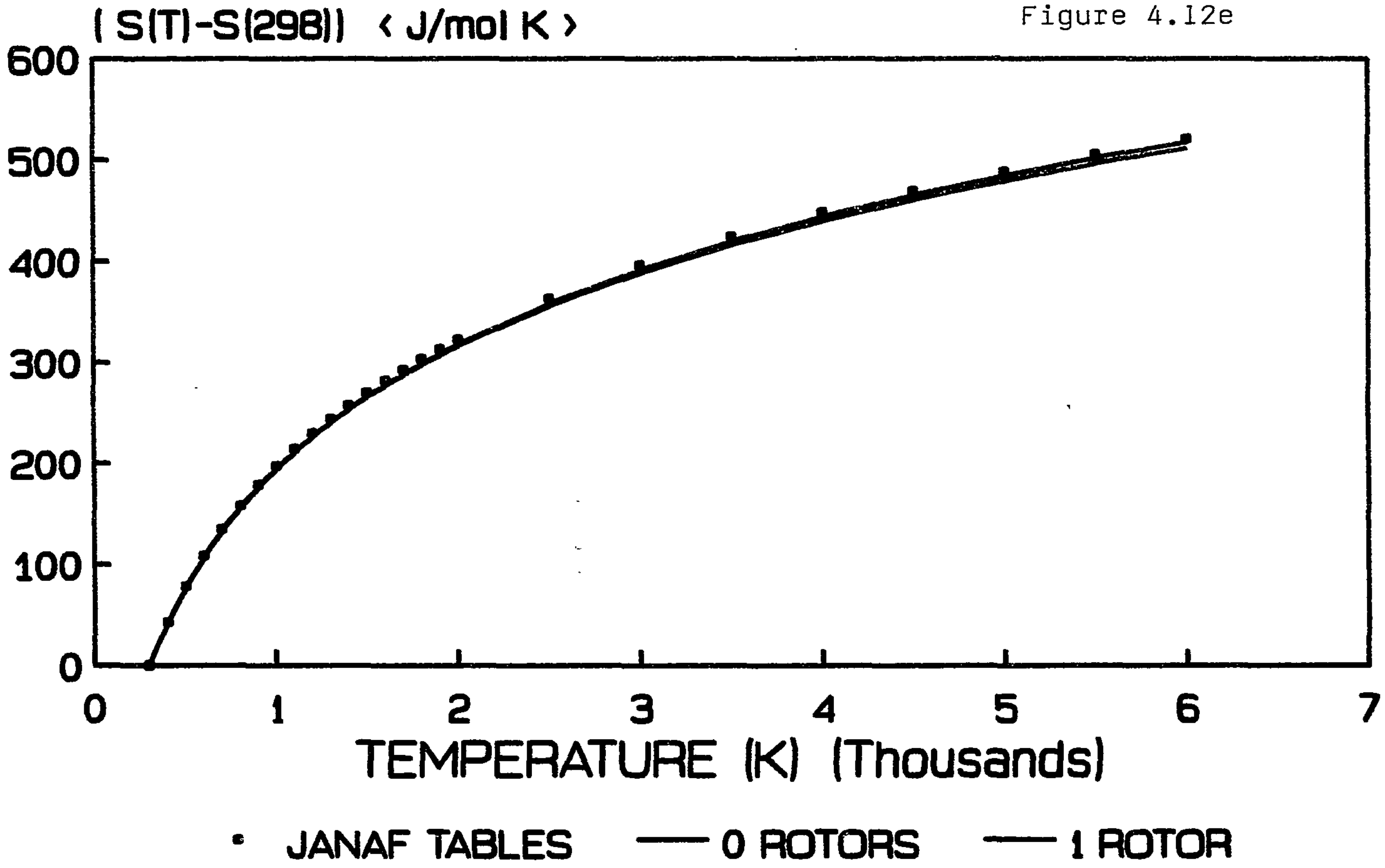
C2Cl6 :ENTHALPY F'N (H(T)-H(298)):RESID
Hf(T) ASSUMING 0 AND 1 ROTOR: RESIDUALS

Figure 4.12d



C2Cl6 :ENTROPY F'N (S(T)-S(298))
S(T) ASSUMING 0 AND 1 ROTOR

Figure 4.12e



4.1-f

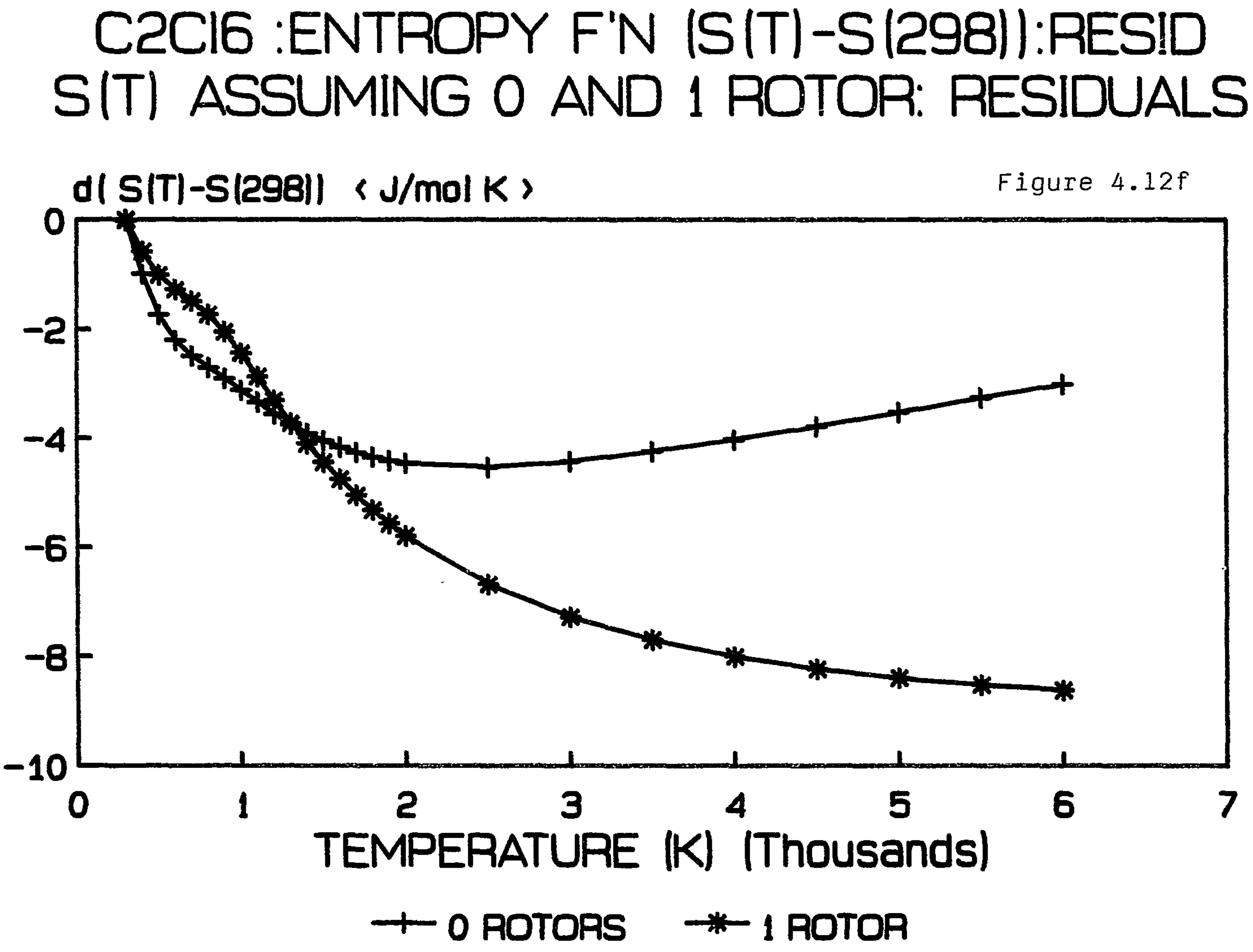
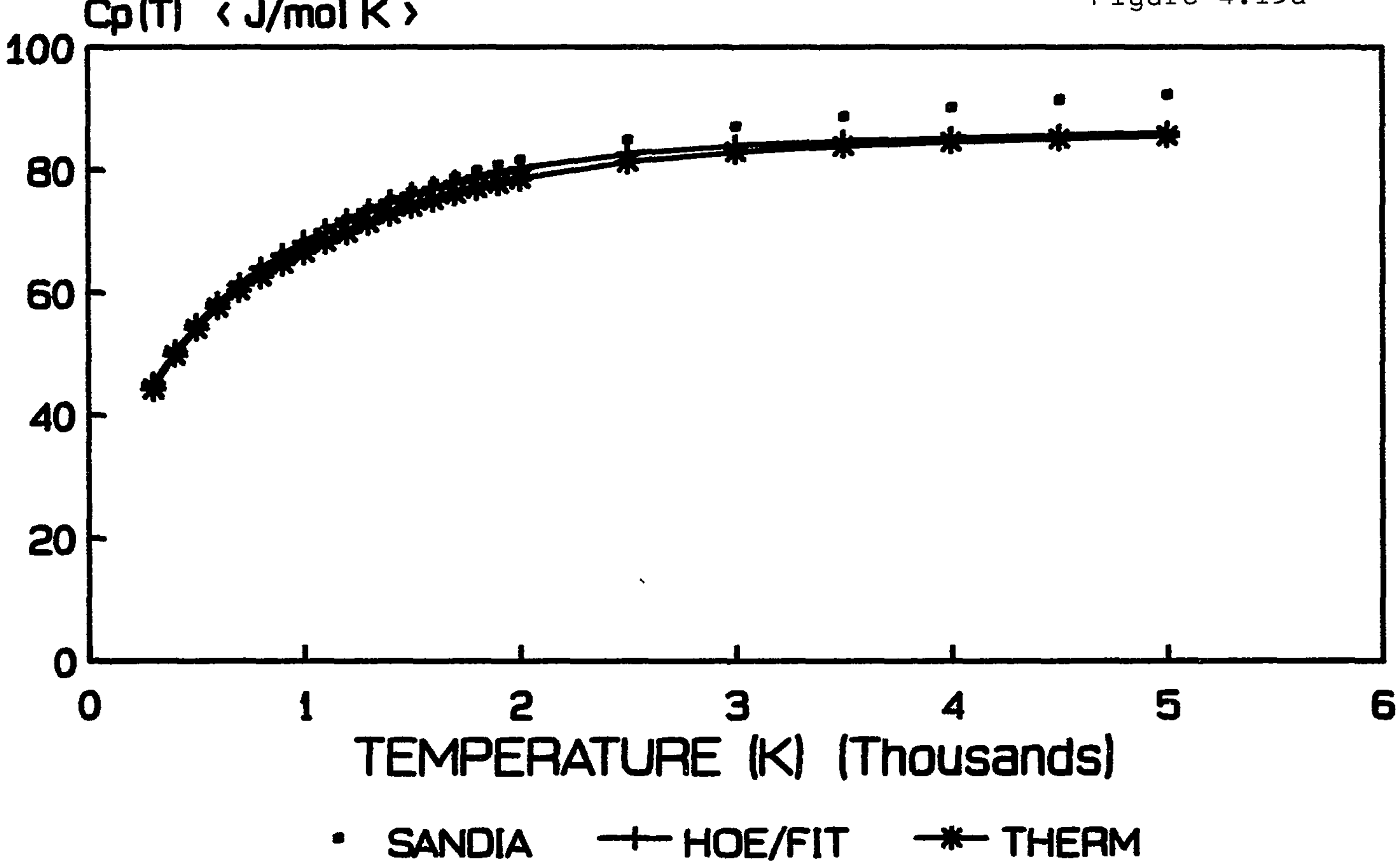


Figure 4.12f

4.13

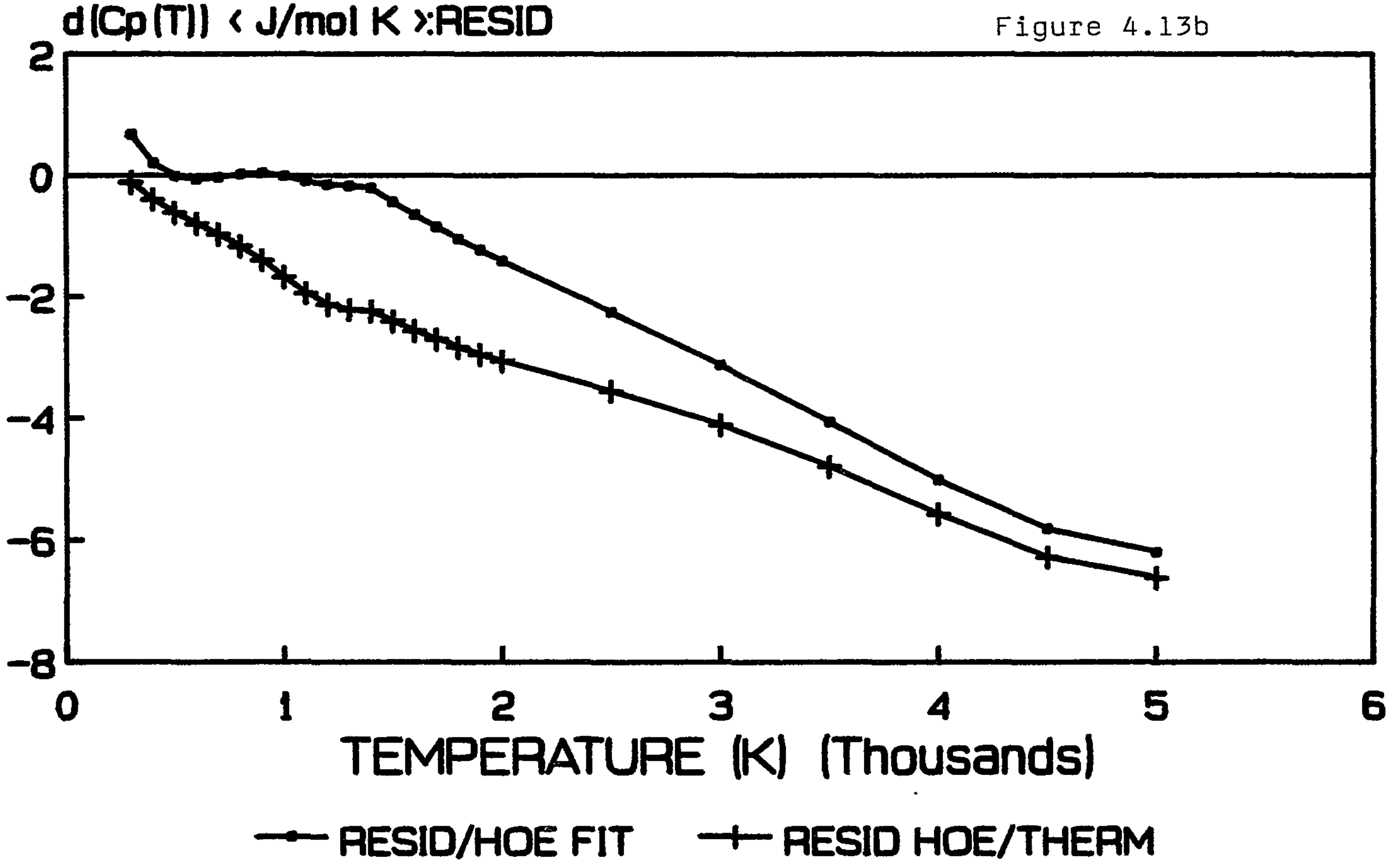
C2H2 : Cp(T) : FIT TO HOE
COMPARE SANDIA DATA W/ FIT

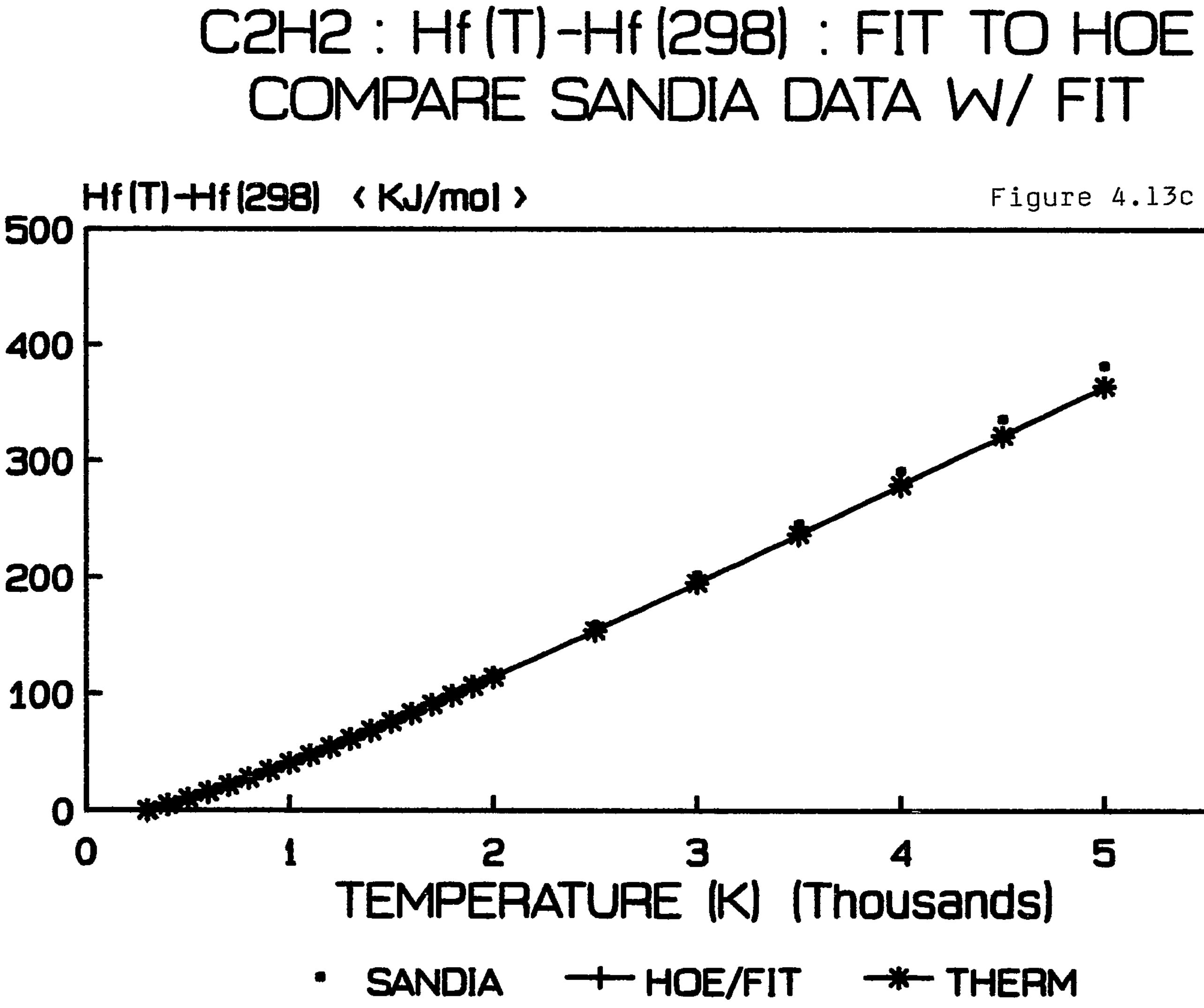
Figure 4.13a



C2H2: Cp (T) : FIT TO HOE: RESIDUALS
COMPARE SANDIA DATA W/ FIT

Figure 4.13b




C2H2 : Hf(T)-Hf(298) : FIT TO HOE
COMPARE SANDIA DATA W/ FIT
Hf(T)-Hf(298) ‹ KJ/mol ›
Figure 4.13c
500
400
300
200
100
0
0
1
2
3
4
5
6
TEMPERATURE (K) (Thousands)
SANDIA
HOE/FIT
THERM

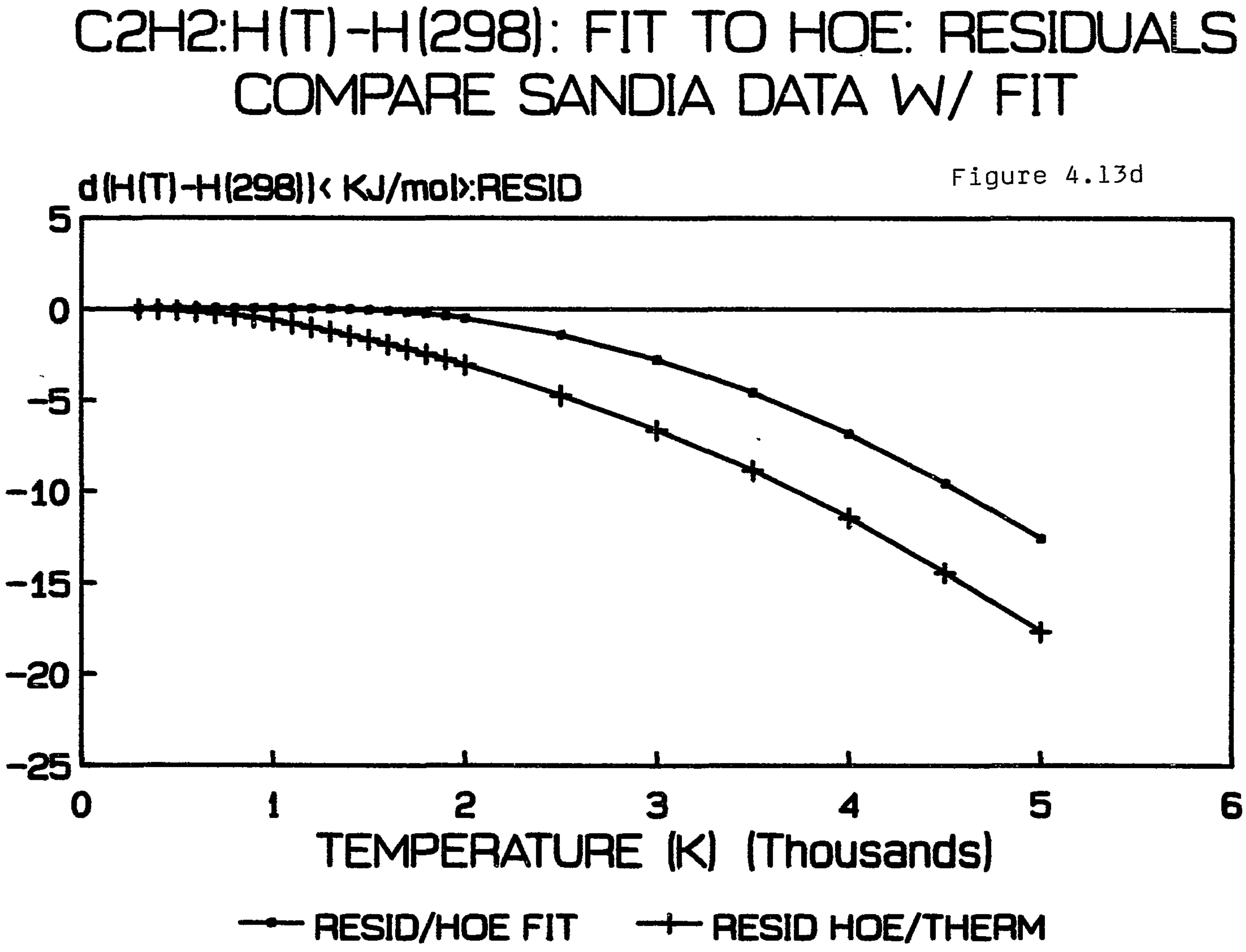

C2H2:H(T)-H(298): FIT TO HOE: RESIDUALS
COMPARE SANDIA DATA W/ FIT
d(H(T)-H(298))< KJ/mol>:RESID
5
0
-5
-10
-15
-20
-25
0
1
2
3
4
5
6
TEMPERATURE (K) (Thousands)
RESID/HOE FIT
RESID HOE/THERM

Figure 4.13d

C2H2 : S(T)-S(298) : FIT TO HOE
COMPARE SANDIA DATA W/ FIT

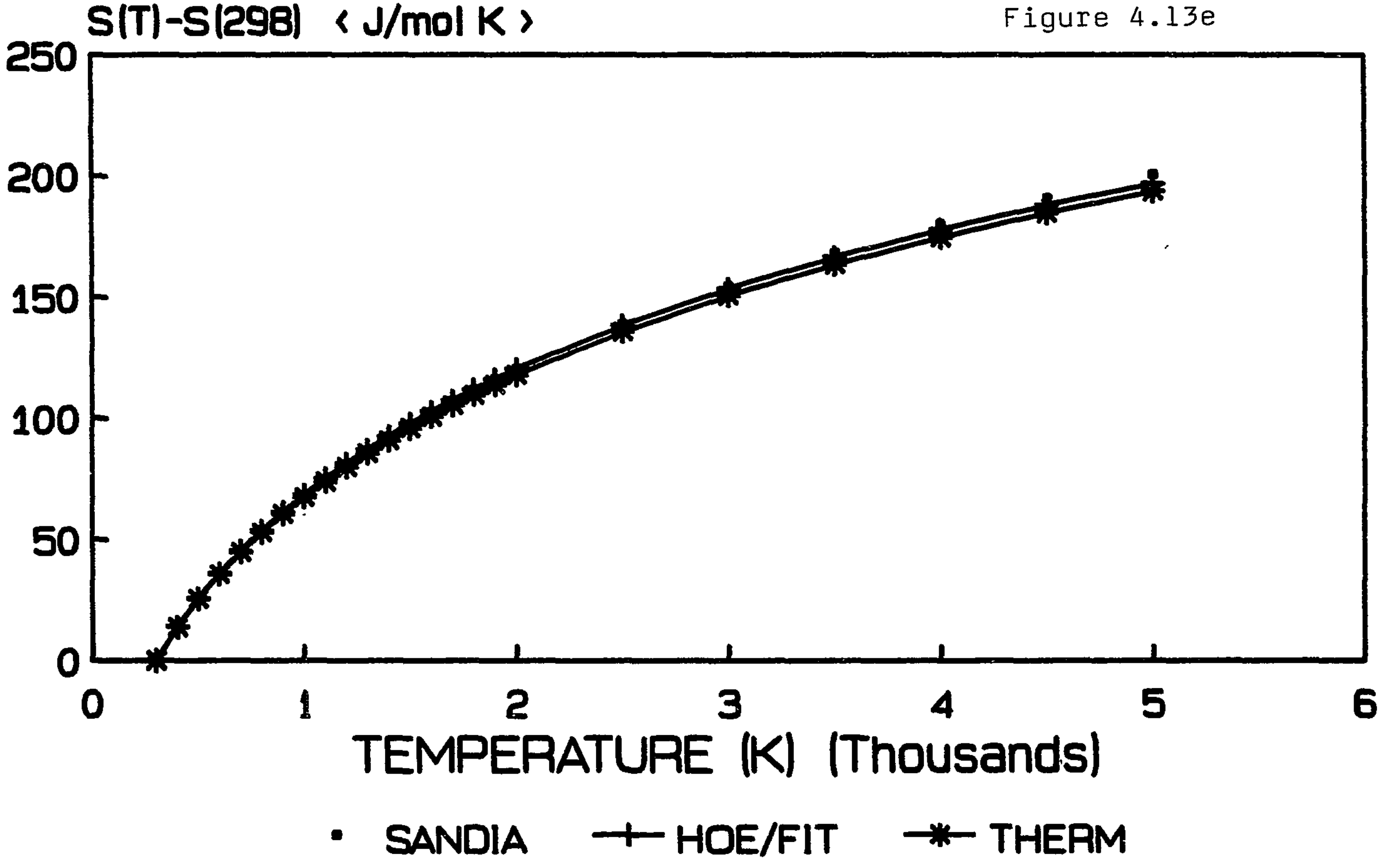
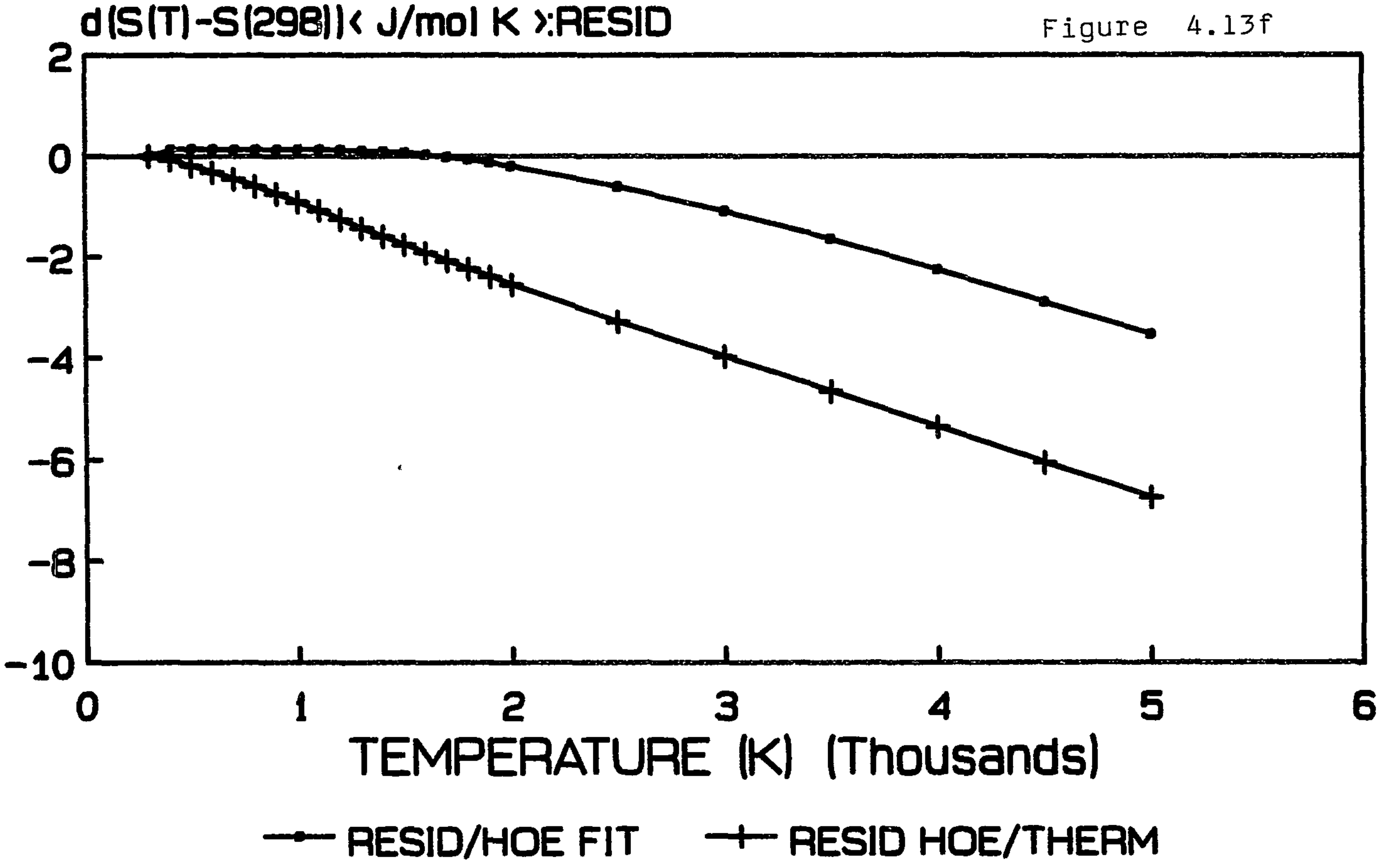


Figure 4.13e

C2H2:S(T)-S(298): FIT TO HOE: RESIDUALS
COMPARE SANDIA DATA W/ FIT

Figure 4.13f

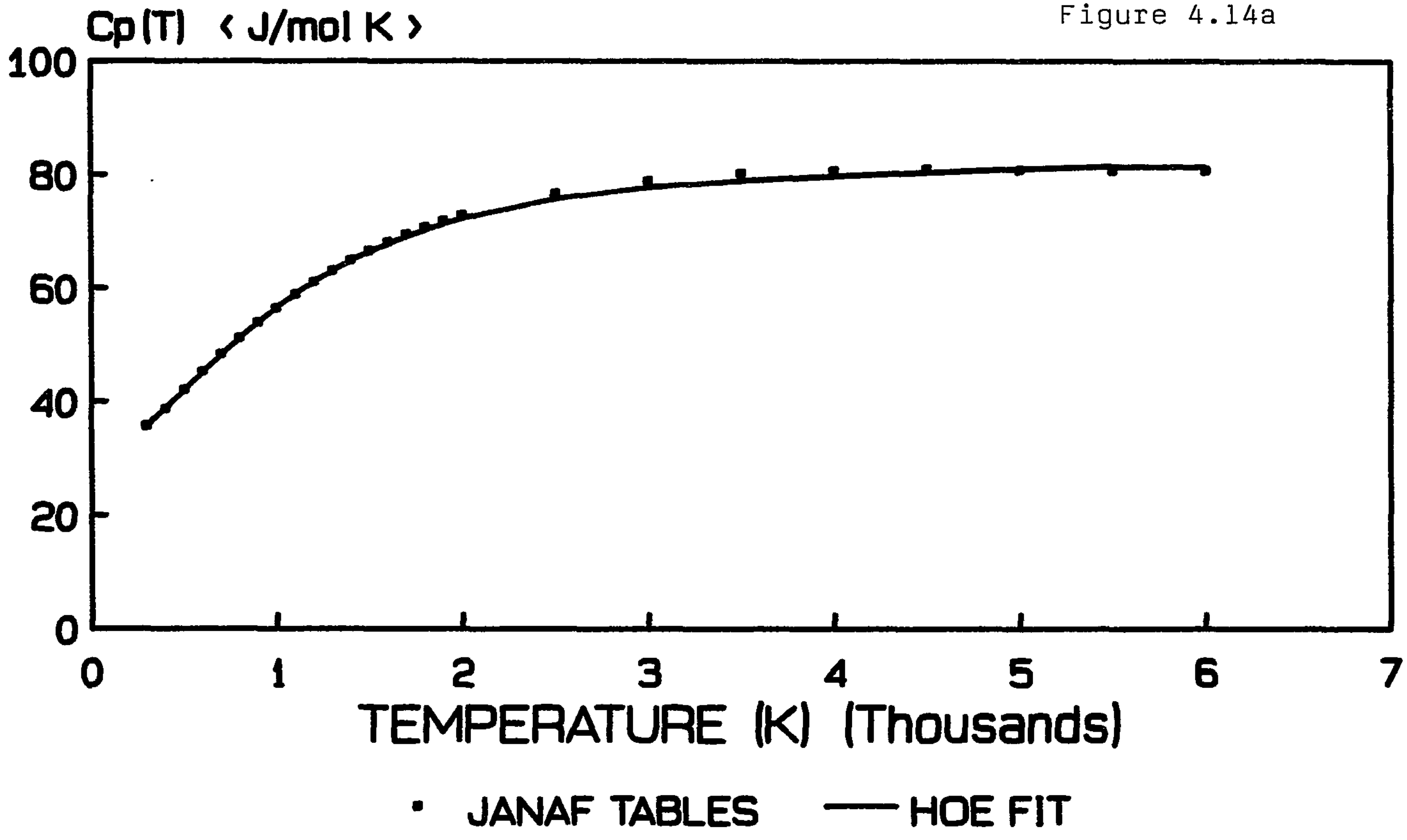


4.17

NH3 : Cp(T) FIT TO HOE

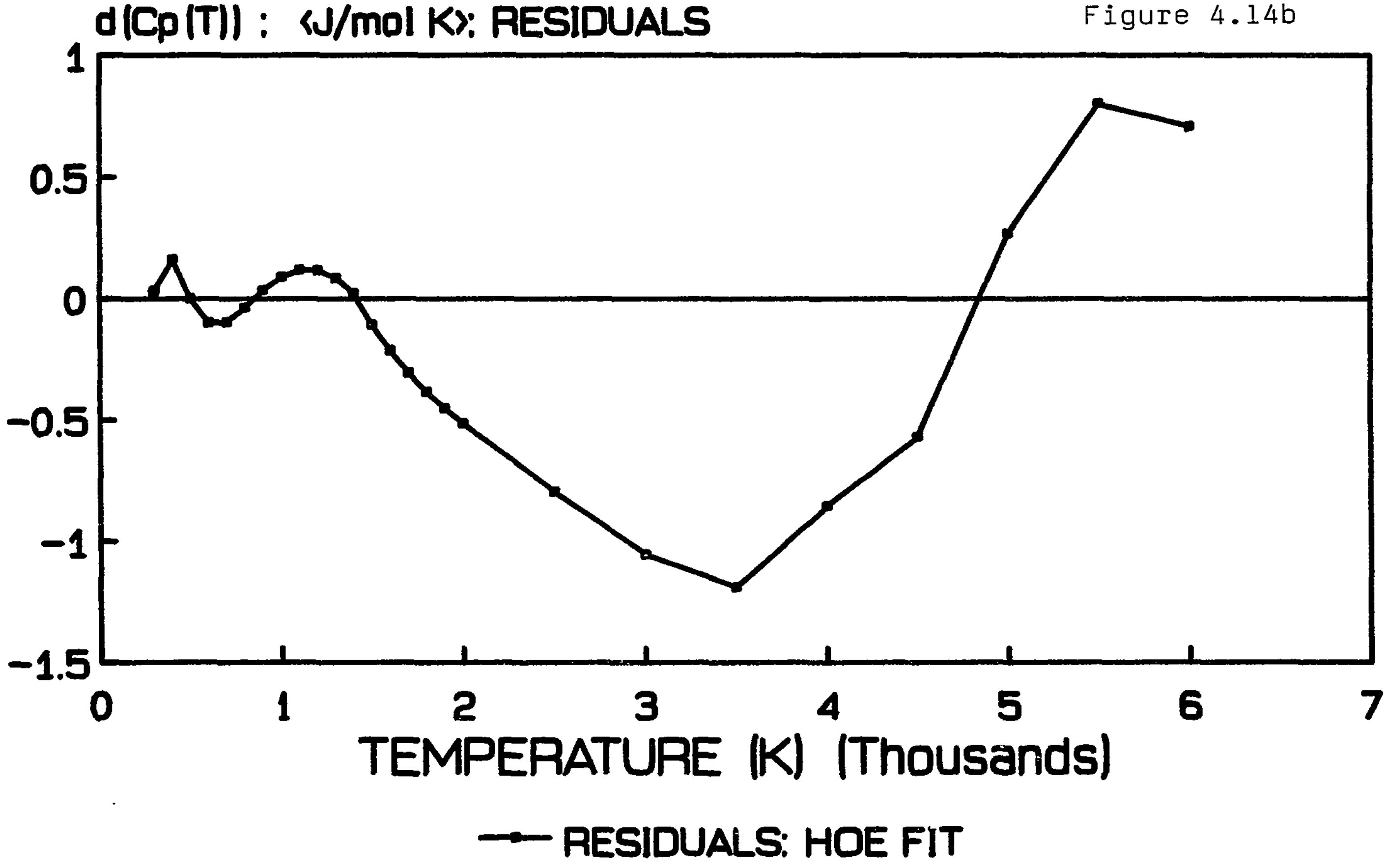
COMPARE LITERATURE W/ FIT

Figure 4.14a



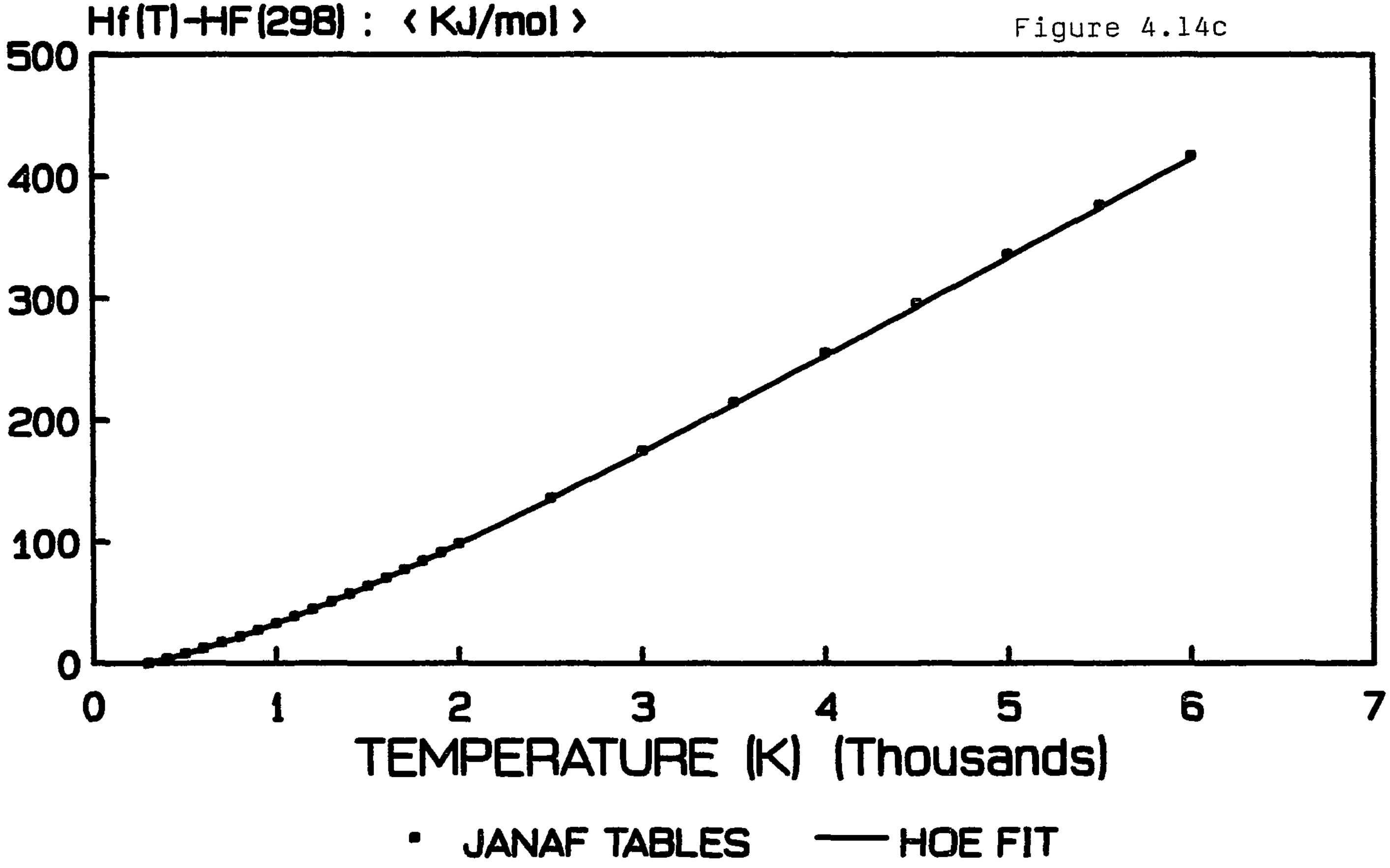
NH3: Cp(T) : FIT TO HOE: RESIDUALS
COMPARE LITERATURE W/ FIT

Figure 4.14b



NH3 : H(T)-H(298): FIT TO HOE
COMPARE LITERATURE W/ FIT

Figure 4.14c



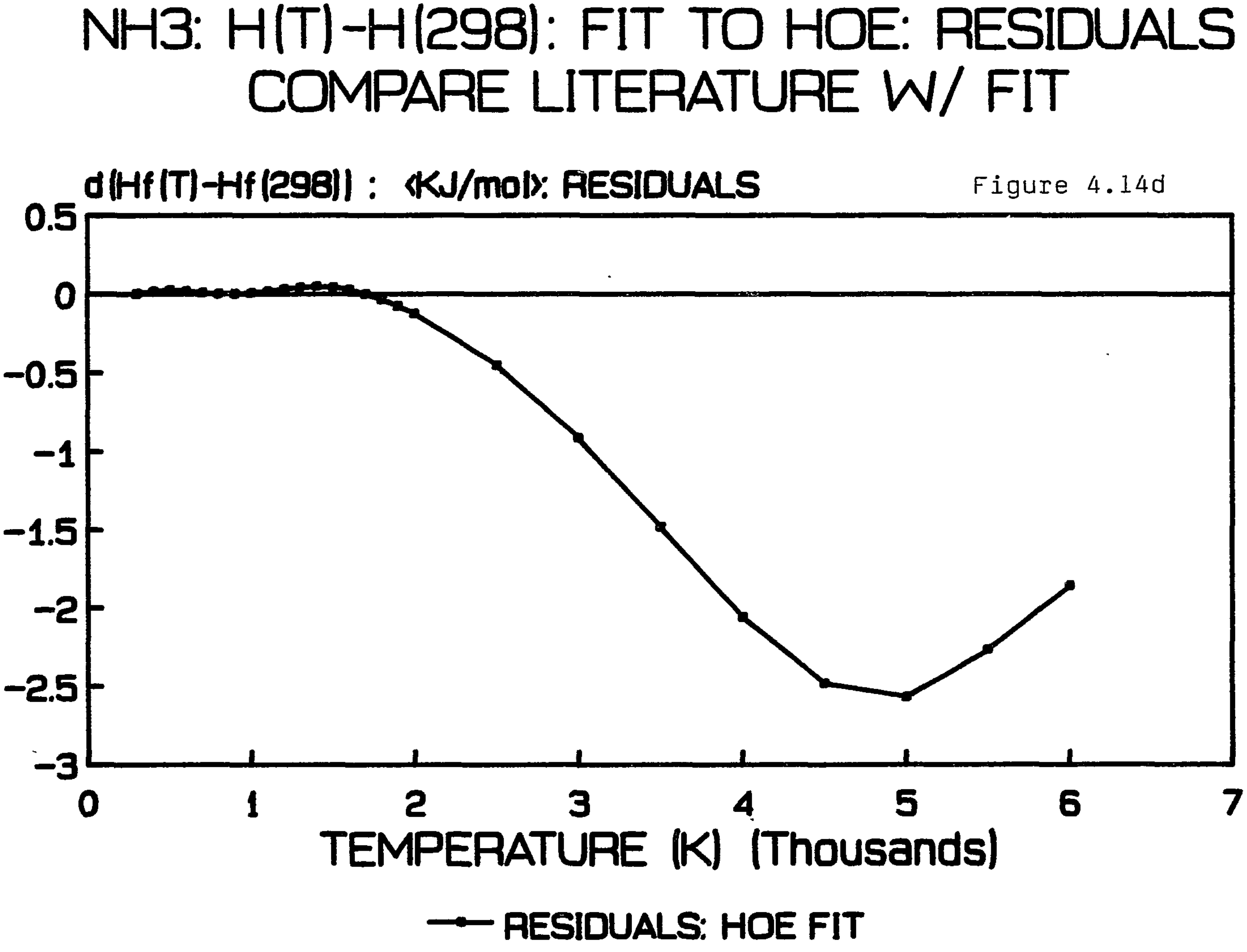
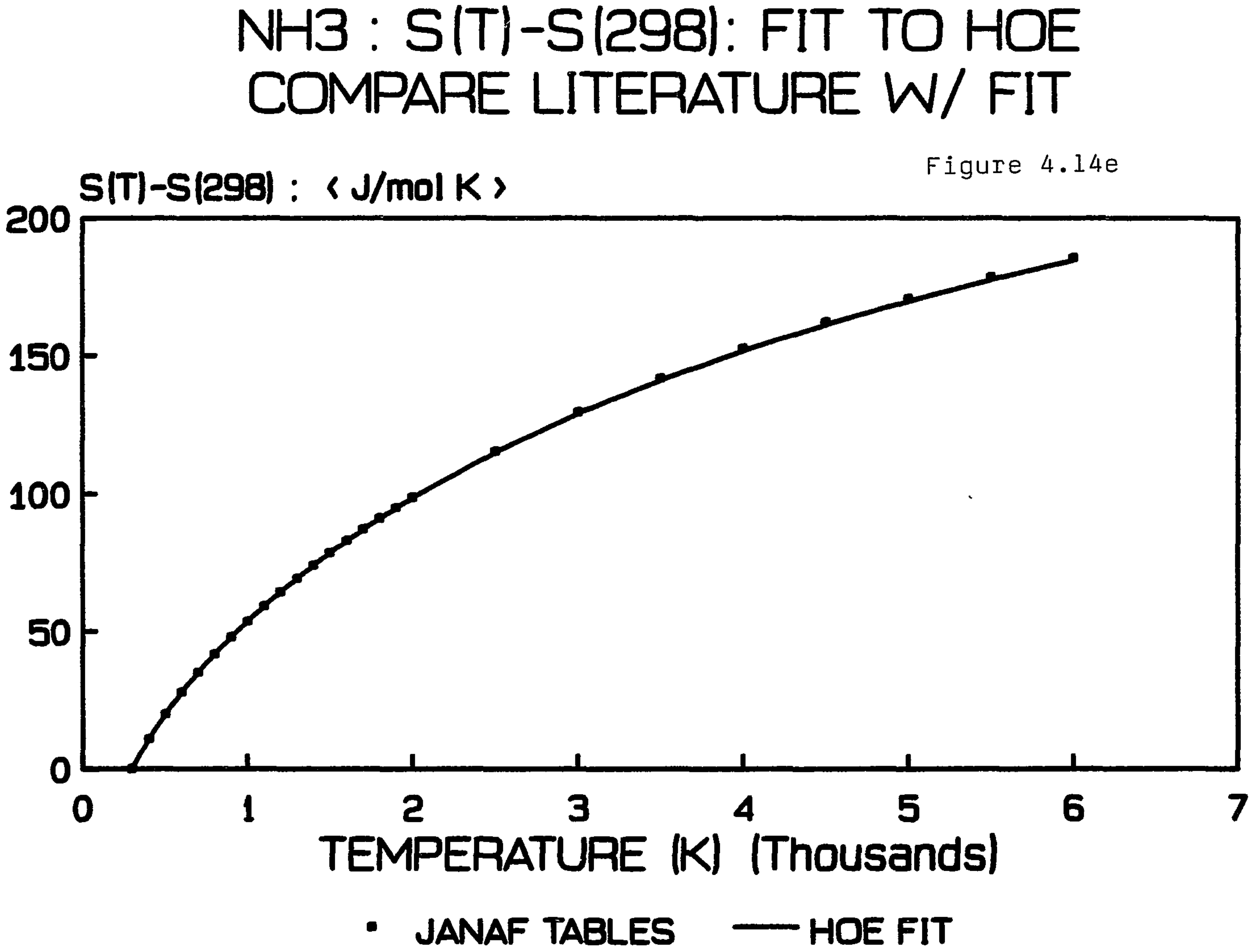

NH3: H(T)-H(298): FIT TO HOE: RESIDUALS
COMPARE LITERATURE W/ FIT
d(Hf(T)-Hf(298)) : <KJ/mol>: RESIDUALS
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
0
1
2
3
4
5
6
7
TEMPERATURE (K) (Thousands)
RESIDUALS: HOE FIT

Figure 4.14d

Figure 4.14e


NH3 : S(T)-S(298): FIT TO HOE
COMPARE LITERATURE W/ FIT
S(T)-S(298) : < J/mol K >
200
150
100
50
0
0
1
2
3
4
5
6
7
TEMPERATURE (K) (Thousands)
• JANAF TABLES
—— HOE FIT

1.14f

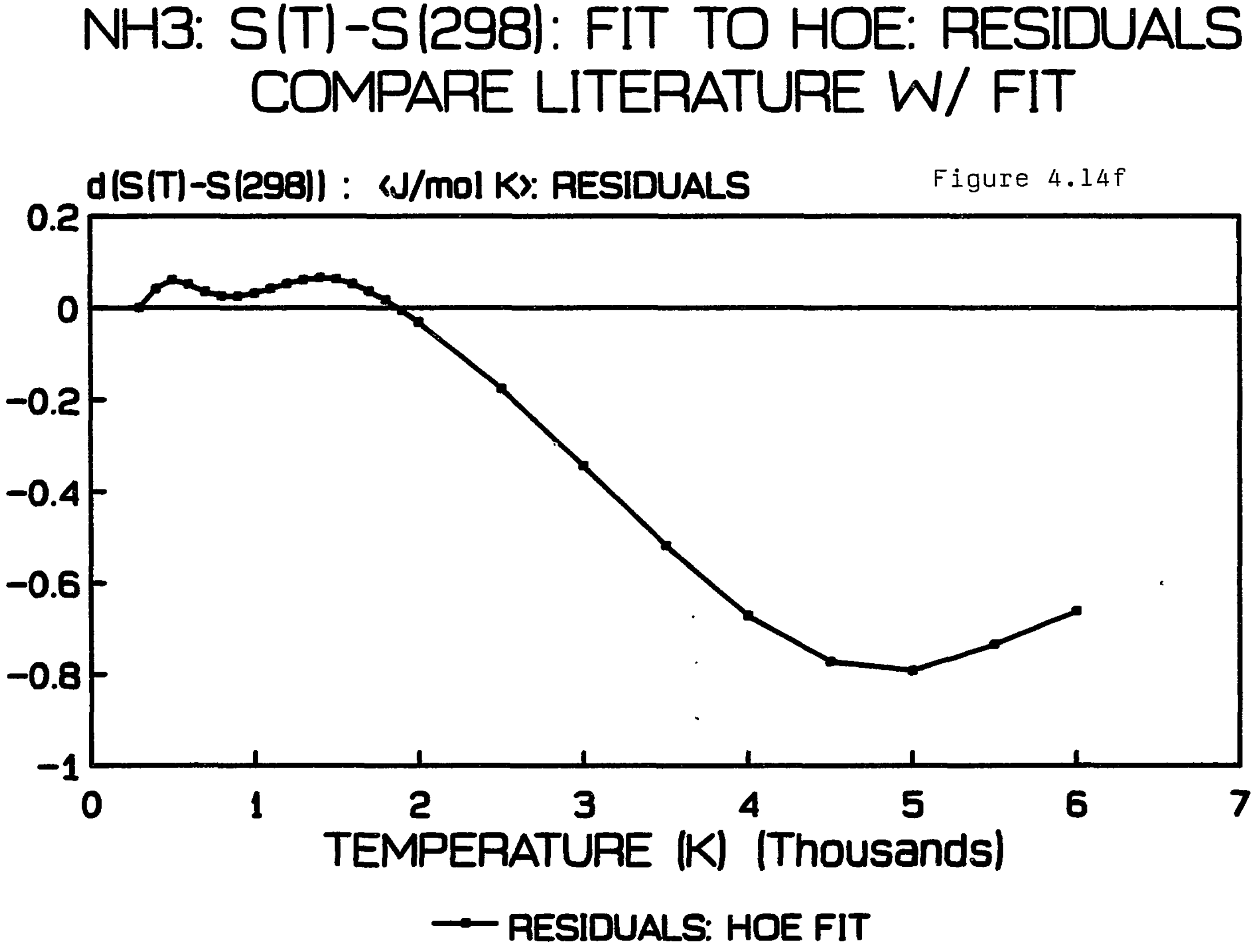
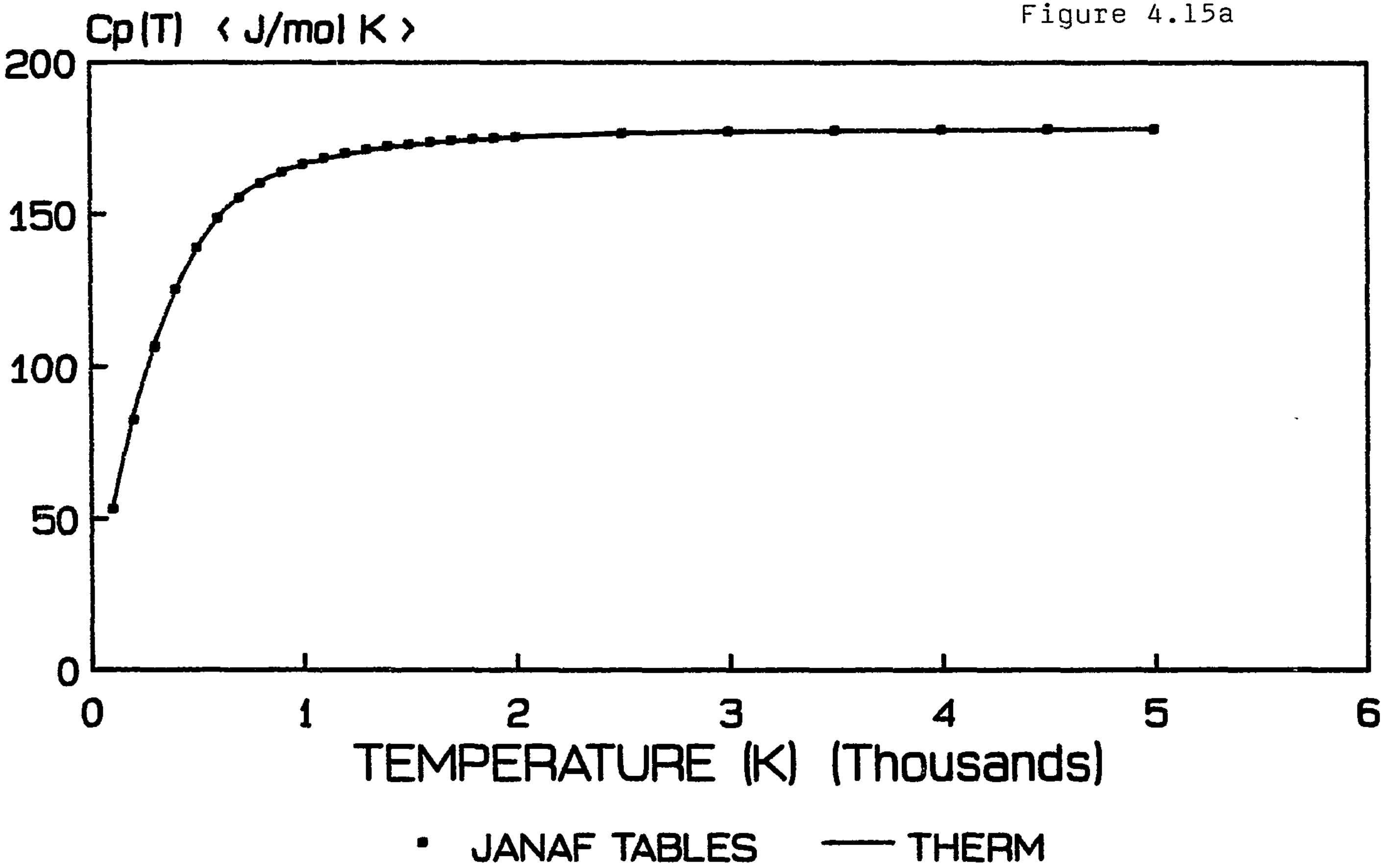


Figure 4.14f

4.15a

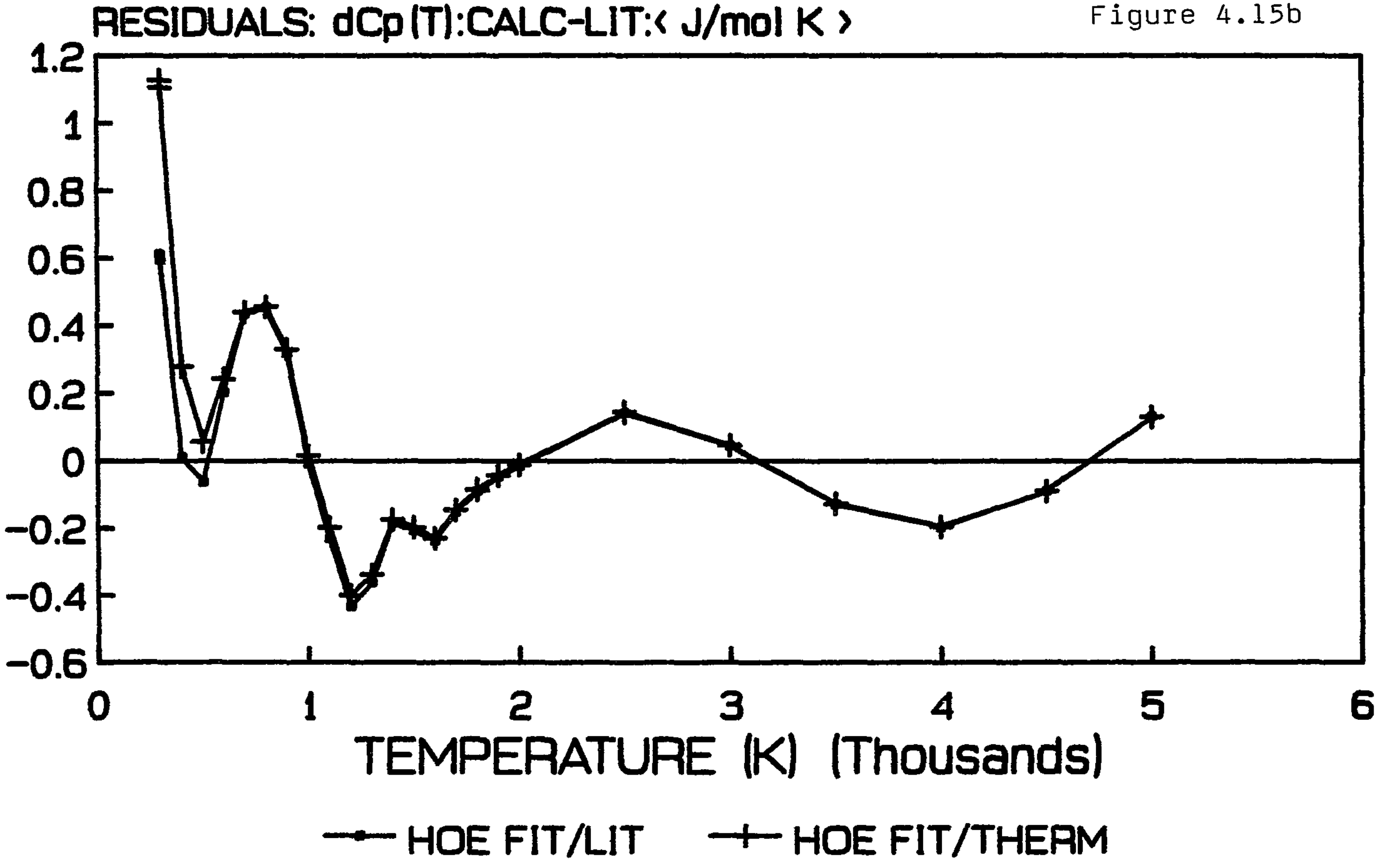
C2F6 : FIT of Cp(T) : LIT & ESTIMATED. JANAF DATA vs. THERM ESTIMATE : HOE

Figure 4.15a



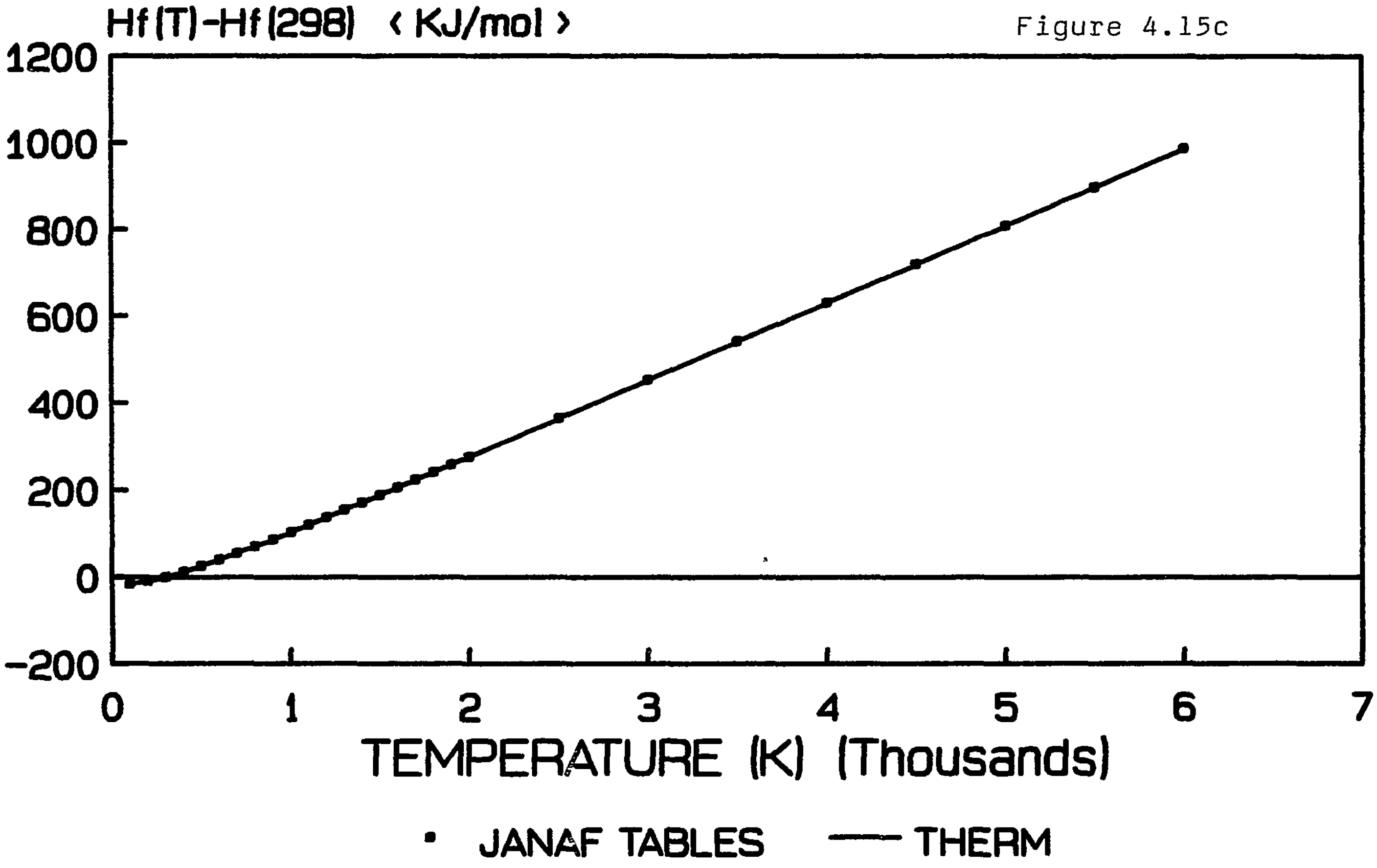
C2F6 : FIT of Cp(T) : LIT & ESTIMATED
RESIDUALS : HOE FIT TO C2F6

Figure 4.15b



C2F6 : H(T)-H(298) : LIT & ESTIMATED
JANAF DATA vs. THERM ESTIMATE : HOE

Figure 4.15c



C2F6 : H(T)-H(298) : LIT & ESTIMATED RESIDUALS : HOE FIT TO C2F6

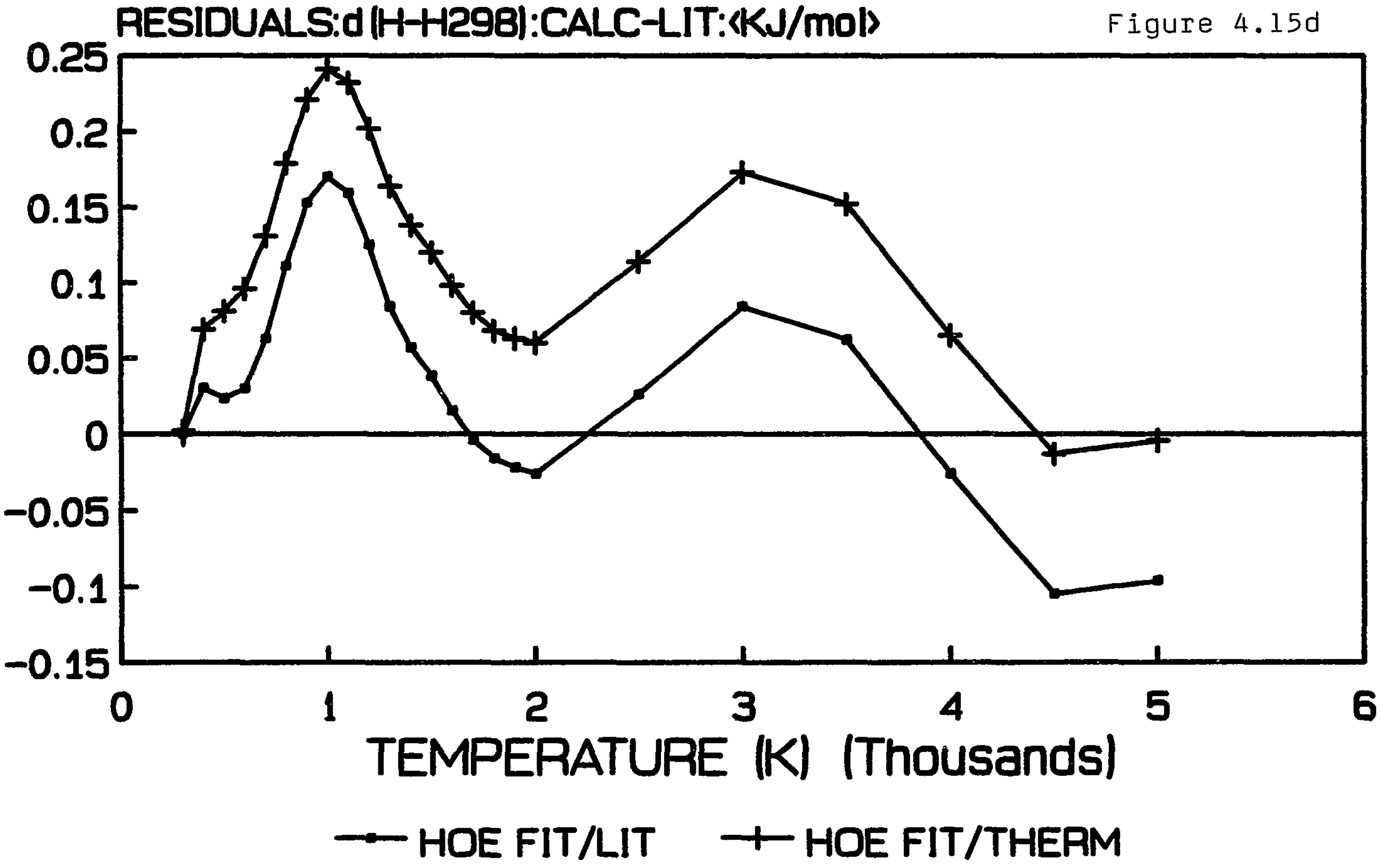
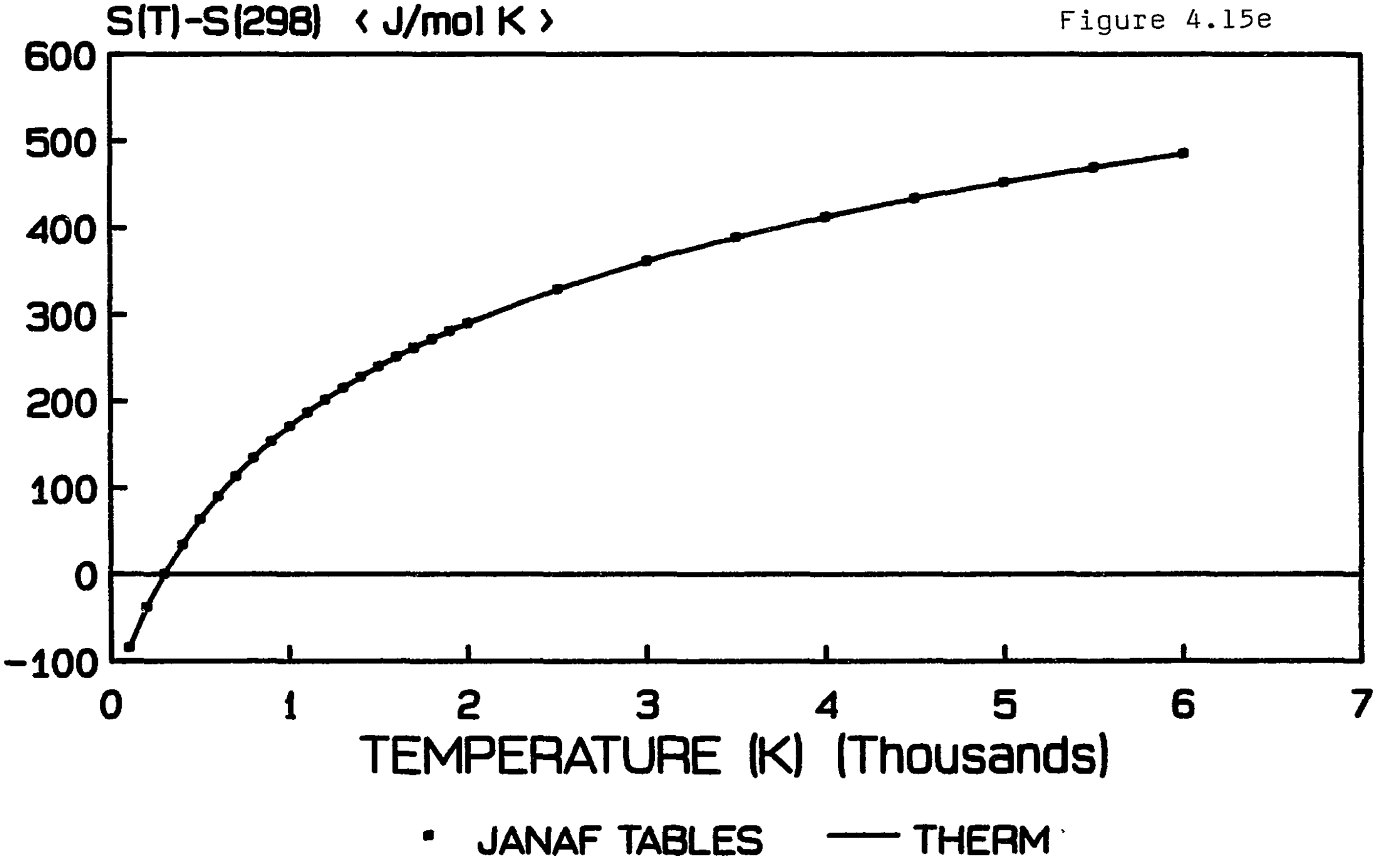


Figure 4.15d

C2F6 : S(T)-S(298) : LIT & ESTIMATED
JANAF DATA vs. THERM ESTIMATE : HOE

Figure 4.15e



4.15f

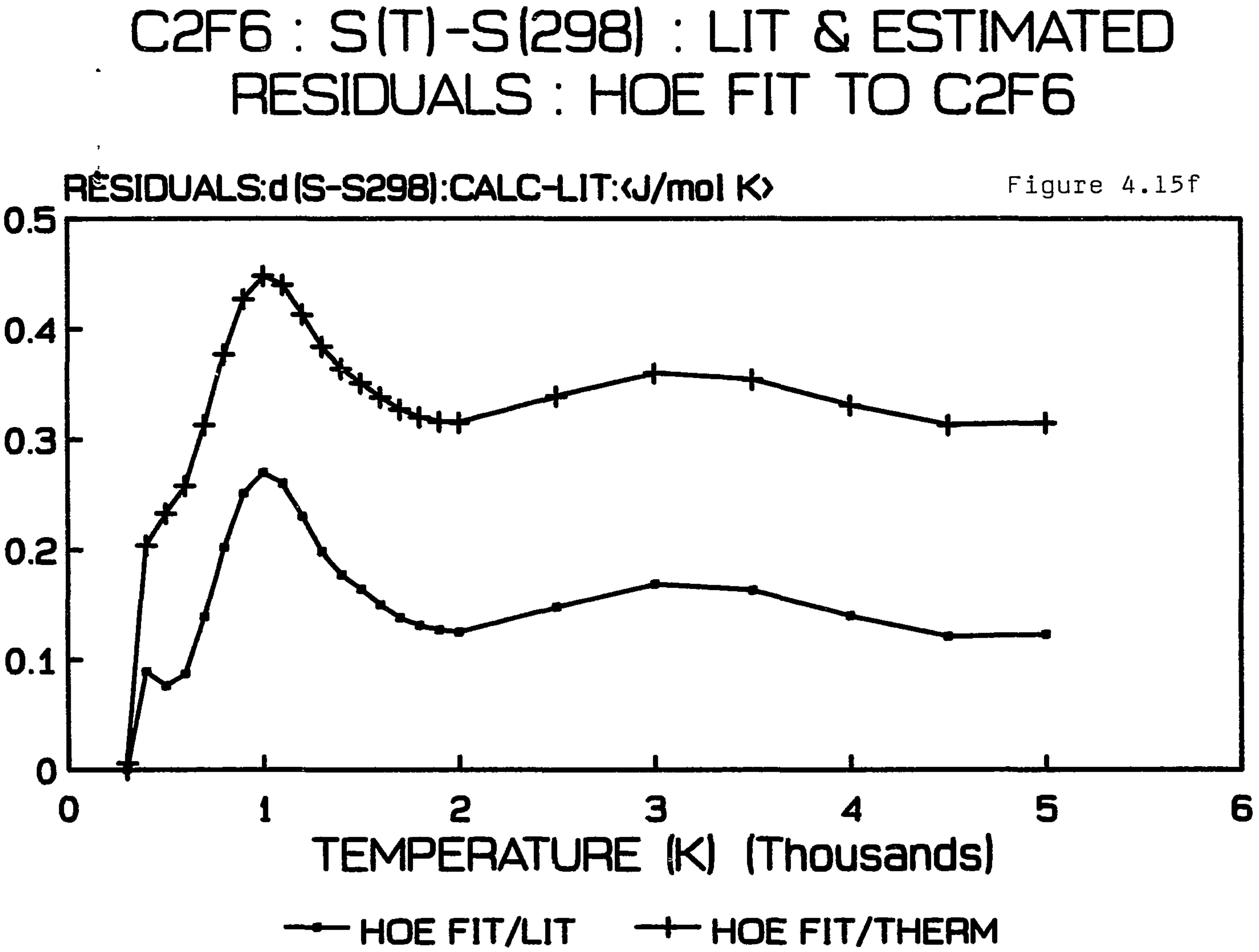
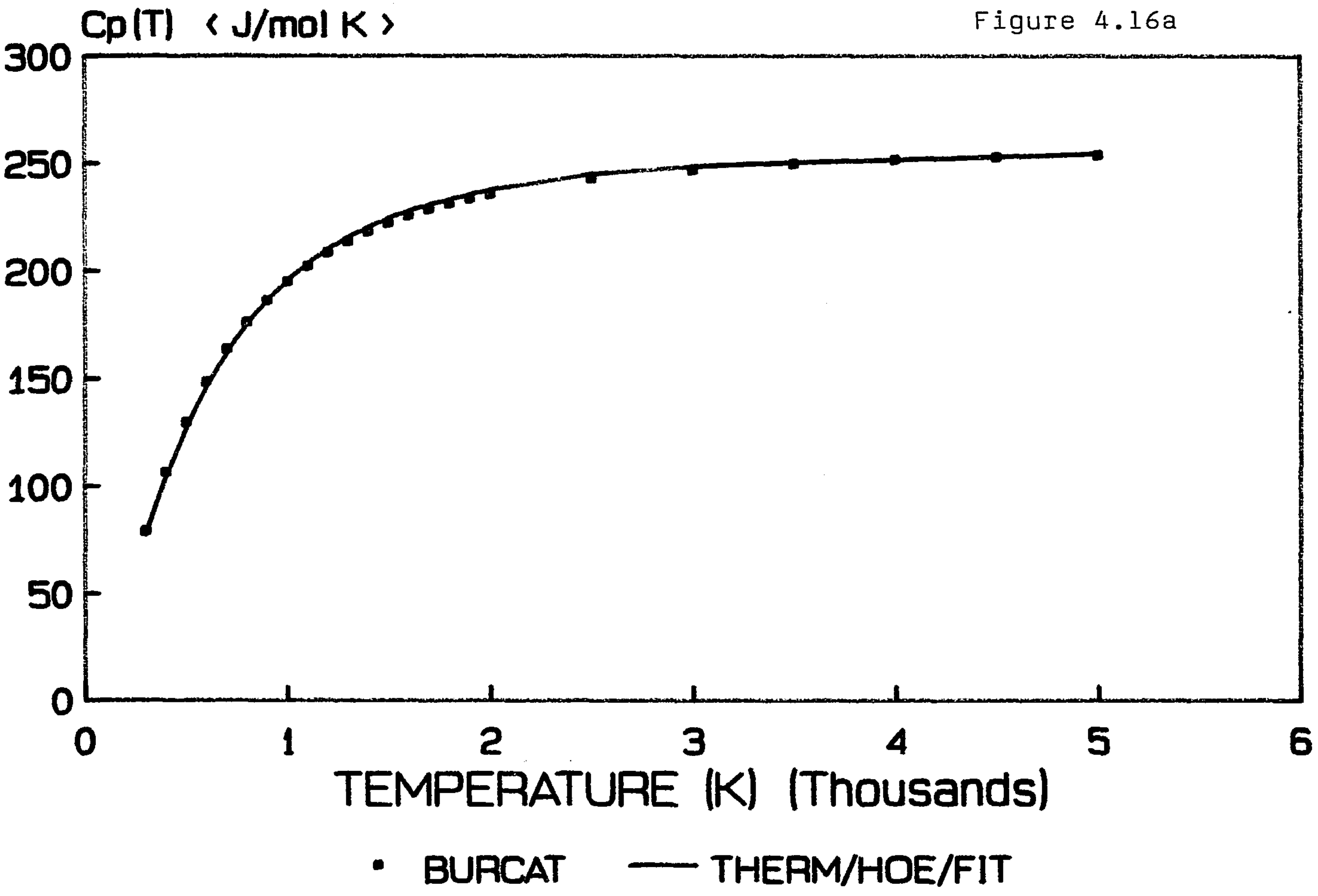


Figure 4.15f

4.16 a

PHENYL : Cp(T) : FIT OF HOE

Figure 4.16a



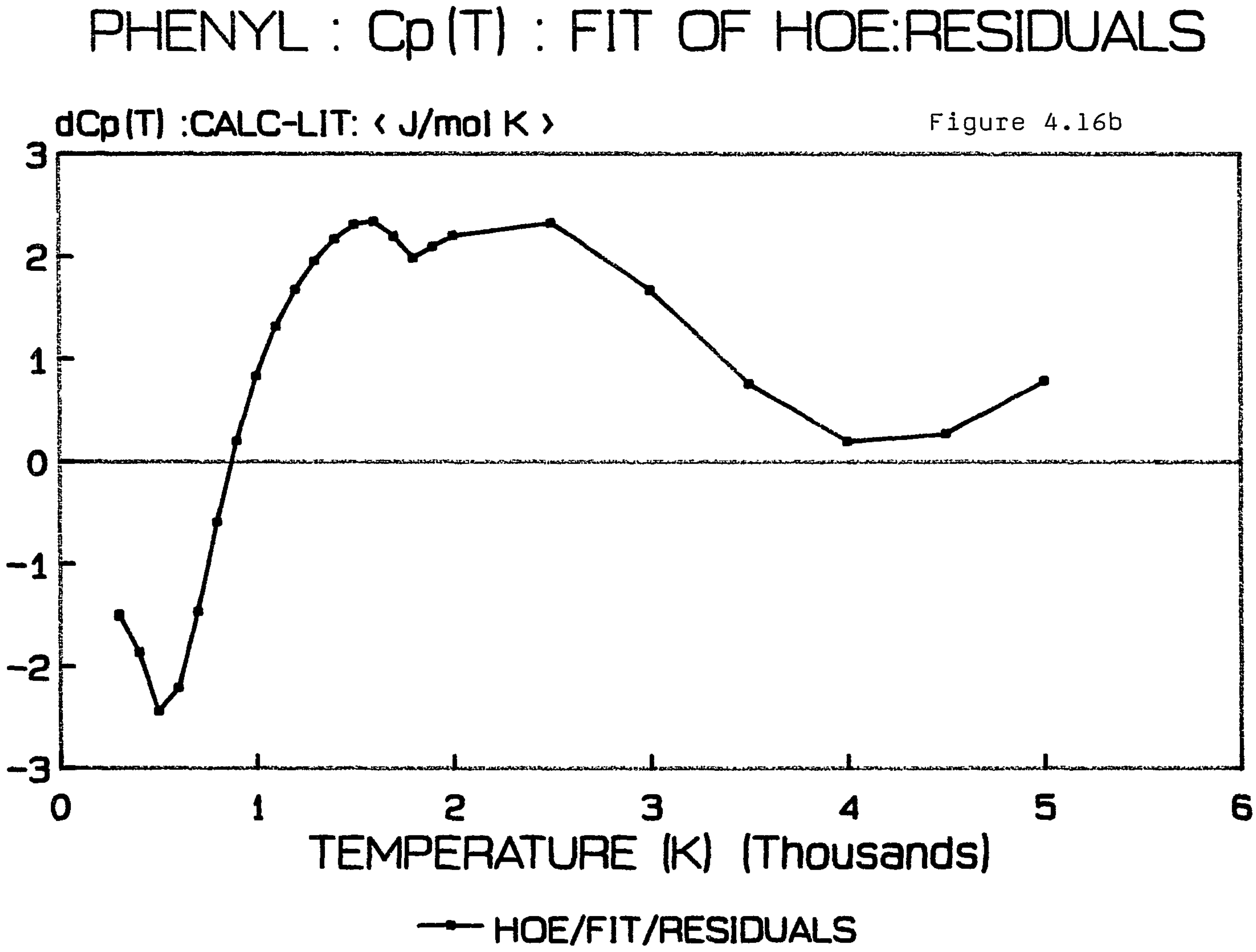


Figure 4.16b

PHENYL : Cp(T) : FIT OF HOE:RESIDUALS
dCp(T) :CALC-LIT: ‹ J/mol K ›
3
2
1
0
−1
−2
−3
0
1
2
3
4
5
6
TEMPERATURE (K) (Thousands)
HOE/FIT/RESIDUALS

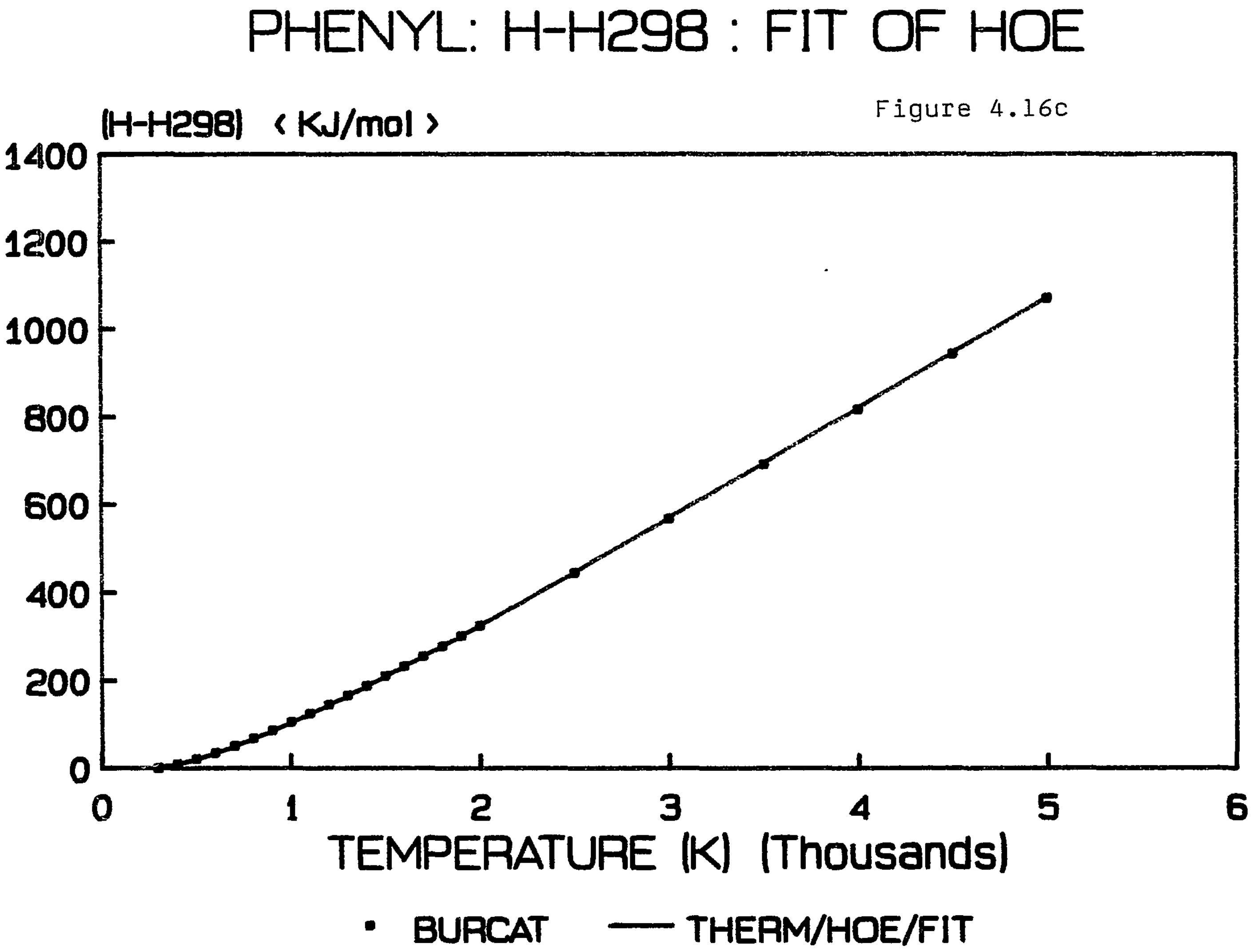
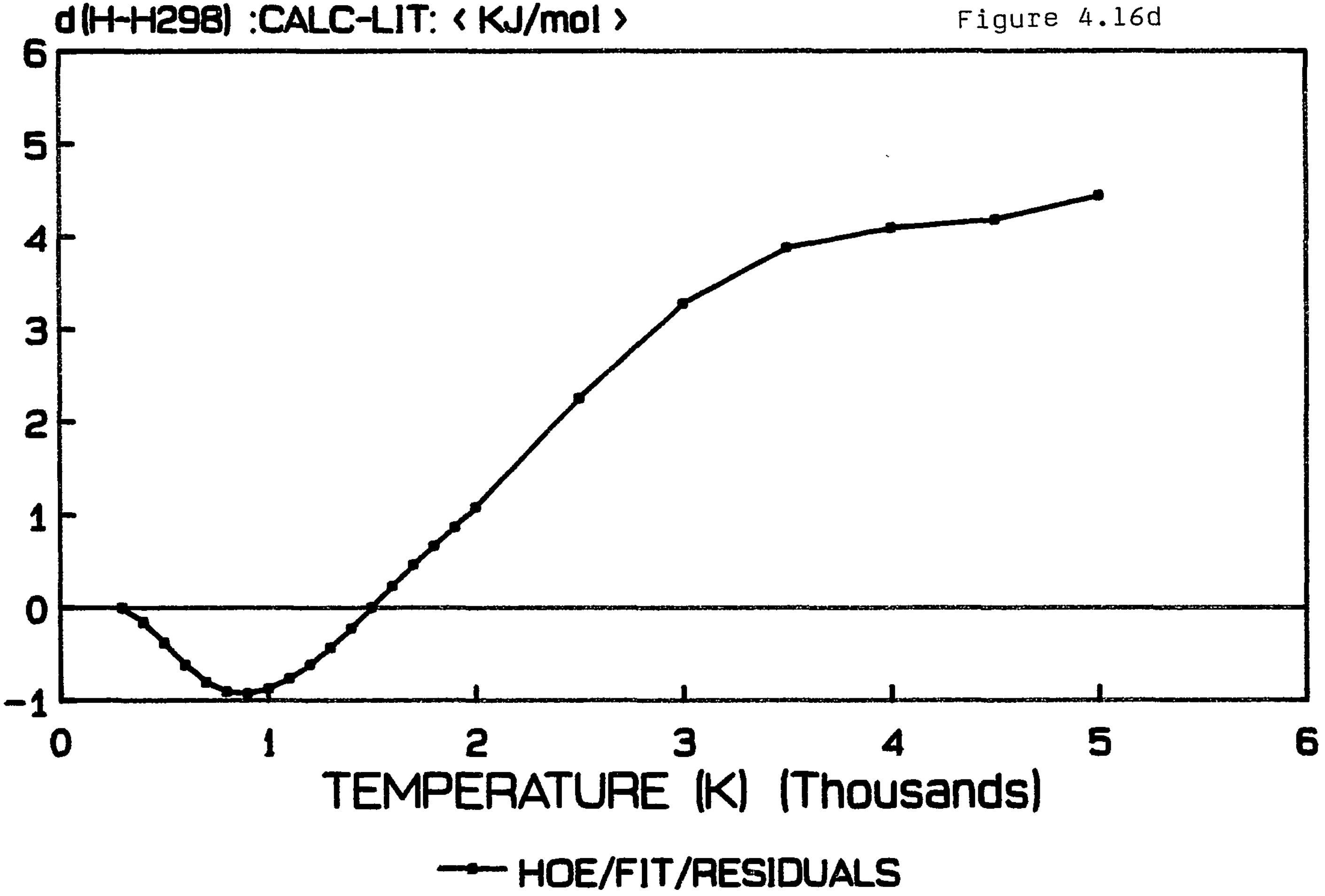

PHENYL: H-H298 : FIT OF HOE
(H-H298) < KJ/mol >
1400
1200
1000
800
600
400
200
0
0
1
2
3
4
5
6
TEMPERATURE (K) (Thousands)
● BURCAT
THERM/HOE/FIT

Figure 4.16c

PHENYL: H-H298 : FIT OF HOE:RESIDUALS

Figure 4.16d



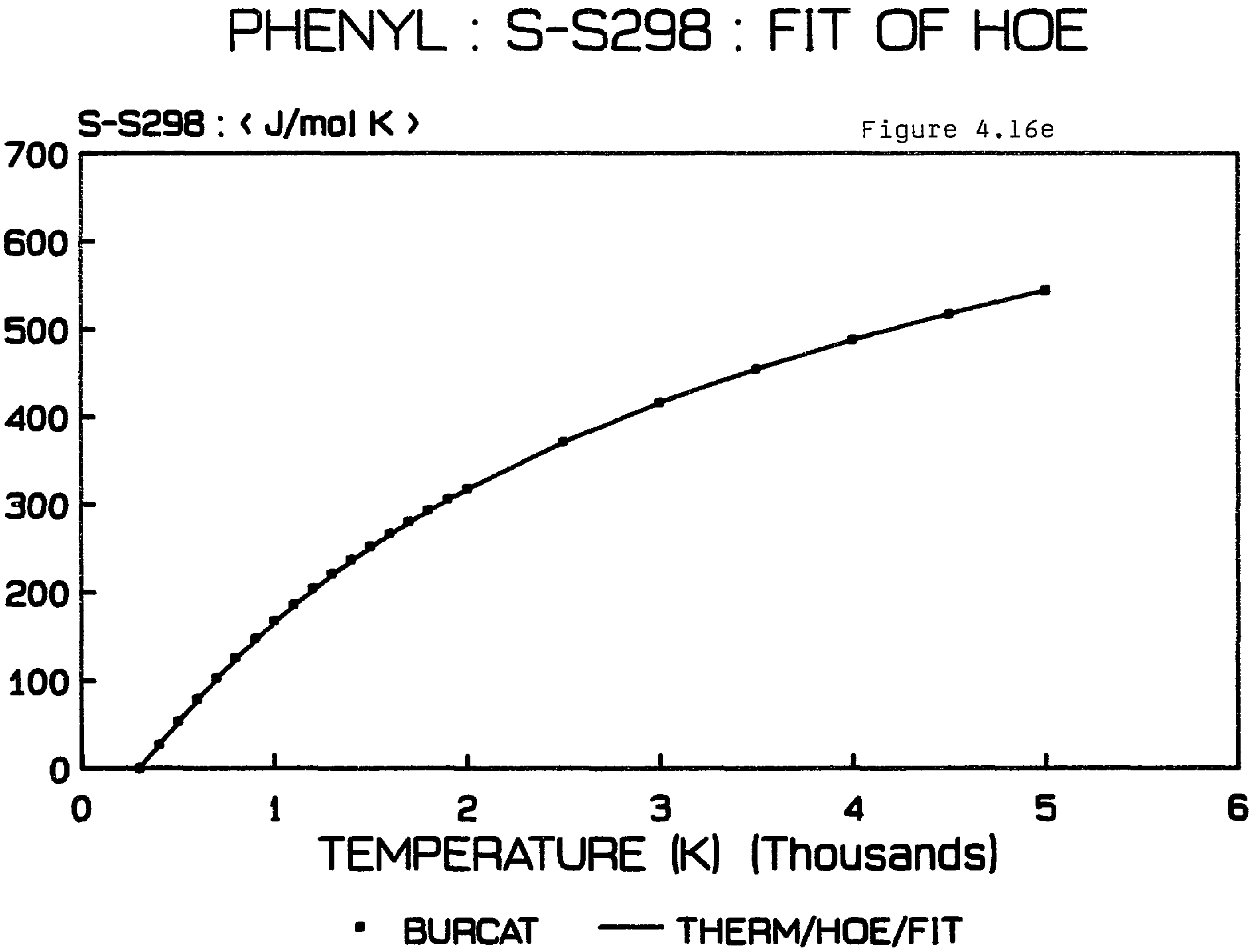

PHENYL : S-S298 : FIT OF HOE
S-S298 : < J/mol K >
700
600
500
400
300
200
100
0
0
1
2
3
4
5
6
TEMPERATURE (K) (Thousands)
• BURCAT
— THERM/HOE/FIT

Figure 4.16e

PHENYL: S-S298 : FIT OF HOE:RESIDUALS

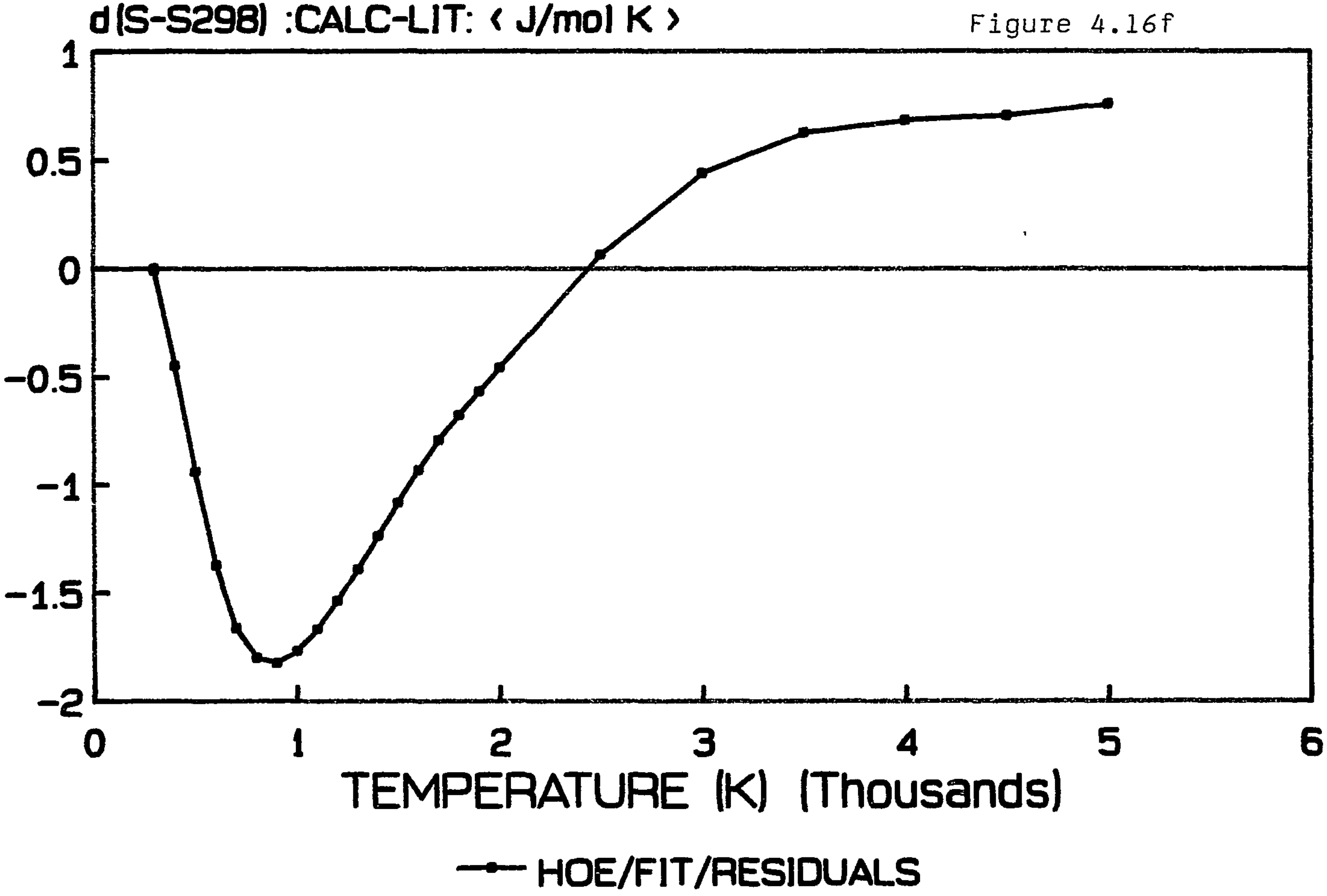
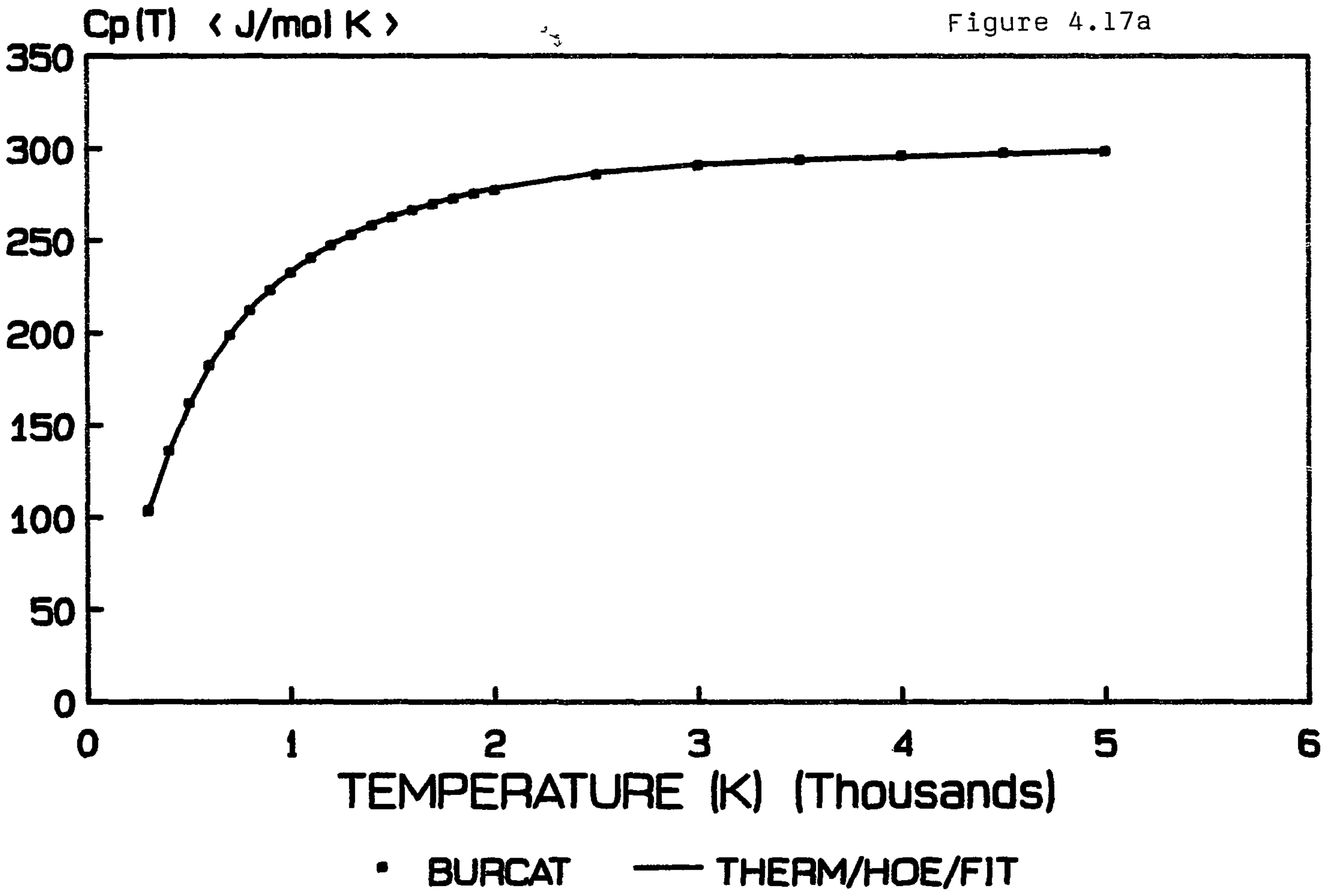


Figure 4.16f

PHENOL : Cp (T) : FIT OF HOE



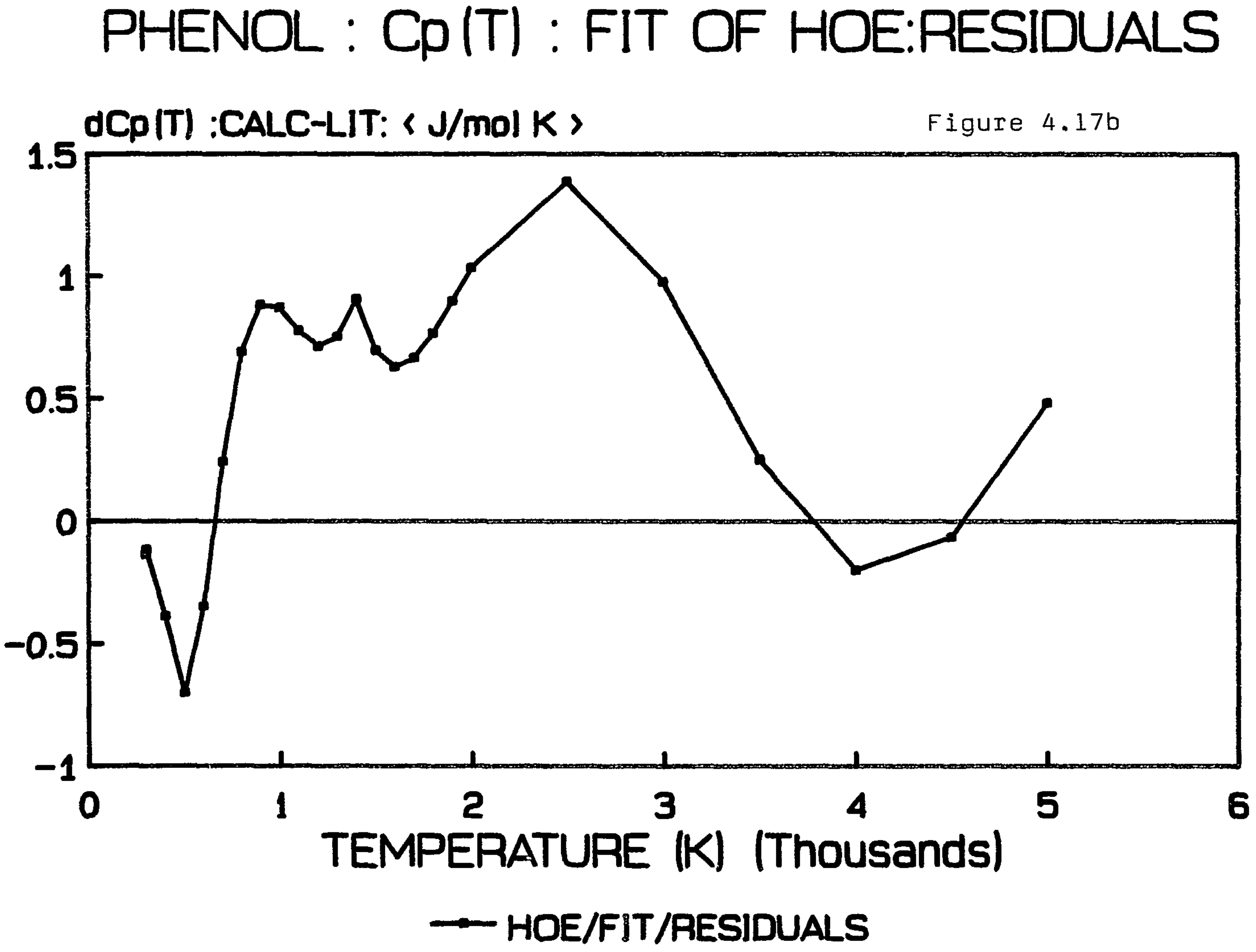
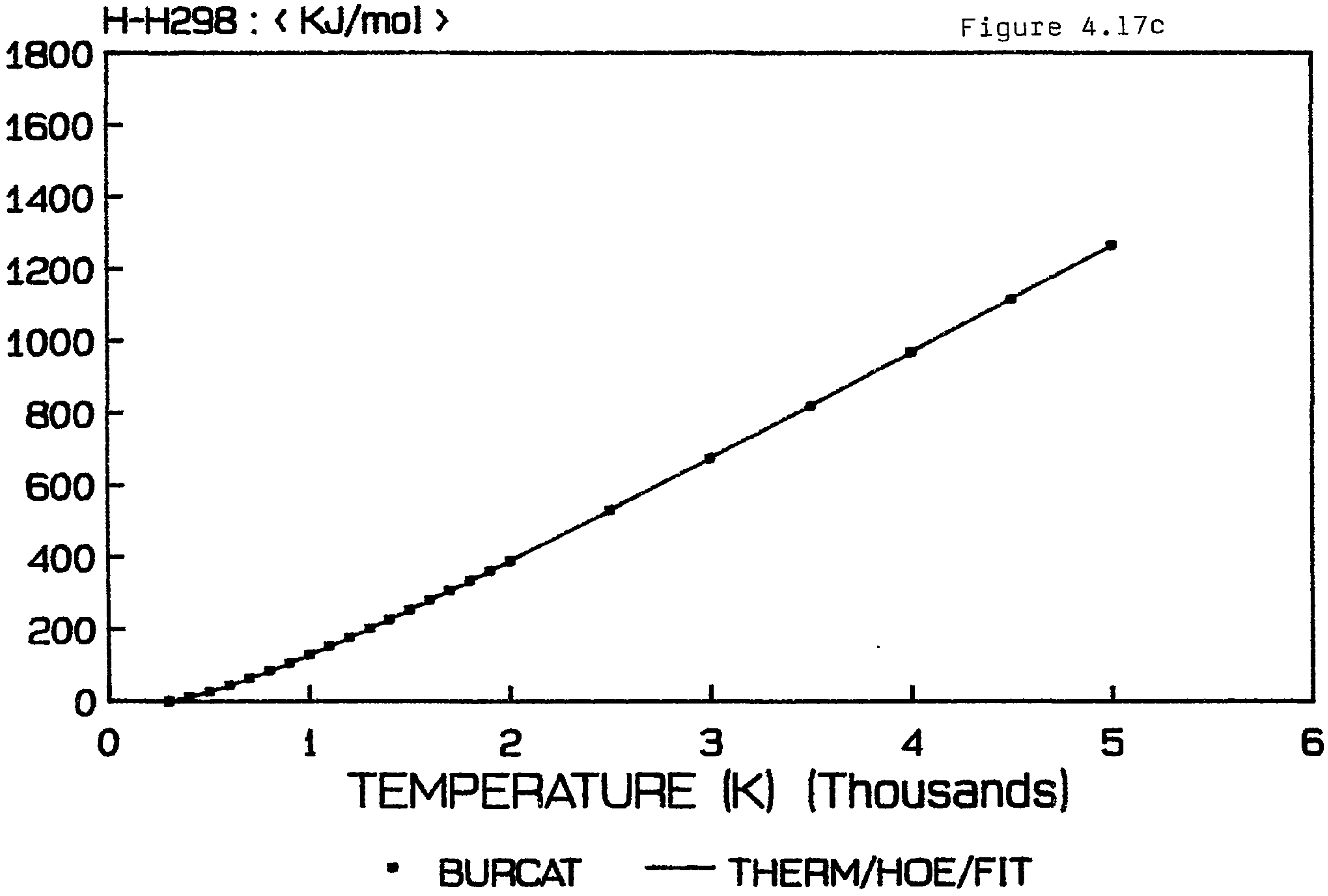

PHENOL : Cp(T) : FIT OF HOE:RESIDUALS
dCp(T) :CALC-LIT: < J/mol K >
1.5
1
0.5
0
-0.5
-1
0
1
2
3
4
5
6
TEMPERATURE (K) (Thousands)
HOE/FIT/RESIDUALS

Figure 4.17b

PHENOL : H-H298 : FIT OF HOE

Figure 4.17c



PHENOL: H-H298 : FIT OF HOE:RESIDUALS

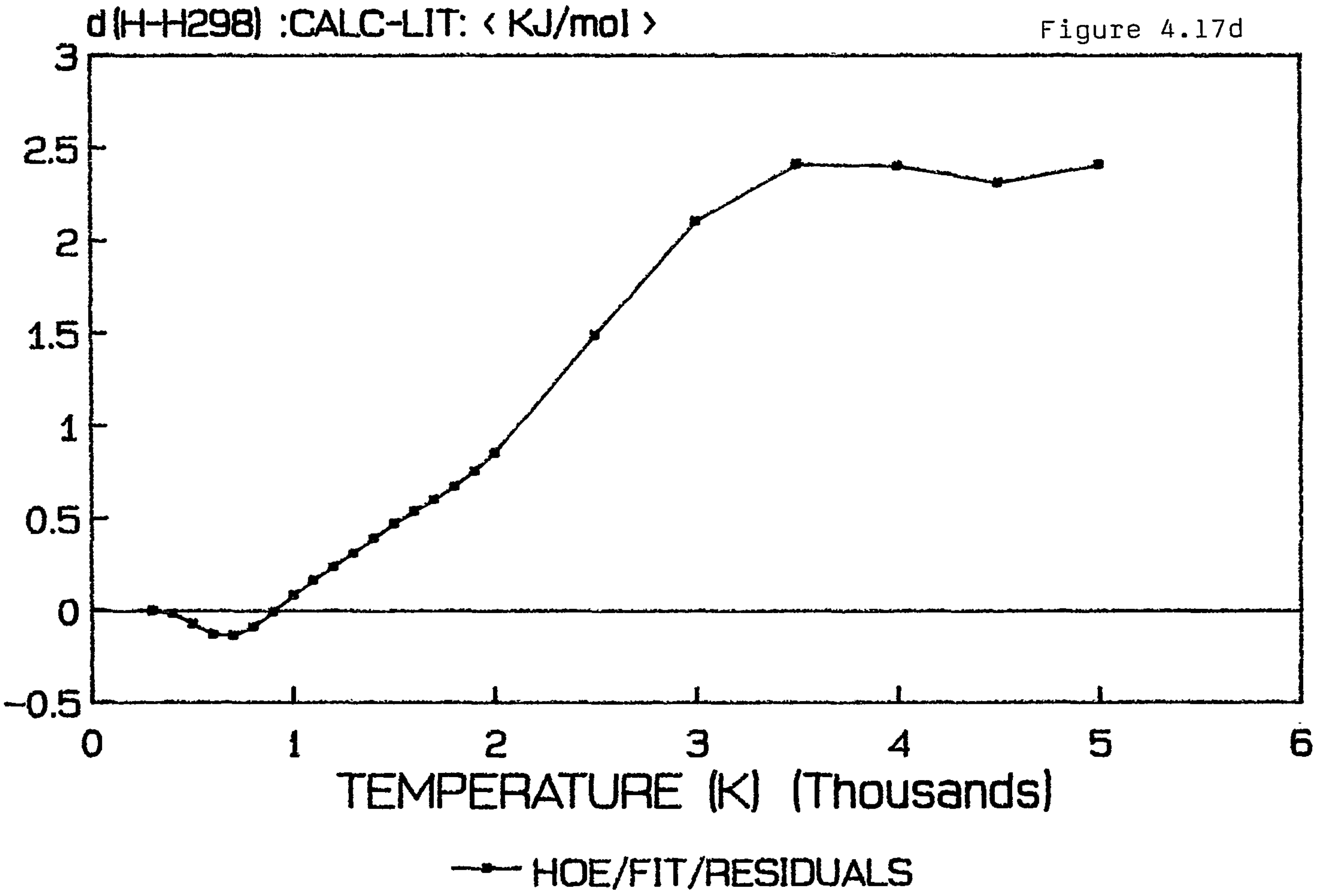


Figure 4.17d

PHENOL : S-S298 : FIT OF HOE

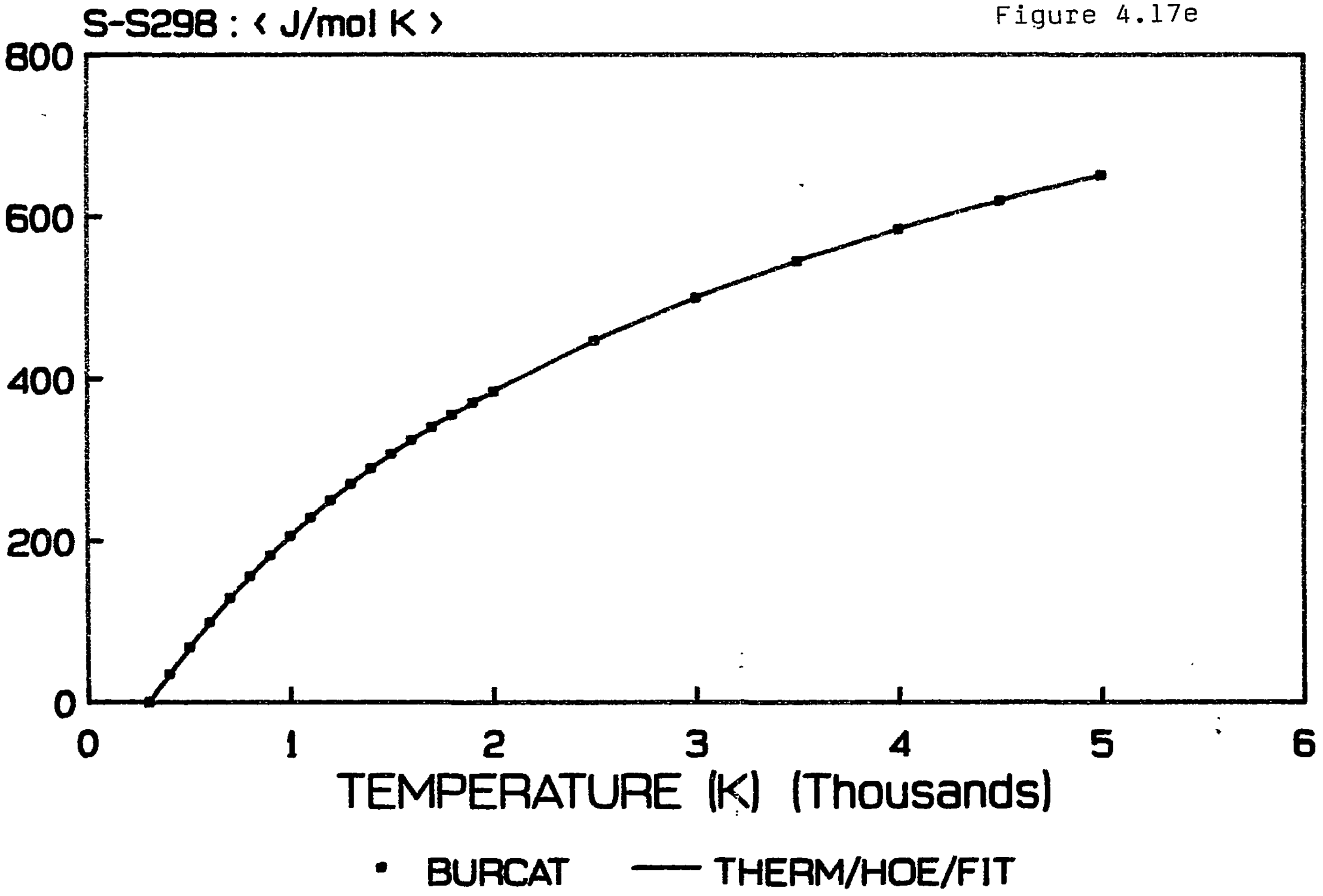


Figure 4.17e

4.17f

PHENOL: S-S298 : FIT OF HOE:RESIDUALS

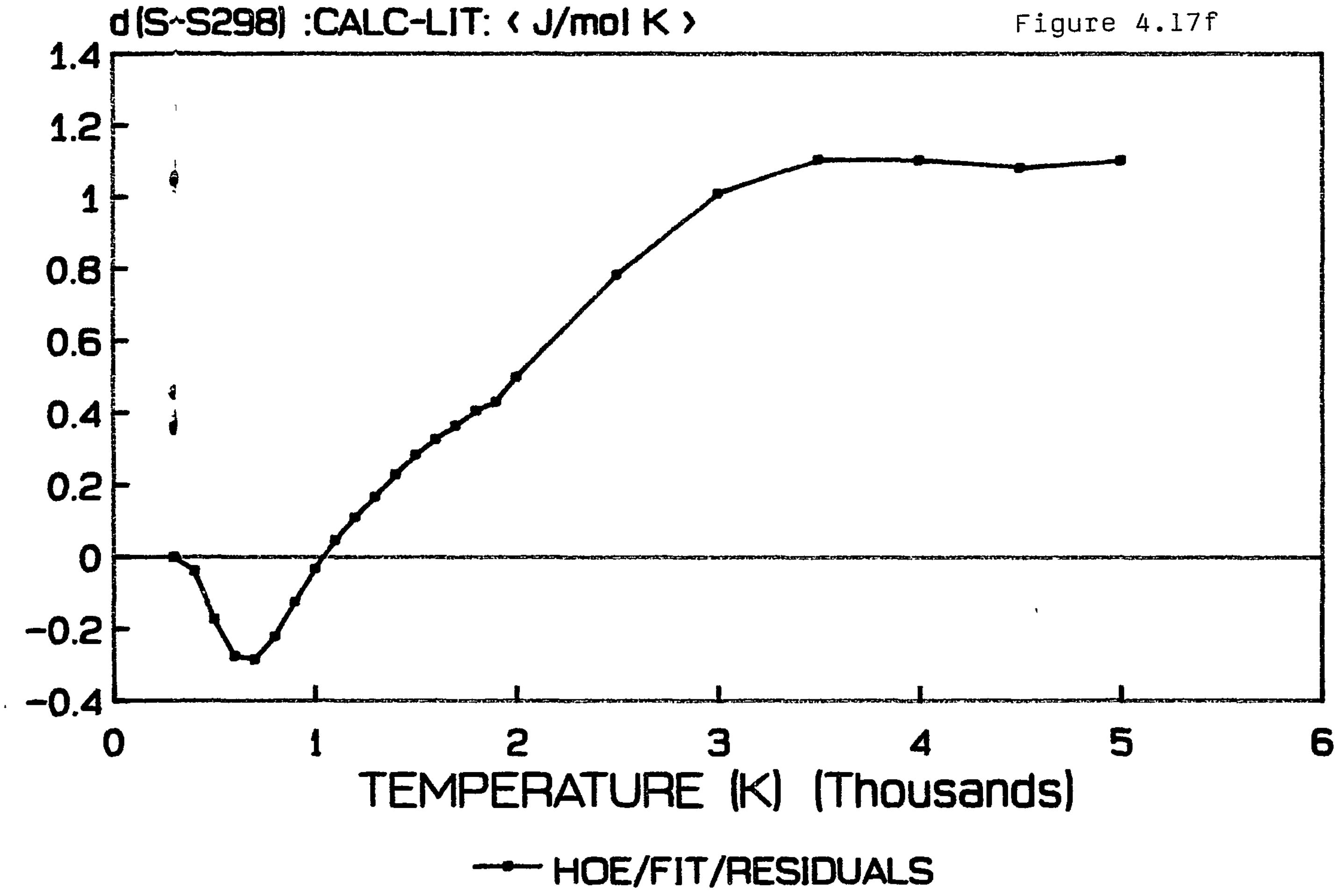


Figure 4.17f

4.18a

BIPHENYL : Cp(T) : FIT OF HOE

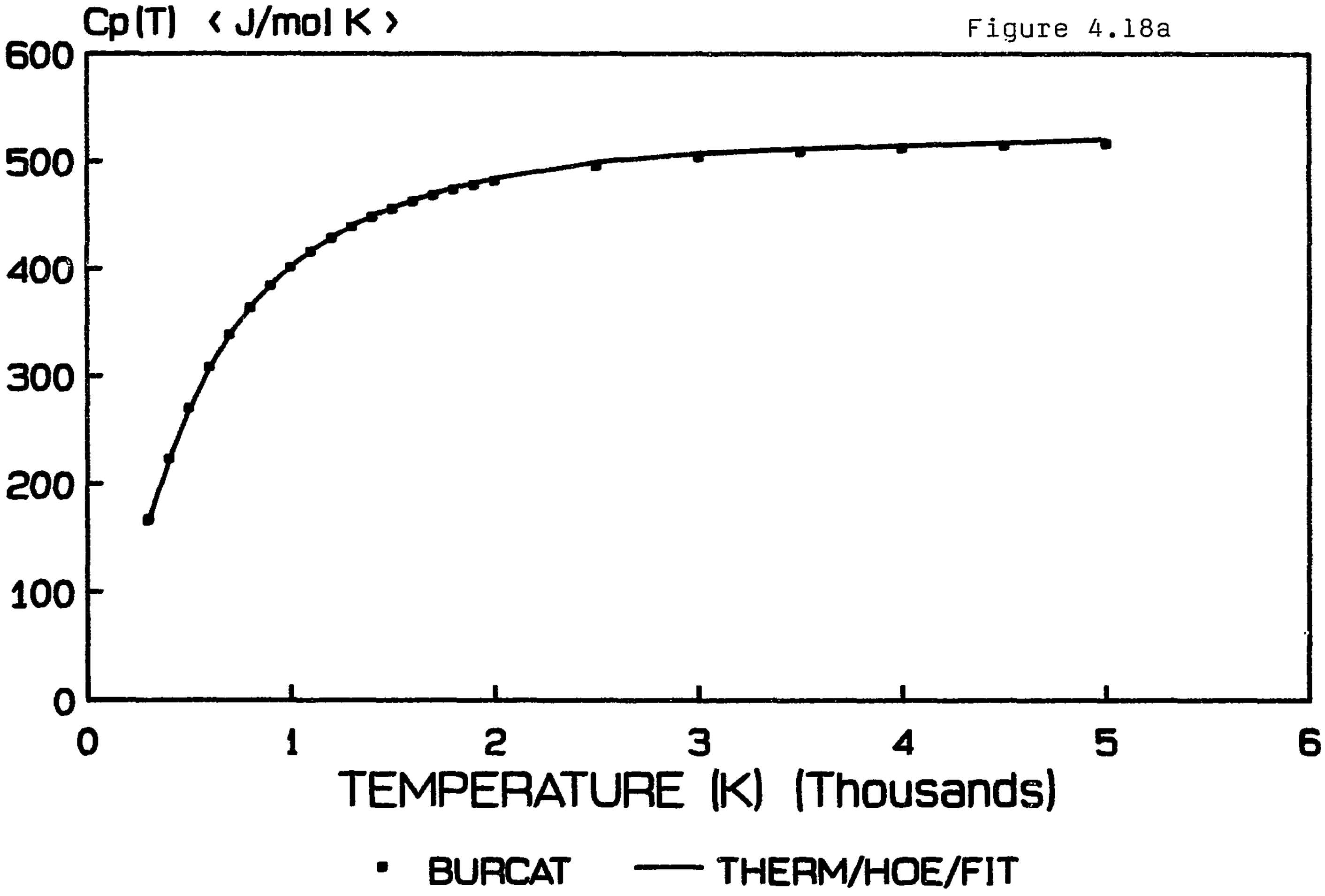


Figure 4.18a

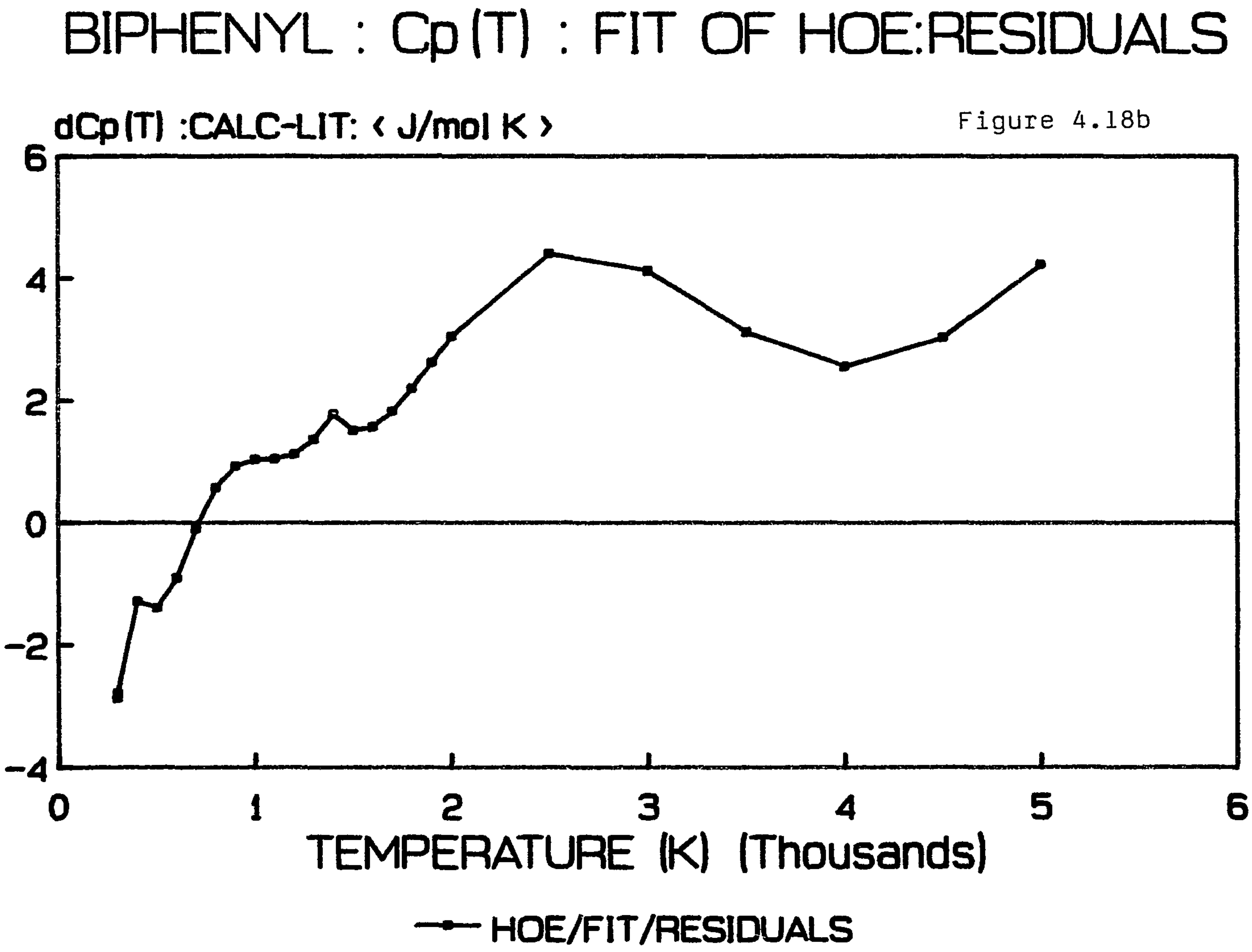
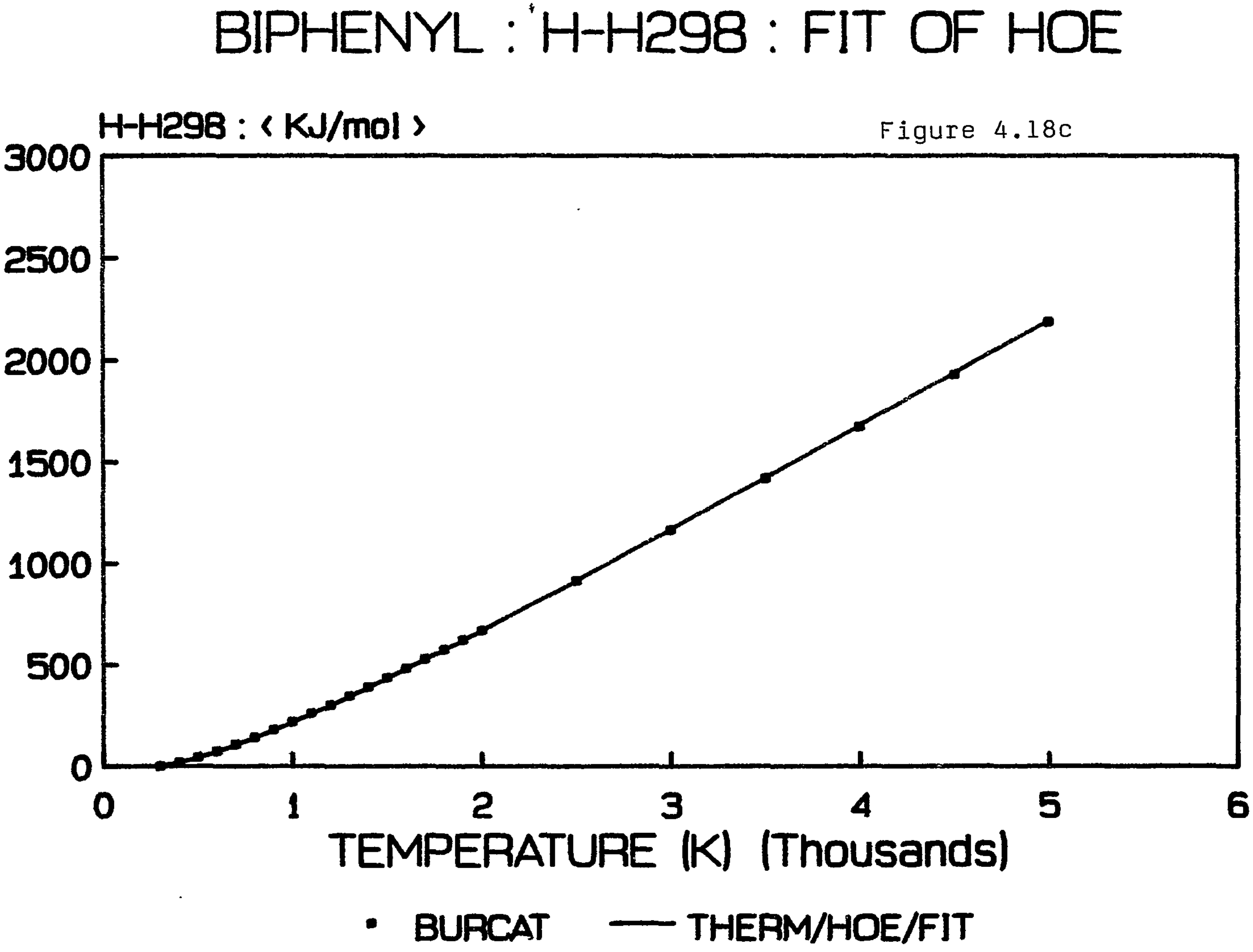

BIPHENYL : Cp(T) : FIT OF HOE:RESIDUALS
dCp(T) :CALC-LIT: ‹ J/mol K ›
6
4
2
0
−2
−4
0
1
2
3
4
5
6
TEMPERATURE (K) (Thousands)
HOE/FIT/RESIDUALS

Figure 4.18b


BIPHENYL : H-H298 : FIT OF HOE
H-H298 : < KJ/mol >
Figure 4.18c
3000
2500
2000
1500
1000
500
0
0
1
2
3
4
5
6
TEMPERATURE (K) (Thousands)
■ BURCAT
— THERM/HOE/FIT

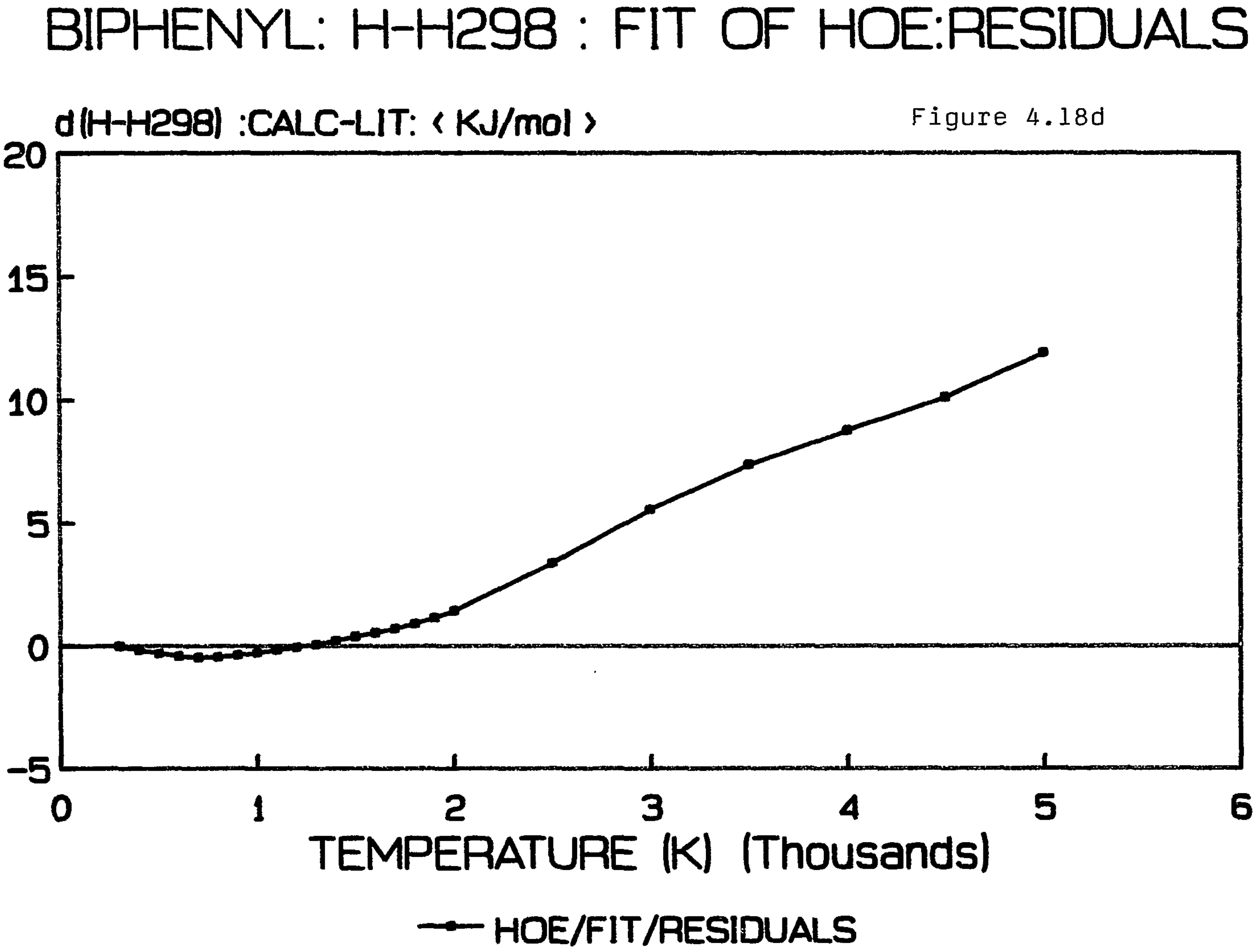

BIPHENYL: H-H298 : FIT OF HOE:RESIDUALS
d(H-H298) :CALC-LIT: < KJ/mol >
20
15
10
5
0
-5
0
1
2
3
4
5
6
TEMPERATURE (K) (Thousands)
HOE/FIT/RESIDUALS

Figure 4.18d

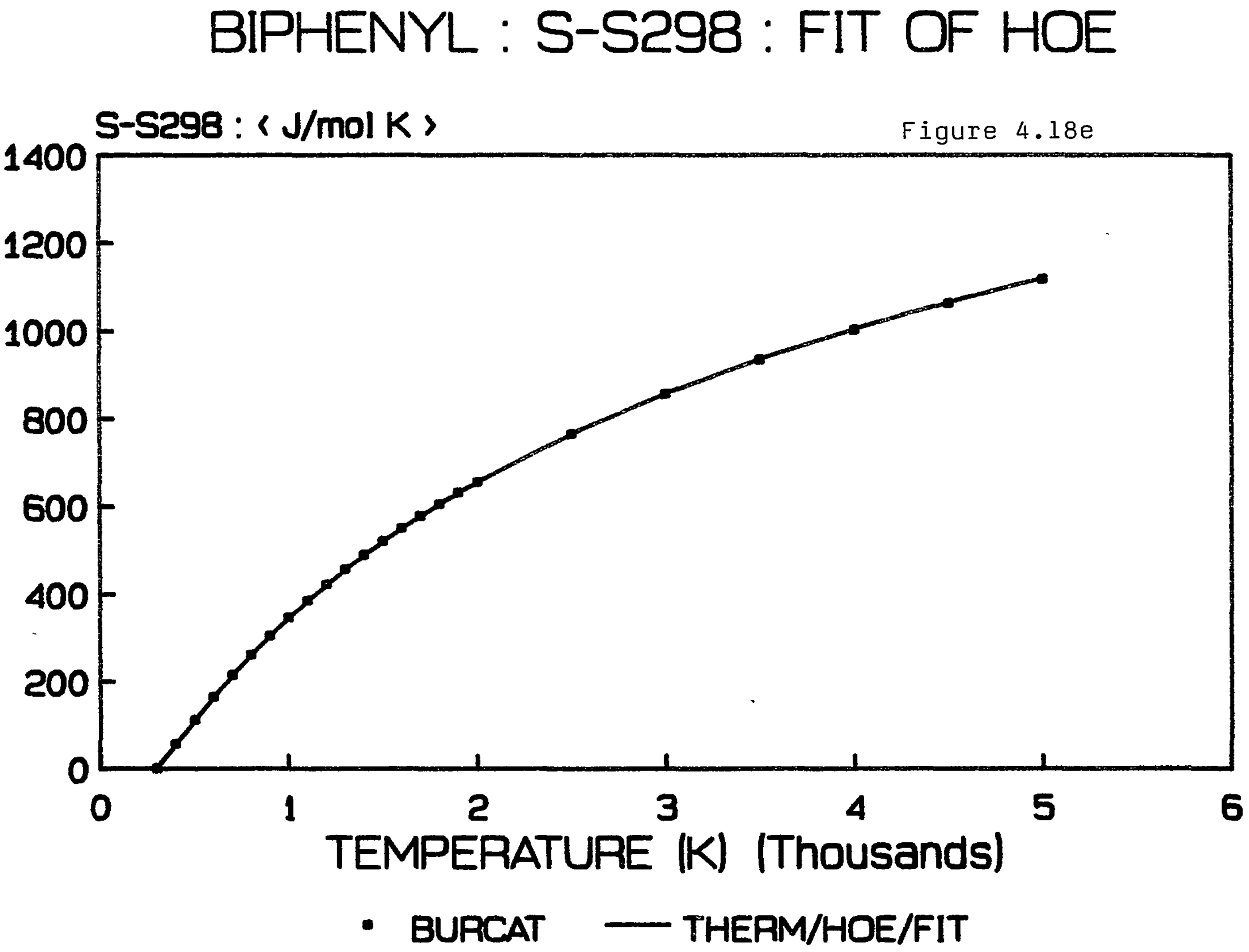

BIPHENYL : S-S298 : FIT OF HOE
S-S298 : ‹ J/mol K ›
1400
1200
1000
800
600
400
200
0
0
1
2
3
4
5
6
TEMPERATURE (K) (Thousands)
▪ BURCAT
—— THERM/HOE/FIT

Figure 4.18e

4.18f

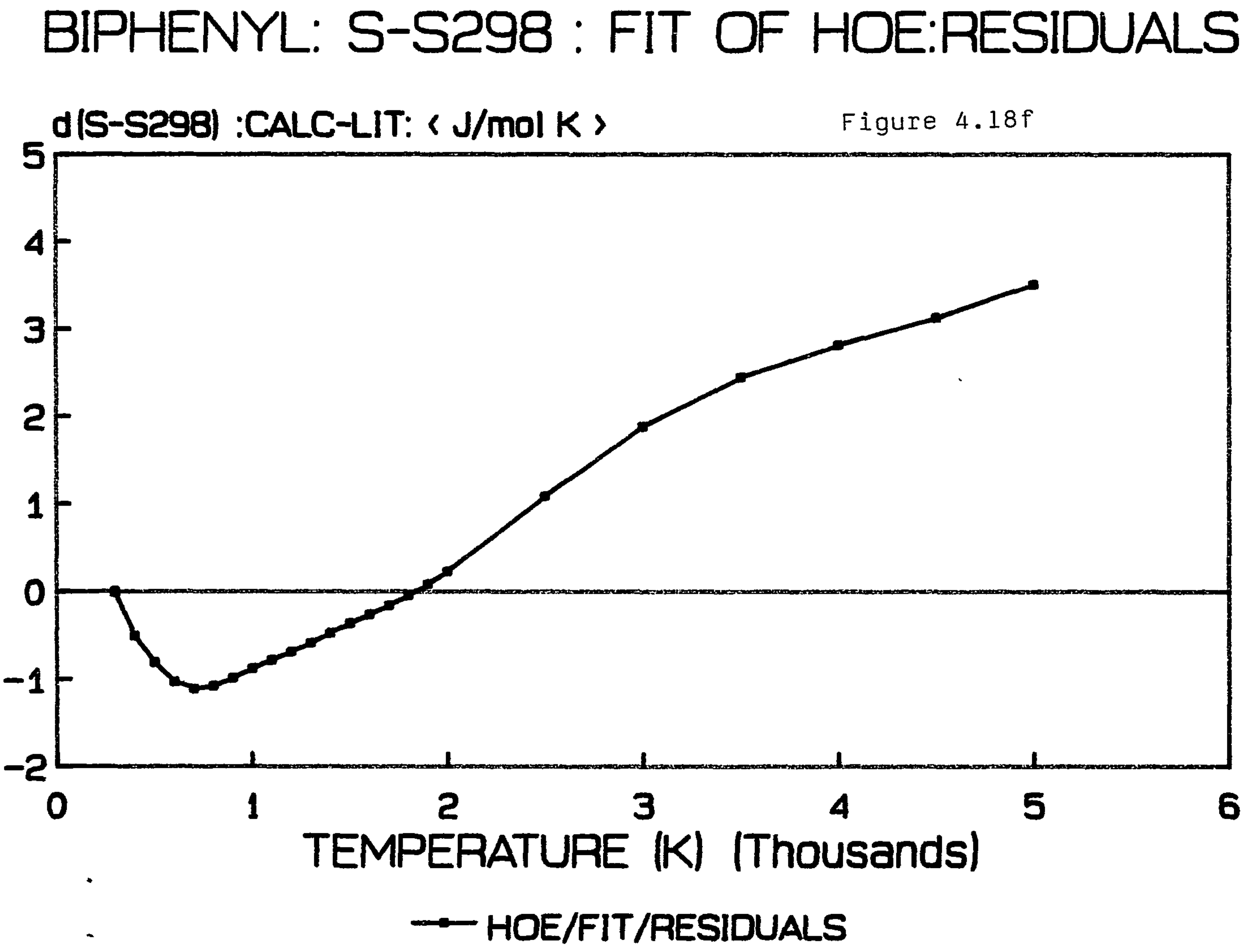


Figure 4.18f

CY13HD : ln(Qvib) : LIT VS THERM VIBRATIONAL PARTITION F'N

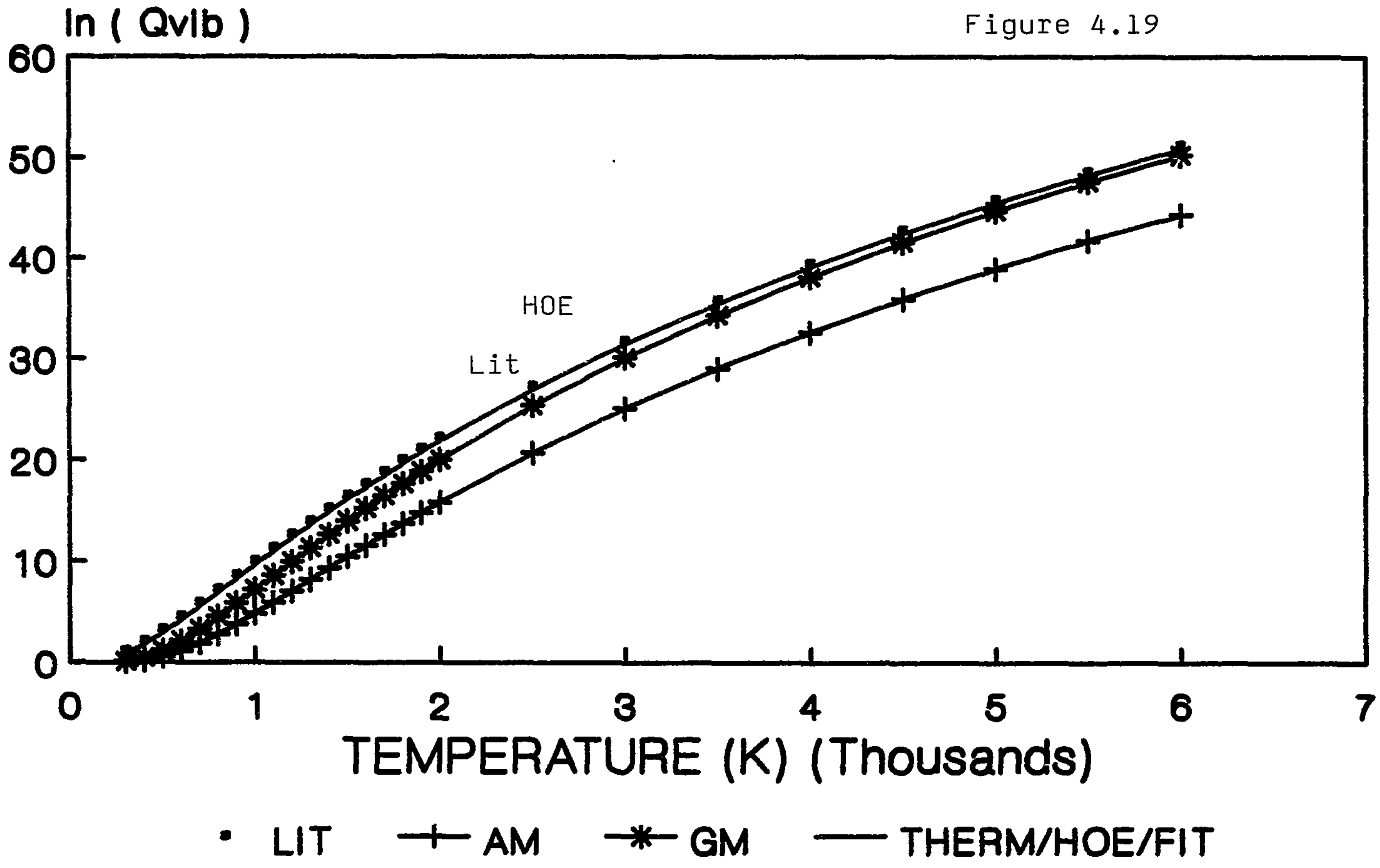
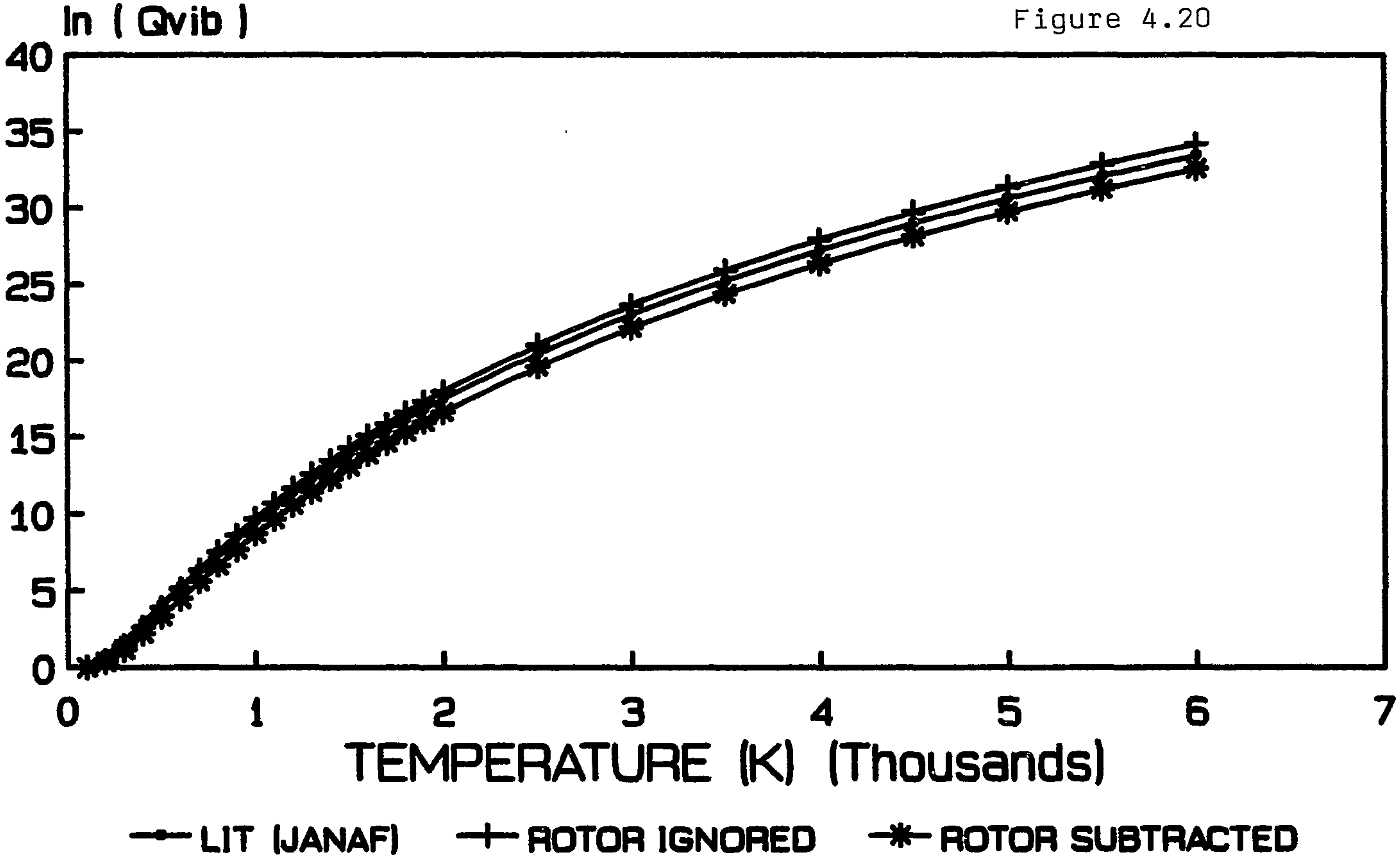


Figure 4.19

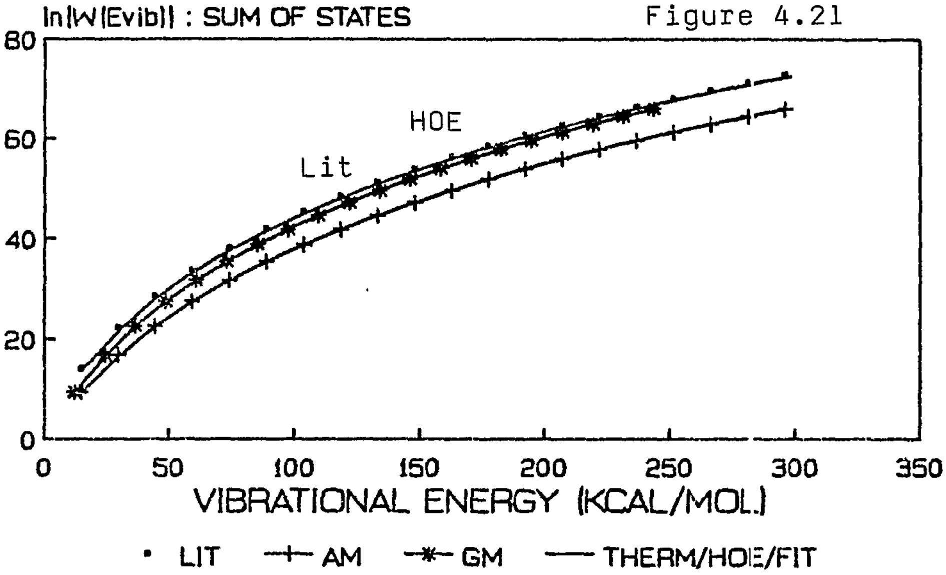
C2F6 : VIBRATIONAL PARTITION FUNCTION COMPARISON OF THERM ESTIMATE W/EXP

Figure 4.20



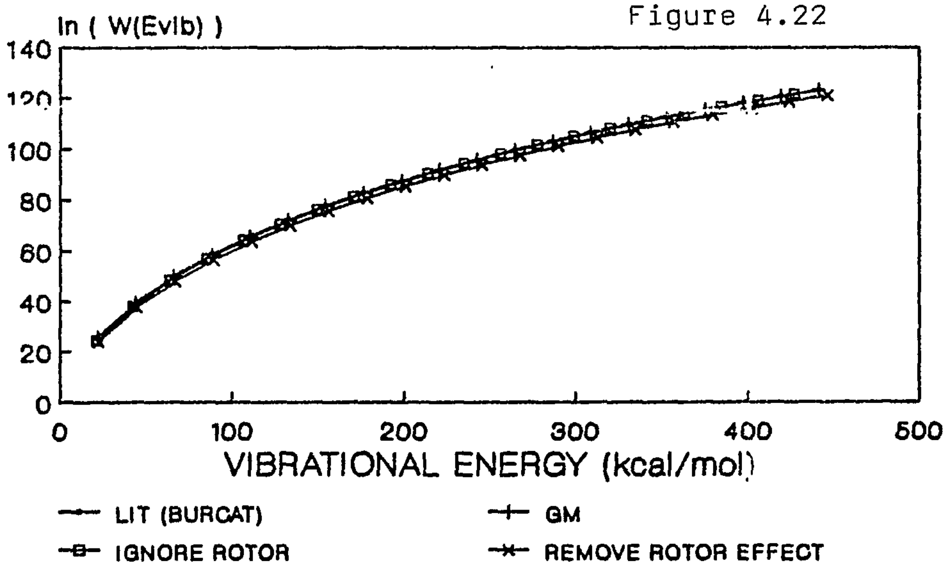
CHD13:SUM OF VIBRATIONAL QUANTUM STATES COMPARISON W/R : FREQUENCIES FROM CPFIT

Figure 4.21



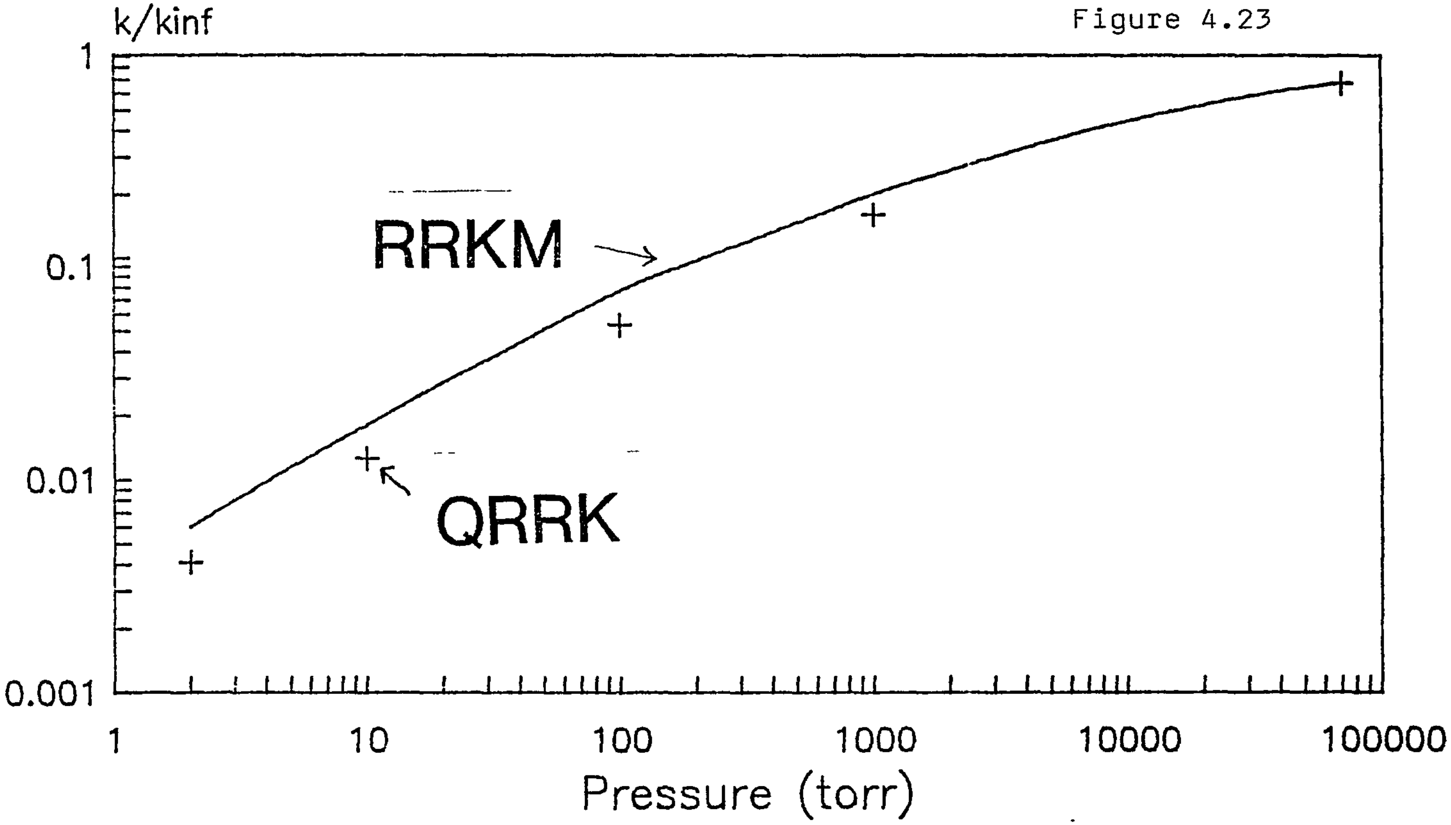
BIPHENYL : SUM OF VIBRATIONAL STATES WHITTEN-RABINOVICH APPROXIMATION

Figure 4.22



SF6 Dissociation
Comparison of QRRK w/ RRKM (Lyman)

Figure 4.23



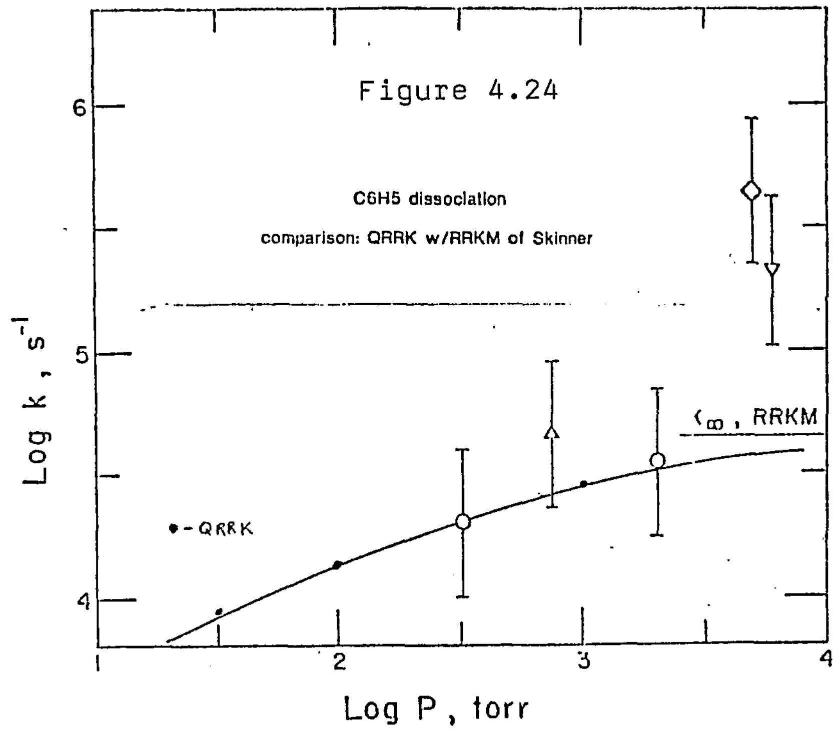


Figure 4.24

C6H5 dissociation

comparison: QRRK w/RRKM of Skinner

C6H6 dissociation
comparison: QRRK w/RRKM of Skinner

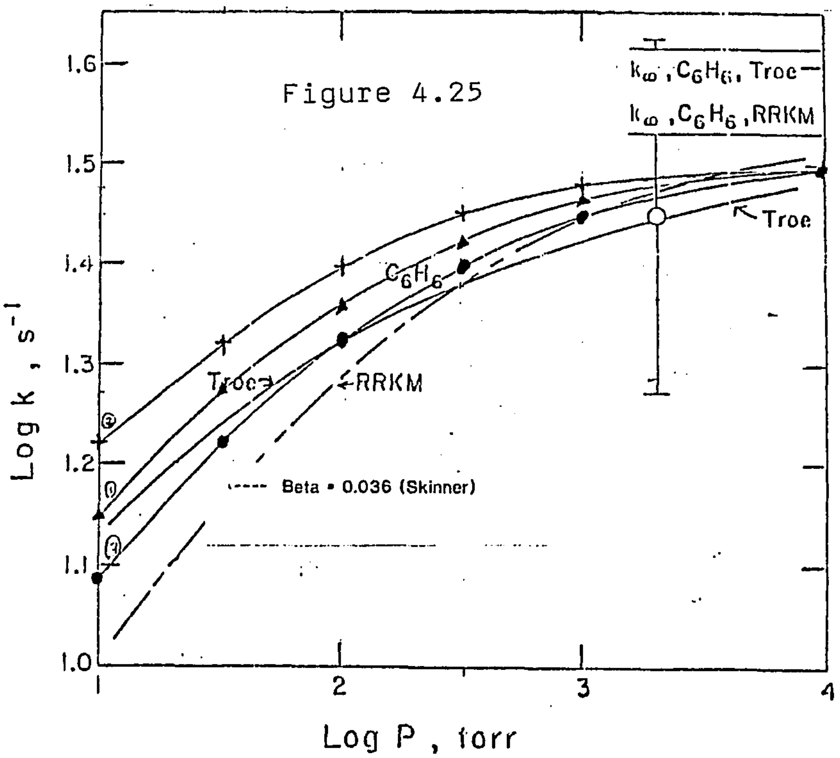


Figure 4.25

APPENDIX IV - C

FORTRAN SOURCE CODE FOR

THERMFEX

```
      PROGRAM THERMFIT
$LARGE
C**********************************************************************
C     THIS PROGRAM CREATES A NASA FORMAT THERMO DATABASE FILE
C
C            TWO SETS OF COEFFICIENTS ARE GENERATED AFTER FITTING CP DATA
C            USING A MODIFIED VERSION OF THE CPFIT PROGRAM
C
C               TMP DATA FILE (UNIT 45) IS USED TO PASS FILENAMES DIRECTLY
C               TO THIS PROCEDURE FROM THERM.  IF THE FILE "NASAGEN.TMP"
C               DOES NOT EXIST, THEN THE USER IS PROMPTED FOR INPUT/OUTPUT
C               FILENAMES.
C
C
C          INPUT FILE (UNIT 1) IS IN THERMLST FORMAT
C
C          OUTPUT FILE (UNIT 3) CONTAINS COEFFICIENTS IN NASA FORMAT
C          FOR USE WITH   CHEMKIN AND OTHER PACKAGES
C
C**********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
             DIMENSION X0(2),WX(2),S(2)
      DIMENSION CPL(30),TRA(3),A(5),B(6),C(14),F(14),FS(3,14),
     +TH(3),TL(3),TM(3),SSRS(3),PERMX(3)
      DIMENSION TFIT(30),CPFIT(30),PWORK(120),D(14),CONST(3)
      DIMENSION X(31,15,1),Y(31,25,1),B1(16,25,1),T(10),TEST(30),
     +CPEST(30)
             REAL*8 TMID
             REAL*8 DELPCT(20)
      CHARACTER*6 TITLE(5),HEADIN(21),REF3,XNAME
      CHARACTER*1 PHASE,CFLAG
      CHARACTER*2 IE(4)
      INTEGER NE(4)
      CHARACTER*4 REF(2),INAME,IFLAG
             CHARACTER*14 FUNITS,NUL
             CHARACTER*70 FILIN,FILOUT,COMMAND,DUMMY,S1
             LOGICAL EXST,OPNED
             EXTERNAL FX
             COMMON/DATA/CPEX1,T1,IT,NT,NS
             COMMON/CONST/CONST,NTERMS
             COMMON/TOL/TOL,TSTEP
             COMMON/TEMPFIT/TFIT,F
             COMMON/FLAG/IREPEAT,DELPCT,IFLG
      DATA NPTS/7/
      DATA IFLAG/'END'/
      DATA IC1,IC2,IC3,IC4/1,2,3,4/
             T(1)=300.0
             T(2)=400.0
             T(3)=500.0
             T(4)=600.0
             T(5)=800.0
             T(6)=1000.0
             T(7)=1500.0
             TEST(1)=300.
             TEST(2)=350.
             TEST(3)=400.
             TEST(4)=425.
             TEST(5)=450.
             TEST(6)=500.
             TEST(7)=550.
             TEST(8)=600.
             TEST(9)=650.
             TEST(10)=700.
             T0=298.15
             TLO=300.0
             THI=5000.0
             TMID=1500.0
             TOL=1.0E-4
             TSTEP=0.0
14           FORMAT(A14)

             COMMAND=' '
             FILIN=' '
             FILOUT=' '
             INQUIRE(FILE='THERMFIT.TMP',EXIST=EXST)
             IF(EXST)THEN
                          OPEN(45,FILE='THERMFIT.TMP',STATUS='OLD')
                          READ(45,3)COMMAND
                          READ(45,3)FILIN
                          READ(45,3)FILOUT
                          CLOSE(45,STATUS='DELETE')
                          GOTO 123
             ENDIF
        WRITE(*,*)'ENTER FILENAME FOR INPUT {.LST}'
             READ(*,3)FILIN
3            FORMAT(A70)
             CALL FNAME(FILIN,'LST')
             WRITE(*,*)'ENTER FILENAME FOR OUTPUT {.DAT}'
             READ(*,3)FILOUT
             CALL FNAME(FILOUT,'DAT')
123          CONTINUE
             INQUIRE(FILE=FILIN,EXIST=EXST)
             IF(EXST)GOTO 324
             STOP 2
324          OPEN(1,FILE=FILIN,STATUS='OLD')
             OPEN(3,FILE=FILOUT,STATUS='UNKNOWN')
             IF(COMMAND.EQ.' ')THEN
                          WRITE(*,*)' '
                          WRITE(*,*)' '
                          WRITE(*,*)'   GENERATING NASA FORMAT DATABASE FROM'
                          WRITE(*,*)'    ',FILIN
                          WRITE(*,*)' '
                          WRITE(*,*)'            please wait'
             ENDIF
      WRITE(3,99)
      WRITE(3,98)
   99 FORMAT('THERMO ',73(' '))
   98 FORMAT('   300.000  1500.000  5000.000 ',49(' '))
    1 ICOUNT=1
      READ(1,3)DUMMY
             CALL DBLNK(DUMMY,LEN,0)
             CONST(1)=1.0
             FUNITS='KCAL'
             IF(DUMMY(:6).EQ.'UNITS:')THEN
                          IF(DUMMY(7:8).EQ.'KJ')THEN
                                       FUNITS='KJ'
                                       CONST(1)=4.184
                          ENDIF
                          READ(1,3)DUMMY
             ENDIF
   95 FORMAT(5A6/,21A6)
             read(1,14)nul
20           CONTINUE
             DO 203 IK1=1,20
203          DELPCT(IK1)=0.0
             YERRS=1.0E+7
             IREPEAT=0
             IFLG=0
             IPOLY=0
             IPOLYC=0
             CFLAG=' '
             THI=5000.
             READ(1,'(1X,A4)',END=900)INAME
             IF(INAME.EQ.' ')THEN
                          GOTO 20
             ELSE
                          BACKSPACE(1)
             ENDIF
             IF(FUNITS.EQ.'KCAL')THEN
      READ(1,170,END=900)CFLAG,INAME,XNAME,HF298,S298,(CPL(I),I=1,7),
     1 (REF(I),I=1,2),REF3, (IE(I),NE(I),I=1,4),PHASE,NROT
  170 FORMAT(A1,A4,A6,F7.2,2X,F7.2,3X,7(F6.2,2X),2A4,A6,
     1 3X,4(A2,I3,1X),A1,1X,I1)
             ELSE IF(FUNITS.EQ.'KJ')THEN
      READ(1,176,END=900)CFLAG,INAME,XNAME,HF298,S298,(CPL(I),I=1,7),
     1 (REF(I),I=1,2),REF3, (IE(I),NE(I),I=1,4),PHASE,NROT
176   FORMAT(A1,A4,A6,F8.2,1X,F8.2,2X,7(F7.2,1X),2A4,A6,3X,4(A2,I3,
     +  1X),A1,1X,I1)
             ENDIF
      IF(INAME(:3).EQ.'END') THEN
                   WRITE(3,1012)
 1012         FORMAT('END',77(' '))
 1011         FORMAT(A4)
                   STOP 'END OF THERMFIT'
      ENDIF
             IF(CFLAG.EQ.' ')THEN
                          WRITE(*,'(1X,A4,A6)')INAME,XNAME
             ELSE
                          WRITE(*,'(1X,A4,A6,2X,A1)')INAME,XNAME,CFLAG
             ENDIF
C---CONVERTING HF298 TO CAL/MOLE OR J/MOLE
```

```
             IF(FUNITS.EQ.'KCAL')THEN
                   HF298=HF298*1000.
             ELSEIF(FUNITS.EQ.'KJ')THEN
                          HF298=HF298*1000.0/4.184
                          S298=S298/4.184
             ENDIF
      TRA(1)=300.
             NS=0
             NATOM=0
             ICARBON=0
             NCARBON=0
             DO 765 IIJ=1,4
             IF(IE(IIJ).EQ.'C '.OR.IE(IIJ).EQ.' C')THEN
                          ICARBON=1
                          NCARBON=NE(IIJ)
             ENDIF
765          NATOM=NATOM+NE(IIJ)
986          CONTINUE
             DIF1=DABS(CPL(1)-CPL(3))
             DIF2=DABS(CPL(1)-CPL(6))
             DIFF=MAX(DIF1,DIF2)
             IF(FUNITS.EQ.'KJ')DIFF=DIFF/4.184
             IF(NATOM.NE.1.AND.DIFF.GT.0.01)THEN
                          IF((ICARBON.EQ.1.AND.(NATOM-NCARBON).EQ.2).OR.NATOM.EQ.2
     +)THEN
                                       NS=(3*NATOM)-5
                                       CONST(3)=1.0
                                       CPINFIN=1.987*(FLOAT(3*NATOM)-1.5)*CONST(1)
                          ELSE
                                       NS=(3*NATOM)-6
                                       CONST(3)=0.0
                          CPINFIN=1.987*(FLOAT(3*NATOM)-(2.+(FLOAT(NROT)/4.0)))
     +*CONST(1)
                          ENDIF
                          IF(CPL(7).EQ.0.0)THEN
                                       CPL(7)=CPL(6)+(0.37*(CPINFIN-CPL(6)))
                          ENDIF
             CPL(8)=CPL(7)+(0.37*(CPINFIN-CPL(7)))
             CPL(9)=0.99999*CPINFIN
             T(1)=300.0
             T(2)=400.0
             T(3)=500.0
             T(4)=600.0
             T(5)=800.0
             T(6)=1000.0
             T(7)=1500.0
             T(8)=2000.0
             T(9)=20000.0
             T(10)=0.0
             NT=9
                          IF(CFLAG.EQ.'*')THEN
                                       IFLG=2
                                       NT=8
                                       GOTO 1994
                          ENDIF
********************** POLYNOMIAL SMOOTHING PROCEDURE *********
1994                      CONTINUE
                          IF(IFLG.NE.0)THEN
                                       IFLG=0
*                                      POLYNOMIAL SMOOTHING DID NOT HELP OR USER SPECIFIES
*                       POLYNOMIAL FIT ONLY THEN
*                                      DEFAULT TO SINGLE REPEATED POLYNOMIAL
*                                      OVER TRANGE 300 - 2000 K
                                       IF(CPL(7).EQ.0.)THEN
                                                   CPL(7)=CPL(6)+(0.37*(CPINFIN-CPL(6)))
                                       ENDIF
                                       CPL(8)=CPL(7)+(0.37*(CPINFIN-CPL(7)))
                                       T(8)=2000.
                                       DO 61678 IK1=1,8
                                       DO 61677 IK2=1,5
61677                                  X(IK1,IK2,1)=T(IK1)**(IK2-1)
                                       CPFIT(IK1)=CPL(IK1)/1.987
                                       Y(IK1,1,1)=CPFIT(IK1)
61678                                  CONTINUE
                                       MDEG=4
                                       NPTS=8
                                       CALL LINLSQ(X,Y,B1,4,NPTS,1,1,SSR,AVGERR)
                                       DO 61679 IK1=1,5
61679                                  C(IK1)=B1(IK1,1,1)
                                       DO 61681 IK1=1,5
                                       F(IK1)=C(IK1)
                                       F(IK1+7)=C(IK1)
61681                                  CONTINUE

                                       C6=F6(0,(HF298/1.987),T0,F)
                   C7=(S298/1.987)-F(1)*DLOG(T0)-F(2)*T0-F(3)/2.*T0**2
     1                    -F(4)/3.*T0**3-F(5)/4.*T0**4
                                       F(6)=C6
                                       F(13)=C6
                                       F(7)=C7
                                       F(14)=C7
                                       TMID=1000.
                                       THI=2000.0
                                       ICOUNT=0
6545                                   DCPDT=DCPT(0,THI,F)
                                       COMPARE=((ABS(DCPDT)*500.)/CPT(0,THI,F))*100.
                                       IF(DCPDT.LT.0..AND.COMPARE.GT.10.)THEN
                                                   ICOUNT=ICOUNT+1
                                                   THI=THI-50.
                                                   IF(ICOUNT.LE.100)GOTO 6545
                                       ENDIF
                                       GOTO 12987
                          ENDIF
*******************************************************************
             ENDIF
73590        CONTINUE
             DO 431 JK=1,2
             IF(NATOM.NE.1.AND.DIFF.GT.0.01)THEN
```

```
              IF(IPOLY.EQ.1)THEN
                            IF(JK.EQ.1)THEN
                            TK=253.3333
                            DO 1290 I8=1,15
                                          TK=TK+46.6667
                                          TFIT(I8)=TK
                                          TFIT(I8+15)=1000.+(71.4236*FLOAT(I8-1))
1290                        CONTINUE
*             set lower temperature below room temp so that 298 is not
*        outside the range of the regression for Cp
                            TFIT(1)=273.15
                            ELSE
                            TK=928.5764
                            DO 1291 I8=1,15
                                          TFIT(I8)=1000.+(71.4236*FLOAT(I8-1))
                                          TFIT(I8+15)=2000.+(266.667*FLOAT(I8))
1291                        CONTINUE
                            ENDIF
              ELSEIF(IPOLY.EQ.0)THEN
                            IF(JK.EQ.1)THEN
                            TK=236.6667
                            DO 21290 I8=1,15
                                          TK=TK+63.3333
                                          TFIT(I8)=TK
                                          TFIT(I8+15)=1250.+(35.714*FLOAT(I8-1))
21290                       CONTINUE
*             set lower temperature below room temp so that 298 is not
*        outside the range of the regression for Cp
                            TFIT(1)=273.15
                            ELSE
                            DO 21291 I8=1,15
                                          TFIT(I8)=1250.+(35.714*FLOAT(I8-1))
                                          TFIT(I8+15)=1700.+(283.333*FLOAT(I8))
21291                       CONTINUE
                            ENDIF
              ELSEIF(IPOLY.EQ.3)THEN
                            IF(JK.EQ.1)THEN
                            TK=263.333
                            DO 22290 I8=1,15
                                          TK=TK+36.667
                                          TFIT(I8)=TK
                                          TFIT(I8+15)=850.+(35.714*FLOAT(I8-1))
22290                       CONTINUE
*             set lower temperature below room temp so that 298 is not
*        outside the range of the regression for Cp
                            TFIT(1)=273.15
                            ELSE
                            DO 22291 I8=1,15
                                          TFIT(I8)=850.+(35.714*FLOAT(I8-1))
                                          TFIT(I8+15)=1350.+(310.0*FLOAT(I8))
22291                       CONTINUE
                            ENDIF
              ELSEIF(IPOLY.EQ.2)THEN
                            IF(JK.EQ.1)THEN
```

```
                          TK=263.333
                          DO 23290 I8=1,15
                                       TK=TK+36.667
                                       TFIT(I8)=TK
                                       TFIT(I8+15)=850.+(21.429*FLOAT(I8-1))
23290                     CONTINUE
*            set lower temperature below room temp so that 298 is not
*        outside the range of the regression for Cp
                          TFIT(1)=273.15
                          ELSE
                          DO 23291 I8=1,15
                                       TFIT(I8)=850.+(21.429*FLOAT(I8-1))
                                       TFIT(I8+15)=1150.+(323.333*FLOAT(I8))
23291                     CONTINUE
                          ENDIF
             ENDIF
*   use linearized exponential function to fit cp data
*    this replaces harmonic oscillator expression.
             call cpexpoft(T,CPL,NT,TFIT,CPFIT,30,CPINFIN)
      TRA(1)=300.
      TRA(2)=1000.
      DO 30 I=1,30
      DO 31 I11=1,5
   31 X(I,I11,1)=TFIT(I)**(I11-1)
             TIN=TFIT(I)
      Y(I,1,1)=CPFIT(I)/(1.987*CONST(1))
   30 CONTINUE
      MDEG=4
      RSQ=100.
      IER=0
             NPTS=30
                   CALL LINLSQ(X,Y,B1,4,NPTS,1,1,SSR,AVGERR)
                   DO 32 I=1,5
   32               C(I)=B1(I,1,1)

      ELSE
*                         ATOM:  USE DIFFERENT METHOD FOR Cp
C    BYPASS FITTING ROUTINE IF CP <> f(T)
                          SUMCP=0.0
                          DATPNT=0.0
                          DO 1209 IJK=1,7
                          IF(CPL(IJK).NE.0.0)THEN
                                       SUMCP=SUMCP+CPL(IJK)
                                       DATPNT=DATPNT+1.0
                          ENDIF
1209                      CONTINUE
             C(1)=SUMCP/(1.98717*DATPNT)
*                         C(1)=CPL(1)/1.98717
                          IF(FUNITS.EQ.'KJ')C(1)=C(1)/4.184
                        DO 15 I=2,5
                        C(I)=0.0
   15               CONTINUE
                          TMID=1000.
      ENDIF
```

```
              IF(JK.EQ.1)THEN
                                 DO 75 I=1,5
                                 F(I)=C(I)
   75                       CONTINUE
              ELSEIF(JK.EQ.2)THEN
                           DO 85 J=1,5
85                          F(J+7)=C(J)
              ENDIF
431           CONTINUE
C      CALCULATE CONST C6 IN ENTHALPY EXPRESSION

*   MUST INSERT HERE CALC OF Hf & S SO THAT 1st DERIVATIVE AND
*     FUNCTION VALUES MATCH AT T BREAK POINT
*
*    TO DO THIS WE HOLD (C1-C5, C8-C12) CONSTANT AND SOLVE FOR C6,C13
*    SUCH THAT Hfl(Tbp)=Hfh(Tbp)
*      MUST INTEGRATE CP BETWEEN 298 AND Tbp
*
*    WE MUST PERFORM A SIMILAR DETERMINATION FOR S TO GET C7,C14
*
*
C
             C6=F6(0,(HF298/1.987),T0,F)
C
C      CALCULATE CONST C7 FROM ENTROPY EXPRESSION
C
      C7=(S298/1.987)-F(1)*DLOG(T0)-F(2)*T0-F(3)/2.*T0**2
     1-F(4)/3.*T0**3-F(5)/4.*T0**4

*       USE F(6)  TO DETERMINE HF(Tbp)
             F(6)=C6
             F(7)=C7
             H0=HF(0,T0,F)*1.987*T0
             S0=ST(0,T0,F)*1.987
**           determine break point temperature
             IF(NATOM.NE.1.AND.CFLAG.NE.'*')THEN
                           TINIT=(TFIT(1)+TFIT(15))/2.0
                           CALL TBREAK (TMID,TINIT)
                           TMID=FLOAT(INT(TMID+0.5))
             ELSE
                           TMID=1000.
             ENDIF
**           THESE MUST BE DETERMINED TO SATISFY H(Tbp)l = H(Tbp)h.
             HFTMID=HF(0,TMID,F)*1.987*TMID
             STMID=ST(0,TMID,F)*1.987
             F(13)=F6(7,(HFTMID/1.987),TMID,F)
      C14=(STMID/1.987)-F(8)*DLOG(TMID)-F(9)*TMID-F(10)/2.*TMID**2
     1-F(11)/3.*TMID**3-F(12)/4.*TMID**4
             F(14)=C14
             HCKTM=HF(7,TMID,F)*1.987*TMID
             SCKTM=ST(7,TMID,F)*1.987
             DELTACP=100.*((CPT(7,TMID,F)-CPT(0,TMID,F))/CPT(0,TMID,F))
             DDELCP=DCPT(7,TMID,F)-DCPT(0,TMID,F)
             DELTAS=100.*((ST(7,TMID,F)-ST(0,TMID,F))/ST(0,TMID,F))
             DDELS=DST(7,TMID,F)-DST(0,TMID,F)
             DELTAH=100.*((HF(7,TMID,F)-HF(0,TMID,F))/HF(0,TMID,F))
             DDELH=DHF(7,TMID,F)-DHF(0,TMID,F)
             YERR=0.0
             IF(ABS(DELTAH).GT.(50.*TOL).OR.ABS(DELTAS).GT.(50.*TOL)
     +.OR.ABS(DDELH).GT.(20.*TOL).OR.ABS(DDELS).GT.(20.*TOL).OR.
     +ABS(DELTACP).GT.(50.*TOL).OR.ABS(DDELCP).GT.(20.*TOL))THEN
             IF(ABS(DELTACP).GT.(50.*TOL))THEN
                          YERR=ABS(DELTACP)
                          WRITE(*,1960)ABS(DELTACP)
1960         FORMAT(1X,'Tmid mismatch: % delta Cp(Tmid) = ',1PE10.3)
             ENDIF
             IF(ABS(DDELCP).GT.(0.005))THEN
                          IF(ABS(DDELCP).GT.YERR)YERR=ABS(DDELCP)
                          WRITE(*,1961)ABS(DDELCP)
1961         FORMAT(1X,'Tmid mismatch:  delta dCp(Tmid) = ',1PE10.3)
             ENDIF
             IF(ABS(DELTAH).GT.(50.*TOL))THEN
                          IF(ABS(DELTAH).GT.YERR)YERR=ABS(DELTAH)
                          WRITE(*,1962)ABS(DELTAH)
1962         FORMAT(1X,'Tmid mismatch: % delta  Hf(Tmid) = ',1PE10.3)
             ENDIF
             IF(ABS(DDELH).GT.(0.005))THEN
                          IF(ABS(DDELH).GT.YERR)YERR=ABS(DDELH)
                          WRITE(*,1963)ABS(DDELH)
1963         FORMAT(1X,'Tmid mismatch:  delta dHf(Tmid) = ',1PE10.3)
             ENDIF
             IF(ABS(DELTAS).GT.(50.*TOL))THEN
                          IF(ABS(DELTAS).GT.YERR)YERR=ABS(DELTAS)
                          WRITE(*,1964)ABS(DELTAS)
1964         FORMAT(1X,'Tmid mismatch: % delta S(Tmid) = ',1PE10.3)
             ENDIF
             IF(ABS(DDELS).GT.(0.005))THEN
                          IF(ABS(DDELS).GT.YERR)YERR=ABS(DDELS)
                          WRITE(*,1965)ABS(DDELS)
1965         FORMAT(1X,'Tmid mismatch:  delta dS(Tmid) = ',1PE10.3)
             ENDIF
             IF(IPOLY.EQ.0.AND.IPOLYC.EQ.0)THEN
                          WRITE(*,*)' repeating polynomial fitting routine'
                          write(*,*)' with a different overlap region'
                          YERRS=YERR
                          YERR1=YERR
                          IPOLYC=IPOLYC+1
                          IPOLY=1
                          SSRS(1)=SSR
                          TM(1)=TMID
                          DO 1357 IK1=1,14
1357                      FS(1,IK1)=F(IK1)
                          GOTO 73590
             ELSEIF(IPOLY.EQ.1)THEN
                          WRITE(*,*)' SECOND REGRESSION FAILED TO MATCH'
                          WRITE(*,*)' PROPERTY AT THE MID-POINT'
             WRITE(*,*)' MAKING A THIRD ATTEMPT TO FIT THE DATA'
                          DO 13571 IK1=1,14
```

```
13571                    FS(2,IK1)=F(IK1)
                         IPOLY=2
                         YERR2=YERR
                         SSRS(2)=SSR
                         TM(2)=TMID
                         GOTO 73590
            ELSEIF(IPOLY.EQ.2)THEN
                         WRITE(*,*)' THIRD ATTEMPT TO MATCH PROPERTIES '
                         WRITE(*,*)' AT THE MIDPOINT HAS FAILED'
                         WRITE(*,*)' LAST REGRESSION IN PROGRESS'
                         IPOLY=3
                         SSRS(3)=SSR
                         YERR3=YERR
                         TM(3)=TMID
                         DO 13572 IK1=1,14
13572                    FS(3,IK1)=F(IK1)
                         GOTO 73590
            ELSEIF(IPOLY.EQ.3)THEN
                         IF(YERR1.LT.YERR2)THEN
                                      IF(YERR1.LT.YERR3)THEN
                                                  KEEP=1
                                      ELSE
                                                  KEEP=3
                                      ENDIF
                         ELSE
                                      IF(YERR2.LT.YERR3)THEN
                                                  KEEP=2
                                      ELSE
                                                  KEEP=3
                                      ENDIF
                         ENDIF
                         WRITE(*,*)' best regression is # ',keep
                         DO 24 IK2=1,14
24                       F(IK2)=FS(KEEP,IK2)
                         TMID=TM(KEEP)
            ENDIF
            ENDIF
*
*       F ( 1-7 )  : LOW TEMPERATURE COEFFICIENTS
*       F ( 8-14 ) : HIGH TEMPERATURE COEFFICIENTS
*
*      REMEMBER:  NASA FORMAT  FIRST 7 COEFFICIENTS (HIGH TEMP)
*                              SECOND 7 COEFFICIENTS (LOW TEMP)
*
12987       CONTINUE

      WRITE(3,1010) INAME,XNAME,(REF(I),I=1,2),REF3,(IE(I),NE(I),I=1,
     1 4),PHASE,TLO,THI,TMID,NROT,IC1
      WRITE(3,112) F(8),F(9),F(10),F(11),F(12),IC2
      WRITE(3,112)F(13),F(14),F(1),F(2),F(3),IC3
      WRITE(3,113)F(4),F(5),F(6),F(7),IC4
 1010 FORMAT(A4,A6,2A4,A6,4(A2,I3),A1,2F10.3,F9.3,4X,2I1)

  112 FORMAT(5(1PE15.8),I5)
```

```
113 FORMAT(4(1PE15.8),15X,I5)
    ICOUNT=ICOUNT+1
    GO TO 20
900 WRITE(3,1012)
           CLOSE(1,STATUS='KEEP')
           CLOSE(3,STATUS='KEEP')
    STOP
    END
```

```
      SUBROUTINE CPEXPOFT(T,CP,NTIN,T1,CP1,NTOUT,CPINF)
$LARGE
*******************************************************************************
*                                                                             *
*       Written by:  Edward R. Ritter   2/26/89                               *
*                                                                             *
*                                                                             *
*                                                                             *
*      This subprogram approximates the heat capacity function  using         *
*    the following expression:                                                *
*                                                                             *
*            Cp (T) = Cpinf { 1 - exp ( - (A+B*T) ) }                         *
*                                                                             *
*      This is rearranged to give the following linear expression:           *
*                                                                             *
*            Y = - ln( 1 - (Cp(T)/Cpinf)) = A + B*T                           *
*                                                                             *
*      This expression is evaluated piece-wise between data points at        *
*     300, 400, 500, 600, 800, and 1000 K.                                    *
*                                                                             *
*     Cp data is then estimated at required temperatures from this eq'n      *
*     using the appropriate expression which covers the T range of interest*
*                                                                             *
*     VARIABLES:                                                              *
*                                                                             *
*         T : vector of temperature points for which Cp is known             *
*        CP : vector of known Cp data points                                  *
*      NTIN : number of data points input                                     *
*        T1 : vector of Temperature for which estimates are required         *
*       CP1 : vector of Cp estimates evaluated at T1                          *
*     NTOUT : number of temperatures for which Cp estimates are required     *
*     CPINF : the high temperature limiting heat capacity                     *
*              CPINF = (3N-2)R    non-linear molecule                         *
*                    = (3N-1.5)R      linear molecule                         *
*******************************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION T(*),CP(*),T1(*),CP1(*),A(30),B(30),YEX(30)
*
*---------------------------------------------------------------
*
*       determine value of Y at each T from input Cp data
*
*          Y =  - LOG( 1 - (Cp(T)/CPinf) )
*
*---------------------------------------------------------------
*
      DO 100 I=1,NTIN
100   YEX (I) = YEXP ( CP(I), CPINF, T(I) )
*     WRITE(*,*)'   T(J)       YEX(J)      YEX(J+1)       A(I)     B(I)'
*
*---------------------------------------------------------------
*
*        determine linear approximations for Y over each
*        temperature interval
```

```
*
*-----------------------------------------------------------------
*

             DO 300 I=1,NTIN-1
             IF(YEX(I+1).EQ.0.0.AND.I.GE.2)THEN
                           A(I)=A(I-1)
                           B(I)=B(I-1)
                           GOTO 300
             ENDIF
             CALL LINE(T(I),T(I+1),YEX(I),YEX(I+1),A(I),B(I))
*333   WRITE(*,'(1X,F10.3,4(2X,1PE15.3))')T(I),YEX(I),YEX(I+1),A(I),B(I)
300          CONTINUE

*
*-----------------------------------------------------------------
*
*       now we calculate Cp at each requested output temperature
*
*-----------------------------------------------------------------
*
*      WRITE(*,*)' K      TJ       CP1(J)       BETA       A(K)       B(K
*     +)'
             CPS=0.0
             DO 10000 J=1,NTOUT
             TJ=T1(J)
             IF(TJ.LE.T(1))THEN
                           K=1
             ELSEIF(TJ.GT.T(1).AND.TJ.LE.T(NTIN))THEN
                           DO 500 I=1,NTIN-1
                           IF(TJ.GT.T(I).AND.TJ.LE.T(I+1))THEN
                                        K=I
                                        GOTO 600
                           ENDIF
500                        CONTINUE
600                        CONTINUE
             ELSE
                           K=NTIN-1
             ENDIF
             IER=0
700          CONTINUE
             Y=A(K)+(B(K)*TJ)
             CP1(J)=CPOFT2(A(K),B(K),CPINF,TJ)
*      WRITE(*,'(1X,I2,2x,2F10.3,3(2X,1PE15.3))')K,TJ,CP1(J),Y,A(K),
*     +B(K)
10000        CONTINUE
             RETURN
             END
```

```
FUNCTION BETAEXP(CP,CPINF,T)
IMPLICIT REAL*8(A-H,O-Z)
Y=LOG(1.0-(CP/CPINF))
BETAEXP=Y/T
RETURN
END


FUNCTION YEXP(CP,CPINF,T)
IMPLICIT REAL*8(A-H,O-Z)
IF(CP.GE.CPINF)THEN
write(*,*)'  Cp > Cpinf:    discarding this data point'
write(*,*)'     T (K)     Cp (T)      Cpinf '
write(*,'(1x,3(2x,1pe15.3))')t,cp,cpinf
            YEXP=0.0
            RETURN
ENDIF
YEXP=LOG(1.0-(CP/CPINF))
RETURN
END

FUNCTION DBDT(B1,B2,T1,T2)
IMPLICIT REAL*8(A-H,O-Z)
DBDT=(B2-B1)/(T2-T1)
RETURN
END

FUNCTION CPOFT(BETA,CPINF,T)
IMPLICIT REAL*8(A-H,O-Z)
CPOFT=CPINF*(1.0-EXP(-1.0*BETA*T))
RETURN
END

FUNCTION CPOFT2(A,B,CPINF,T)
IMPLICIT REAL*8(A-H,O-Z)
Y=A+(B*T)
CPOFT2=CPINF*(1.0-EXP(Y))
RETURN
END


FUNCTION BETAO(BETA,T,A,B)
IMPLICIT REAL*8(A-H,O-Z)
BETAO=BETA-((A*T)+((B/2.0)*T**2))
RETURN
END

FUNCTION BOFT(BETAO,A,B,T)
IMPLICIT REAL*8(A-H,O-Z)
BOFT=BETAO+(A*T)+((B/2)*T**2)
RETURN
END
```

```
SUBROUTINE LINE(X1,X2,Y1,Y2,A,B)
IMPLICIT REAL*8 (A-H,O-Z)
B=(Y2-Y1)/(X2-X1)
A=Y1-(B*X1)
RETURN
END
```

APPENDIX|V - D

FORTRAN SOURCE CODE FOR

THERMFIT

```
      PROGRAM THERMFIT
$LARGE
C***********************************************************************
C     THIS PROGRAM CREATES A NASA FORMAT THERMO DATABASE FILE
C
C            TWO SETS OF COEFFICIENTS ARE GENERATED AFTER FITTING CP DATA
C            USING A MODIFIED VERSION OF THE CPFIT PROGRAM
C
C               TMP DATA FILE (UNIT 45) IS USED TO PASS FILENAMES DIRECTLY
C               TO THIS PROCEDURE FROM THERM.  IF THE FILE "NASAGEN.TMP"
C               DOES NOT EXIST, THEN THE USER IS PROMPTED FOR INPUT/OUTPUT
C               FILENAMES.
C
C
C          INPUT FILE (UNIT 1) IS IN THERMLST FORMAT
C
C          OUTPUT FILE (UNIT 3) CONTAINS COEFFICIENTS IN NASA FORMAT
C          FOR USE WITH   CHEMKIN AND OTHER PACKAGES
C
C***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
             DIMENSION XO(2),WX(2),S(2)
      DIMENSION CPL(30),TRA(3),A(5),B(6),C(14),F(14),FS(3,14),
     +TH(3),TL(3),TM(3),SSRS(3),PERMX(3)
      DIMENSION TFIT(30),CPFIT(30),PWORK(120),D(14),CONST(3)
      DIMENSION X(31,15,1),Y(31,25,1),B1(16,25,1),T(10),TEST(30),
     +CPEST(30)
             REAL*8 TMID
             REAL*8 DELPCT(20)
      CHARACTER*6 TITLE(5),HEADIN(21),REF3,XNAME
      CHARACTER*1 PHASE,CFLAG
      CHARACTER*2 IE(4)
      INTEGER NE(4)
      CHARACTER*4 REF(2),INAME,IFLAG
             CHARACTER*14 FUNITS,NUL
             CHARACTER*70 FILIN,FILOUT,COMMAND,DUMMY,S1
             LOGICAL EXST,OPNED
             EXTERNAL FX
             COMMON/DATA/CPEX1,T1,IT,NT,NS
             COMMON/CONST/CONST,NTERMS
             COMMON/TOL/TOL,TSTEP
             COMMON/TEMPFIT/TFIT,F
             COMMON/FLAG/IREPEAT,DELPCT,IFLG
             COMMON/NROTOR/NROT
      DATA NPTS/7/
      DATA IFLAG/'END'/
      DATA IC1,IC2,IC3,IC4/1,2,3,4/
             T(1)=300.0
             T(2)=400.0
             T(3)=500.0
             T(4)=600.0
             T(5)=800.0
             T(6)=1000.0
             T(7)=1500.0
             TEST(1)=300.
             TEST(2)=350.
             TEST(3)=400.
             TEST(4)=425.
             TEST(5)=450.
             TEST(6)=475.
             TEST(7)=500.
             TEST(8)=550.
             TEST(9)=575.
             TEST(10)=600.
             T0=298.15
             TLO=300.0
             THI=5000.0
             TMID=1500.0
             TOL=1.0E-4
             TSTEP=0.0
14           FORMAT(A14)
             COMMAND=' '
             FILIN=' '
             FILOUT=' '
             INQUIRE(FILE='THERMFIT.TMP',EXIST=EXST)
             IF(EXST)THEN
                           OPEN(45,FILE='THERMFIT.TMP',STATUS='OLD')
                           READ(45,3)COMMAND
                           READ(45,3)FILIN
                           READ(45,3)FILOUT
                           CLOSE(45,STATUS='DELETE')
                           GOTO 123
             ENDIF
        WRITE(*,*)'ENTER FILENAME FOR INPUT {.LST}'
             READ(*,3)FILIN
3            FORMAT(A70)
             CALL FNAME(FILIN,'LST')
             WRITE(*,*)'ENTER FILENAME FOR OUTPUT {.DAT}'
             READ(*,3)FILOUT
             CALL FNAME(FILOUT,'DAT')
123          CONTINUE
             INQUIRE(FILE=FILIN,EXIST=EXST)
             IF(EXST)GOTO 324
             STOP 2
324          OPEN(1,FILE=FILIN,STATUS='OLD')
             OPEN(3,FILE=FILOUT,STATUS='UNKNOWN')
             IF(COMMAND.EQ.' ')THEN
                           WRITE(*,*)' '
                           WRITE(*,*)' '
                           WRITE(*,*)'   GENERATING NASA FORMAT DATABASE FROM'
                           WRITE(*,*)'      ',FILIN
                           WRITE(*,*)' '
                           WRITE(*,*)'            please wait'
             ENDIF
      WRITE(3,99)
      WRITE(3,98)
   99 FORMAT('THERMO ',73(' '))
   98 FORMAT('   300.000  1500.000  5000.000 ',49(' '))
    1 ICOUNT=1
      READ(1,3)DUMMY
             CALL DBLNK(DUMMY,LEN,0)
             CONST(1)=1.0
             FUNITS='KCAL'
             IF(DUMMY(:6).EQ.'UNITS:')THEN
                          IF(DUMMY(7:8).EQ.'KJ')THEN
                                       FUNITS='KJ'
                                       CONST(1)=4.184
                          ENDIF
                          READ(1,3)DUMMY
             ENDIF
   95 FORMAT(5A6/,21A6)
             read(1,14)nul
20           CONTINUE
             DO 203 IK1=1,20
203          DELPCT(IK1)=0.0
             YERRS=1.0E+7
             IREPEAT=0
             IFLG=0
             IPOLY=0
             IPOLYC=0
             CFLAG=' '
             THI=5000.
             READ(1,'(1X,A4)',END=900)INAME
             IF(INAME.EQ.' ')THEN
                          GOTO 20
             ELSE
                          BACKSPACE(1)
             ENDIF
             IF(FUNITS.EQ.'KCAL')THEN
      READ(1,170,END=900)CFLAG,INAME,XNAME,HF298,S298,(CPL(I),I=1,7),
     1 (REF(I),I=1,2),REF3, (IE(I),NE(I),I=1,4),PHASE,NROT
  170 FORMAT(A1,A4,A6,F7.2,2X,F7.2,3X,7(F6.2,2X),2A4,A6,
     1 3X,4(A2,I3,1X),A1,1X,I1)
             ELSE IF(FUNITS.EQ.'KJ')THEN
      READ(1,176,END=900)CFLAG,INAME,XNAME,HF298,S298,(CPL(I),I=1,7),
     1 (REF(I),I=1,2),REF3, (IE(I),NE(I),I=1,4),PHASE,NROT
176   FORMAT(A1,A4,A6,F8.2,1X,F8.2,2X,7(F7.2,1X),2A4,A6,3X,4(A2,I3,
     +  1X),A1,1X,I1)
             ENDIF
      IF(INAME(:3).EQ.'END') THEN
                  WRITE(3,1012)
 1012         FORMAT('END',77(' '))
 1011         FORMAT(A4)
                  STOP 'END OF THERMFIT'
      ENDIF
             IF(CFLAG.EQ.' ')THEN
                          WRITE(*,'(1X,A4,A6)')INAME,XNAME
             ELSE
                          WRITE(*,'(1X,A4,A6,2X,A1)')INAME,XNAME,CFLAG
             ENDIF
C---CONVERTING HF298 TO CAL/MOLE OR J/MOLE
```

```
             IF(FUNITS.EQ.'KCAL')THEN
                   HF298=HF298*1000.
             ELSEIF(FUNITS.EQ.'KJ')THEN
                          HF298=HF298*1000.0/4.184
                          S298=S298/4.184
             ENDIF
      TRA(1)=300.
             NS=0
             NATOM=0
             ICARBON=0
             NCARBON=0
             DO 765 IIJ=1,4
             IF(IE(IIJ).EQ.'C '.OR.IE(IIJ).EQ.' C')THEN
                          ICARBON=1
                          NCARBON=NE(IIJ)
             ENDIF
765          NATOM=NATOM+NE(IIJ)
*            WRITE(*,*)'NATOM= ',NATOM
986          CONTINUE
*            BYPASS FITTING ROUTINE IF ATOM OR Cp <> F(T)
             DIF1=DABS(CPL(1)-CPL(3))
             DIF2=DABS(CPL(1)-CPL(6))
             DIFF=MAX(DIF1,DIF2)
             IF(FUNITS.EQ.'KJ')DIFF=DIFF/4.184
             IF(NATOM.NE.1.AND.DIFF.GT.0.01)THEN
                          IF((ICARBON.EQ.1.AND.(NATOM-NCARBON).EQ.2).OR.NATOM.EQ.2
     +)THEN
                                        NS=(3*NATOM)-5
                                        CONST(2)=FLOAT(NS)
                                        CONST(3)=1.0
                                        CPINFIN=1.987*(FLOAT(3*NATOM)-1.5)*CONST(1)
                          ELSE
                                        NS=(3*NATOM)-6
                                        CONST(3)=0.0
*               count hindered rotor as 1/2 degree of freedom
*                                       CONST(2)=FLOAT(NS)-(FLOAT(NROT)/2.0)
                                        CONST(2)=FLOAT(NS)-(FLOAT(NROT))
                          CPINFIN=1.987*(FLOAT(3*NATOM)-(2+(FLOAT(NROT)/2.0)))*
     +CONST(1)
                          ENDIF
                          IF(CPL(7).EQ.0.0)THEN
                                        NT=6
                          ELSE
                                        NT=7
                          ENDIF
                          IF(CFLAG.EQ.'*')THEN
                                        IFLG=2
                                        GOTO 1994
                          ENDIF
                          CALL THCPFIT(NT,CPL,NATOM,B,T,CPINFIN)

**********************  POLYNOMIAL SMOOTHING PROCEDURE *********

                          IF(IFLG.EQ.1)THEN
```

```
                        DO 1678 IK1=1,NT
                        DO 1677 IK2=1,5
1677                    X(IK1,IK2,1)=T(IK1)**(IK2-1)
                        CPFIT(IK1)=CPL(IK1)/(1.987*CONST(1))
                        Y(IK1,1,1)=CPFIT(IK1)
1678                    CONTINUE
                        MDEG=4
                        NPTS=NT
                        CALL LINLSQ(X,Y,B1,4,NPTS,1,1,SSR,AVGERR)
                        DO 1679     IK1=1,5
1679                    C(IK1)=B1(IK1,1,1)
                        DO 1680 IK1=1,10
                        TIN=TEST(IK1)
                        CPEST(IK1)=CPT(0,TIN,C)*1.987*CONST(1)
1680                    CONTINUE
                        CALL THCPFIT(10,CPEST,NATOM,B,TEST,CPINFIN)
                ENDIF
1994            CONTINUE
                IF(IFLG.EQ.2)THEN
                        IFLG=0
*                       POLYNOMIAL SMOOTHING DID NOT HELP OR USER SPECIFIES
*               POLYNOMIAL FIT ONLY THEN
*                       DEFAULT TO SINGLE REPEATED POLYNOMIAL
*                       OVER TRANGE 300 - 2000 K
                        IF(CPL(7).EQ.0.)THEN
                                CPL(7)=CPL(6)+(0.37*(CPINFIN-CPL(6)))
                        ENDIF
                        CPL(8)=CPL(7)+(0.37*(CPINFIN-CPL(7)))
                        T(8)=2000.
                        DO 61678 IK1=1,8
                        DO 61677 IK2=1,5
61677                   X(IK1,IK2,1)=T(IK1)**(IK2-1)
                        CPFIT(IK1)=CPL(IK1)/(1.987*CONST(1))
                        Y(IK1,1,1)=CPFIT(IK1)
61678                   CONTINUE
                        MDEG=4
                        NPTS=8
                        CALL LINLSQ(X,Y,B1,4,NPTS,1,1,SSR,AVGERR)
                        DO 61679 IK1=1,5
61679                   C(IK1)=B1(IK1,1,1)
                        DO 61681 IK1=1,5
                        F(IK1)=C(IK1)
                        F(IK1+7)=C(IK1)
61681                   CONTINUE

                        C6=F6(0,(HF298/1.987),T0,F)
             C7=(S298/1.987)-F(1)*DLOG(T0)-F(2)*T0-F(3)/2.*T0**2
     1          -F(4)/3.*T0**3-F(5)/4.*T0**4
                        F(6)=C6
                        F(13)=C6
                        F(7)=C7
                        F(14)=C7
                        TMID=1000.
                        THI=2000.0
```

```
                                         ICOUNT=0
6545                                     DCPDT=DCPT(0,THI,F)
                                         COMPARE=((ABS(DCPDT)*500.)/CPT(0,THI,F))*100.
                                         IF(DCPDT.LT.0..AND.COMPARE.GT.10.)THEN
                                                       ICOUNT=ICOUNT+1
                                                       THI=THI-50.
                                                       IF(ICOUNT.LE.100)GOTO 6545
                                         ENDIF
                                         GOTO 12987
                            ENDIF
*********************************************************************
              ENDIF
73590         CONTINUE
              DO 431 JK=1,2
              IF(NATOM.NE.1.AND.DIFF.GT.0.01)THEN
              IF(IPOLY.EQ.1)THEN
                            IF(JK.EQ.1)THEN
                            TK=253.3333
                            DO 1290 I8=1,15
                                         TK=TK+46.6667
                                         TFIT(I8)=TK
                                         TFIT(I8+15)=1000.+(71.4236*FLOAT(I8-1))
1290                        CONTINUE
*             set lower temperature below room temp so that 298 is not
*         outside the range of the regression for Cp
                            TFIT(1)=273.15
                            ELSE
                            TK=928.5764
                            DO 1291 I8=1,15
                                         TFIT(I8)=1000.+(71.4236*FLOAT(I8-1))
                                         TFIT(I8+15)=2000.+(266.667*FLOAT(I8))
1291                        CONTINUE
                            ENDIF
              ELSEIF(IPOLY.EQ.0)THEN
                            IF(JK.EQ.1)THEN
                            TK=236.6667
                            DO 21290 I8=1,15
                                         TK=TK+63.3333
                                         TFIT(I8)=TK
                                         TFIT(I8+15)=1250.+(35.714*FLOAT(I8-1))
21290                       CONTINUE
*             set lower temperature below room temp so that 298 is not
*         outside the range of the regression for Cp
                            TFIT(1)=273.15
                            ELSE
                            DO 21291 I8=1,15
                                         TFIT(I8)=1250.+(35.714*FLOAT(I8-1))
                                         TFIT(I8+15)=1700.+(283.333*FLOAT(I8))
21291                       CONTINUE
                            ENDIF
              ELSEIF(IPOLY.EQ.3)THEN
                            IF(JK.EQ.1)THEN
                            TK=263.333
                            DO 22290 I8=1,15
```

```
                                       TK=TK+36.667
                                       TFIT(I8)=TK
                                       TFIT(I8+15)=850.+(35.714*FLOAT(I8-1))
22290                      CONTINUE
*            set lower temperature below room temp so that 298 is not
*        outside the range of the regression for Cp
                           TFIT(1)=273.15
                           ELSE
                           DO 22291 I8=1,15
                                       TFIT(I8)=850.+(35.714*FLOAT(I8-1))
                                       TFIT(I8+15)=1350.+(310.0*FLOAT(I8))
22291                      CONTINUE
                           ENDIF
             ELSEIF(IPOLY.EQ.2)THEN
                           IF(JK.EQ.1)THEN
                           TK=263.333
                           DO 23290 I8=1,15
                                       TK=TK+36.667
                                       TFIT(I8)=TK
                                       TFIT(I8+15)=850.+(21.429*FLOAT(I8-1))
23290                      CONTINUE
*            set lower temperature below room temp so that 298 is not
*        outside the range of the regression for Cp
                           TFIT(1)=273.15
                           ELSE
                           DO 23291 I8=1,15
                                       TFIT(I8)=850.+(21.429*FLOAT(I8-1))
                                       TFIT(I8+15)=1150.+(323.333*FLOAT(I8))
23291                      CONTINUE
                           ENDIF
             ENDIF
      TRA(1)=300.
      TRA(2)=1000.
      DO 30 I=1,30
      DO 31 I11=1,5
   31 X(I,I11,1)=TFIT(I)**(I11-1)
             TIN=TFIT(I)
             CPFIT(I)=CPSERIES(B,TIN)
      Y(I,1,1)=CPFIT(I)
   30 CONTINUE
      MDEG=4
      RSQ=100.
      IER=0
             NPTS=30
                   CALL LINLSQ(X,Y,B1,4,NPTS,1,1,SSR,AVGERR)
                   DO 32 I=1,5
   32               C(I)=B1(I,1,1)

      ELSE
*                          ATOM:  USE DIFFERENT METHOD FOR Cp
C    BYPASS FITTING ROUTINE IF CP <> f(T)
                           SUMCP=0.0
                           DATPNT=0.0
                           DO 1209 IJK=1,7
                           IF(CPL(IJK).NE.0.0)THEN
                                        SUMCP=SUMCP+CPL(IJK)
                                        DATPNT=DATPNT+1.0
                           ENDIF
1209                       CONTINUE
              C(1)=SUMCP/(1.98717*DATPNT)
                           IF(FUNITS.EQ.'KJ')C(1)=C(1)/4.184
                         DO 15 I=2,5
                         C(I)=0.0
   15                CONTINUE
                           TMID=1000.
      ENDIF
             IF(JK.EQ.1)THEN
                                 DO 75 I=1,5
                                 F(I)=C(I)
   75                       CONTINUE
             ELSEIF(JK.EQ.2)THEN
                           DO 85 J=1,5
85                          F(J+7)=C(J)
             ENDIF
431          CONTINUE
C      CALCULATE CONST C6 IN ENTHALPY EXPRESSION

*   MUST INSERT HERE CALC OF Hf & S SO THAT 1st DERIVATIVE AND
*     FUNCTION VALUES MATCH AT T BREAK POINT
*
*    TO DO THIS WE HOLD (C1-C5, C8-C12) CONSTANT AND SOLVE FOR C6,C13
*    SUCH THAT Hfl(Tbp)=Hfh(Tbp)
*      MUST INTEGRATE CP BETWEEN 298 AND Tbp
*
*    WE MUST PERFORM A SIMILAR DETERMINATION FOR S TO GET C7,C14
*
*
C
             C6=F6(0,(HF298/1.987),T0,F)
C
C      CALCULATE CONST C7 FROM ENTROPY EXPRESSION
C
      C7=(S298/1.987)-F(1)*DLOG(T0)-F(2)*T0-F(3)/2.*T0**2
     1-F(4)/3.*T0**3-F(5)/4.*T0**4

*       USE F(6)  TO DETERMINE HF(Tbp)
             F(6)=C6
             F(7)=C7
             H0=HF(0,T0,F)*1.987*T0
             S0=ST(0,T0,F)*1.987
**           determine break point temperature
             IF(NATOM.NE.1.AND.CFLAG.NE.'*')THEN
                           TINIT=(TFIT(1)+TFIT(15))/2.0
                           CALL TBREAK (TMID,TINIT)
                           TMID=FLOAT(INT(TMID+0.5))
             ELSE
                           TMID=1000.
             ENDIF
```

```
**          THESE MUST BE DETERMINED TO SATISFY H(Tbp)l = H(Tbp)h.
            HFTMID=HF(0,TMID,F)*1.987*TMID
            STMID=ST(0,TMID,F)*1.987
            F(13)=F6(7,(HFTMID/1.987),TMID,F)
      C14=(STMID/1.987)-F(8)*DLOG(TMID)-F(9)*TMID-F(10)/2.*TMID**2
     1-F(11)/3.*TMID**3-F(12)/4.*TMID**4
            F(14)=C14
            HCKTM=HF(7,TMID,F)*1.987*TMID
            SCKTM=ST(7,TMID,F)*1.987
            DELTACP=100.*((CPT(7,TMID,F)-CPT(0,TMID,F))/CPT(0,TMID,F))
            DDELCP=DCPT(7,TMID,F)-DCPT(0,TMID,F)
            DELTAS=100.*((ST(7,TMID,F)-ST(0,TMID,F))/ST(0,TMID,F))
            DDELS=DST(7,TMID,F)-DST(0,TMID,F)
            DELTAH=100.*((HF(7,TMID,F)-HF(0,TMID,F))/HF(0,TMID,F))
            DDELH=DHF(7,TMID,F)-DHF(0,TMID,F)
            YERR=0.0
            IF(ABS(DELTAH).GT.(50.*TOL).OR.ABS(DELTAS).GT.(50.*TOL)
     +.OR.ABS(DDELH).GT.(20.*TOL).OR.ABS(DDELS).GT.(20.*TOL).OR.
     +ABS(DELTACP).GT.(50.*TOL).OR.ABS(DDELCP).GT.(20.*TOL))THEN
            IF(ABS(DELTACP).GT.(50.*TOL))THEN
                        YERR=ABS(DELTACP)
                        WRITE(*,1960)ABS(DELTACP)
1960        FORMAT(1X,'Tmid mismatch: % delta Cp(Tmid) = ',1PE10.3)
            ENDIF
            IF(ABS(DDELCP).GT.(20.*TOL))THEN
                        IF(ABS(DDELCP).GT.YERR)YERR=ABS(DDELCP)
                        WRITE(*,1961)ABS(DDELCP)
1961        FORMAT(1X,'Tmid mismatch:  delta dCp(Tmid) = ',1PE10.3)
            ENDIF
            IF(ABS(DELTAH).GT.(50.*TOL))THEN
                        IF(ABS(DELTAH).GT.YERR)YERR=ABS(DELTAH)
                        WRITE(*,1962)ABS(DELTAH)
1962        FORMAT(1X,'Tmid mismatch: % delta  Hf(Tmid) = ',1PE10.3)
            ENDIF
            IF(ABS(DDELH).GT.(20.*TOL))THEN
                        IF(ABS(DDELH).GT.YERR)YERR=ABS(DDELH)
                        WRITE(*,1963)ABS(DDELH)
1963        FORMAT(1X,'Tmid mismatch:  delta dHf(Tmid) = ',1PE10.3)
            ENDIF
            IF(ABS(DELTAS).GT.(50.*TOL))THEN
                        IF(ABS(DELTAS).GT.YERR)YERR=ABS(DELTAS)
                        WRITE(*,1964)ABS(DELTAS)
1964        FORMAT(1X,'Tmid mismatch: % delta S(Tmid) = ',1PE10.3)
            ENDIF
            IF(ABS(DDELS).GT.(20.*TOL))THEN
                        IF(ABS(DDELS).GT.YERR)YERR=ABS(DDELS)
                        WRITE(*,1965)ABS(DDELS)
1965        FORMAT(1X,'Tmid mismatch:  delta dS(Tmid) = ',1PE10.3)
            ENDIF
            IF(IPOLY.EQ.0.AND.IPOLYC.EQ.0)THEN
                        WRITE(*,*)' repeating polynomial fitting routine'
                        write(*,*)' with a different overlap region'
                        YERRS=YERR
                        YERR1=YERR
```

```
                          IPOLYC=IPOLYC+1
                          IPOLY=1
                          SSRS(1)=SSR
                          TM(1)=TMID
                          DO 1357 IK1=1,14
1357                      FS(1,IK1)=F(IK1)
                          GOTO 73590
             ELSEIF(IPOLY.EQ.1)THEN
                          WRITE(*,*)' SECOND REGRESSION FAILED TO MATCH'
                          WRITE(*,*)' PROPERTY AT THE MID-POINT'
             WRITE(*,*)' MAKING A THIRD ATTEMPT TO FIT THE DATA'
                          DO 13571 IK1=1,14
13571                     FS(2,IK1)=F(IK1)
                          IPOLY=2
                          YERR2=YERR
                          SSRS(2)=SSR
                          TM(2)=TMID
                          GOTO 73590
             ELSEIF(IPOLY.EQ.2)THEN
                          WRITE(*,*)' THIRD ATTEMPT TO MATCH PROPERTIES '
                          WRITE(*,*)' AT THE MIDPOINT HAS FAILED'
                          WRITE(*,*)' LAST REGRESSION IN PROGRESS'
                          IPOLY=3
                          SSRS(3)=SSR
                          YERR3=YERR
                          TM(3)=TMID
                          DO 13572 IK1=1,14
13572                     FS(3,IK1)=F(IK1)
                          GOTO 73590
             ELSEIF(IPOLY.EQ.3)THEN
                          IF(YERR1.LT.YERR2)THEN
                                       IF(YERR1.LT.YERR3)THEN
                                                    KEEP=1
                                       ELSE
                                                    KEEP=3
                                       ENDIF
                          ELSE
                                       IF(YERR2.LT.YERR3)THEN
                                                    KEEP=2
                                       ELSE
                                                    KEEP=3
                                       ENDIF
                          ENDIF
                          WRITE(*,*)' best regression is # ',keep
                          DO 24 IK2=1,14
24                        F(IK2)=FS(KEEP,IK2)
                          TMID=TM(KEEP)
             ENDIF
             ENDIF
*
*       F ( 1-7 )  : LOW TEMPERATURE COEFFICIENTS
*       F ( 8-14 ) : HIGH TEMPERATURE COEFFICIENTS
*
*      REMEMBER:  NASA FORMAT  FIRST 7 COEFFICIENTS (HIGH TEMP)
*                            SECOND 7 COEFFICIENTS (LOW TEMP)
*
12987        CONTINUE
      WRITE(3,1010) INAME,XNAME,(REF(I),I=1,2),REF3,(IE(I),NE(I),I=1,
     1 4),PHASE,TLO,THI,TMID,NROT,IC1
      WRITE(3,112) F(8),F(9),F(10),F(11),F(12),IC2
      WRITE(3,112)F(13),F(14),F(1),F(2),F(3),IC3
      WRITE(3,113)F(4),F(5),F(6),F(7),IC4
 1010 FORMAT(A4,A6,2A4,A6,4(A2,I3),A1,2F10.3,F9.3,4X,2I1)
  112 FORMAT(5(1PE15.8),I5)
  113 FORMAT(4(1PE15.8),15X,I5)
      ICOUNT=ICOUNT+1
      GO TO 20
  900 WRITE(3,1012)
             CLOSE(1,STATUS='KEEP')
             CLOSE(3,STATUS='KEEP')
      STOP
      END
```

```
              SUBROUTINE THCPFIT(NT,CP,NATOM,AI,TEMPIN,CPINF)
$LARGE
C
              IMPLICIT REAL*8(A-H,O-Z)
      CHARACTER*8 FMT1
      CHARACTER*39 FMT2
      CHARACTER*33 FMT3A(5)
      CHARACTER*44 FMT3B
      CHARACTER*21 FMT4A(5)
      CHARACTER*13 FMT4B
      CHARACTER*1 PLTCHR,DIGIT(10)
      CHARACTER*72 TITLE(20),TEXT
      CHARACTER*4 BLANK,TEST
      CHARACTER*72 FIN, FOUT
      LOGICAL DONE, NEG
      REAL*8 NORML, MEAN
      DIMENSION B(6),BMIN(6),BMAX(6),Y(20),Z(20),PDERIV(6,20),
     @          BI(6),ZI(20),CONV(5),X(1,20),NORMAL(7,7),WORK(7),
     @          SCALE(6),GRAD(6),DIAG(7),CONST(3),VAR(6),STD(6),
     @          IPIVOT(6), COVAR(6,6), NORML(7,7), DIAG2(6),
     @          IX(120), XTEMP(120),CP(30),AI(6),DELPCT(20),BS(5),
     @          BSMIN(5),BSMAX(5),TEMPIN(*)
      DOUBLE PRECISION SUMSQ,B,BMIN,BMAX,BI,NORMAL,GRAD,WORK,DIAG,CONST
     @,SCALE,BTEMP,MLARGE,PLARGE,AVG,COEF2,COEF3,AI,CP,BS,BSMIN,BSMAX,
     @BTMP,TEMPIN
      INTEGER PARM,DATA,ERROR,ERROR2,INPUT,OUTPUT,
     %        OPTION,FREQ,MAXPAR,MAXIND,
     %        MAXDAT,NTITLE,MAXIT,NIND,OPT2,INTMSK,
     %        PRTOPT,PLTOPT,MAXCON,NT,BFLAG
      EQUIVALENCE (IPIVOT(1), SCALE(1)), (COVAR(1,1), PDERIV(1,1)),
     @            (NORML(1,1), NORMAL(1,1)), (DIAG2(1), DIAG(1)),
     @            (IOVER1, OVERF1), (IOVER2, OVERF2),
     @            (IX(1), PDERIV(1,1)), (XTEMP(1), PDERIV(1,1))
      COMMON /DINFO/ IPARM1, IDATA, IPARM, INDPT
      COMMON /CONST/ CONST,NTERMS
      COMMON /PDV/ PDERIV
      COMMON/FLAG/IREPEAT,DELPCT,IFLAG
              SAVE BSMAX,BSMIN,BS
      DATA INPUT/5/,OUTPUT/6/,BLANK/'    '/,MAXPAR/20/,MAXIND/10/
      DATA MAXDAT/400/,MAXCON/40/
      DATA MLARGE/-1.6D308/,PLARGE/1.6D308/
      DATA FMT1/'(8F10.5)'/,DONE/.TRUE./
      DATA FMT2/'(3X  /(9H       B(,I2,4H)  =,1PE16.7))'/
      DATA FMT3A/'(3X  /8H        ,1(1HX,I1,12X)\)',
     &           '(3X  /8H        ,2(1HX,I1,12X)\)',
     &           '(3X  /8H        ,3(1HX,I1,12X)\)',
     &           '(3X  /8H        ,4(1HX,I1,12X)\)',
     &           '(3X  /8H        ,5(1HX,I1,12X)\)'/
      DATA FMT3B/'(2HX,13X,2HY,13X,2HYC,11X,4HYC-Y,10X,3HPCT//) '/
      DATA FMT4A/'(5H     ,4(1PE14.5)\)','(5H     ,5(1PE14.5)\)',
     &           '(5H     ,6(1PE14.5)\)','(5H     ,7(1PE14.5)\)',
     &           '(5H     ,8(1PE14.5)\)'/, FMT4B/'(1H ,0PF10.2)'/
      DATA DIGIT/'0','1','2','3','4','5','6','7','8','9'/,PLTCHR/'O'/
```

```
       CALL RESET
      IPARM = MAXPAR
      IDATA = MAXDAT
      IPARM1 = MAXPAR + 1
      INDPT = MAXIND
100   NTITLE = 1
110   CONTINUE
1     FORMAT(A72)
120          CONTINUE
             FREQ=1000
             MAXIT=65
             CONV(1)=1.0E-2
             CONV(2)=1.0E-2
             CONV(3)=0.0
             CONV(3)=0.0
             CONV(4)=0.0
             CONV(5)=0.0
*            DEFINE  DEFAULTS
             IREPEAT=0
             IREPNUM=0
             option=1
             opt2=1
             nprt=1
             prtopt=1
             pltopt=1
             icall=0
             parm=5
             nterms=3
             nind=1
             ISTART=1
8     FORMAT(3D10.8)
             iconst=3
             fvalue=0.
             tvalue=0.
32    FORMAT(7E10.0,8X,I2)
11    FORMAT(8D10.0)
*            CONST(1): GAS CONSTANT CONVERSION FACTOR;   CONST(1)=1.0 (CAL)
*                                                   CONST(1)=4.184 (J)
*        CONST(2) = TOTAL NUMBER OF VIBRATIONAL DEGREES OF FREEDOM
*
*            CONST(3) = 0., IF NON-LINEAR MOLECULE; = 1., IF LINEAR
170   NIND1 = NIND + 1
C
C             FILL DATA POINT ARRAY WITH CP VALUES.
C
180   CONTINUE
             DO 190 IK1=1,NT
             Y(IK1)=CP(IK1)
             X(1,IK1)=TEMPIN(IK1)
190   CONTINUE
             DATA=NT+1
195           CONTINUE
```

```
*   DETERMINE TYPE OF MOLECULE:  LOW, MID, HIGH FREQUENCY
*   FROM CP(300)/CONST(2) : HEAT CAPACITY PER OSCILLATOR
*         FIRST DATA POINT MUST BE CP(300)

**                            set last data point equal to cpinf
**                            at last temperature = 99999. K
**
                              Y(DATA)=CPINF

**                  continue test for frequency range
*
*         CPRATIO=Y(1)/CPINF

* SET DEFAULT STARTING POINT TO MID FREQUENCY WEIGHTING
          CPRATIO=0.5
          IREPEAT=0
          IREPNUM=0
          IF(IFLAG.EQ.1)THEN
                    DO 7634 IK1=1,5
                    B(IK1)=BS(IK1)
                    BMAX(IK1)=BSMAX(IK1)
                    BMIN(IK1)=BSMIN(IK1)
7634                CONTINUE
          GOTO 9865
          ENDIF
9268      CONTINUE
          IF(IREPEAT.NE.0)THEN
                    IREPNUM=IREPNUM+1
                    IREPEAT=0
                    IF(IREPNUM.GE.7)THEN
                              WRITE(*,*) ' CpFIT FAILURE '
                              IF(IFLAG.EQ.0)THEN
                    WRITE(*,*) ' POLYNOMIAL SMOOTHING OF DATA IN PROGRESS'
                              IFLAG=1
                              ELSE
          WRITE(*,*)' TEMPERATURE RANGE DEFAULTING TO 300 - 2000 K'
                              IFLAG=2
                              ENDIF
                              RETURN
                    ENDIF
                    IF(IREPNUM.GT.1)THEN
                              IBS=0
                              DO 9655 I=1,5
                              DELTAB=(ABS(BS(I)-B(I))/BS(I))
                              IF(DELTAB.GT.0.01)IBS=1
9655                          CONTINUE
                              IF(IBS.EQ.0)THEN
                                        IREPNUM=NREPEAT
                                        IREPEAT=2
                                        GOTO 600
                              ENDIF
                    ENDIF
                    IF(DELPCT(1).LE.0.0.AND.IREPNUM.EQ.1)THEN
                              CPRATIO=0.75
```

```
                                      DO 9651 I=1,5
                                      BS(I)=B(I)
                                      BSMIN(I)=BMIN(I)
                                      BSMAX(I)=BMAX(I)
9651                                  CONTINUE
                                      BFLAG=0
                         ELSE IF(DELPCT(1).GT.0.0.AND.IREPNUM.EQ.1)THEN
                                      CPRATIO=0.25
                                      DO 9652 I=1,5
                                      BS(I)=B(I)
                                      BSMIN(I)=BMIN(I)
                                      BSMAX(I)=BMAX(I)
9652                                  CONTINUE
                                      BFLAG=1
                         ELSEIF((DELPCT(1).LT.0.0.AND.DELPCT(2).LT.0.0)
     +                 .AND.IREPNUM.GT.1)THEN
                                      IF(ABS(B(3)-BMIN(3)).LE.0.1)THEN
                                                   BMIN(3)=BMIN(3)-50.
                                      ELSE
                                                   IF(BFLAG.EQ.1)THEN
                                                                DO 9653 I=1,5
                                                                BTMP=B(I)
                                                                B(I)=(B(I)+BS(I))/2.0
                                                                BS(I)=BTMP
                                                   IF(BMAX(I).LT.BSMAX(I))BMAX(I)=BSMAX(I)
                                                   IF(BMIN(I).GT.BSMIN(I))BMIN(I)=BSMIN(I)
9653                                               CONTINUE
                                                   ENDIF
                                      ENDIF
                                      BFLAG=0
                                      GOTO 9865
                         ELSEIF((DELPCT(1).GT.0.0.AND.DELPCT(2).GT.0.0)
     +                 .AND.IREPNUM.GT.1)THEN
                                      IF(ABS(B(3)-BMIN(3)).LE.0.1)THEN
                                                   BMIN(3)=BMIN(3)+50.
                                                   B(3)=BMIN(3)+50.
                                      ELSE
                                                   IF(BFLAG.EQ.0)THEN
                                                                DO 9654 I=1,5
                                                                BTMP=B(I)
                                                                B(I)=(B(I)+BS(I))/2.0
                                                                BS(I)=BTMP
                                                   IF(BMAX(I).LT.BSMAX(I))BMAX(I)=BSMAX(I)
                                                   IF(BMIN(I).GT.BSMIN(I))BMIN(I)=BSMIN(I)
9654                                               CONTINUE
                                                   BLFAG=1
                                                   ELSE
                                                                BMIN(1)=0.
                                                                BMIN(2)=0.
                                                                BMIN(3)=100.
                                                                BMIN(4)=300.
                                                                BMIN(5)=700.
                                                                BMAX(1)=CONST(2)
                                                                BMAX(2)=CONST(2)
                                                                BMAX(3)=999.
                                                                BMAX(4)=1999.
                                                                BMAX(5)=3600.
                                                    ENDIF

                                        ENDIF
                                        GOTO 9865
                           ENDIF

            ENDIF
*           set initial parameter values, max & min
*           RANGE DETERMINED BY CPRATIO
*                        CPRATIO < 0.4  :  VERY HIGH FREQUENCY WEIGHTED
*             0.4 < CPRATIO < 0.65 :  MID FREQUENCY WEIGHTED
*                   CPRATIO > 0.65 :  LOW FREQUENCY WEIGHTED
            IF(CPRATIO.LT.0.4)THEN
*                        starting point : high frequency weighted
                         B(1)=0.2*CONST(2)
                         B(2)=0.5*CONST(2)
                         B(3)=600.
                         B(4)=1500.
                         B(5)=3000.
                         BMIN(1)=0.
                         BMIN(2)=0.
                         BMIN(3)=350.
                         BMIN(4)=900.
                         BMIN(5)=1500.
                         BMAX(1)=0.6*CONST(2)
                         BMAX(2)=CONST(2)
                         BMAX(3)=1500.
                         BMAX(4)=2500.
                         BMAX(5)=3600.
                         IF(IREPNUM.NE.1)THEN
                                     DO 911 IKJ=1,5
911                                  BS(IKJ)=B(IKJ)
                         ENDIF
            ELSE IF(CPRATIO.GE.0.4.AND.CPRATIO.LE.0.63)THEN
*                        starting point : spread over range
                         B(1)=0.3*CONST(2)
                         B(2)=0.3*CONST(2)
                         B(3)=600.
                         B(4)=1200.
                         B(5)=2400.
                         BMIN(1)=0.
                         BMIN(2)=0.
                         BMIN(3)=250.
                         BMIN(4)=800.
                         BMIN(5)=900.
                         BMAX(1)=CONST(2)
                         BMAX(2)=CONST(2)
                         BMAX(3)=1500.
                         BMAX(4)=2500.
                         BMAX(5)=4000.
                         IF(IREPNUM.NE.1)THEN
```

```
                                        DO 9121 IKJ=1,5
9121                                    BS(IKJ)=B(IKJ)
                           ENDIF
             ELSE IF(CPRATIO.GT.0.63)THEN
*                          starting point: low frequency weighted
*                          parameters are constrained to low frequencies;
*                          with lower minimum frequencies acceptable.
                           B(1)=0.5*CONST(2)
                           B(2)=0.4*CONST(2)
                           B(3)=300.
                           B(4)=500.
                           B(5)=700.
                           BMIN(1)=CONST(2)/10.
                           BMIN(2)=CONST(2)/10.
                           BMIN(3)=100.
                           BMIN(4)=401.
                           BMIN(5)=601.
                           BMAX(1)=CONST(2)
                           BMAX(2)=CONST(2)
                           BMAX(3)=450.
                           BMAX(4)=1000.
                           BMAX(5)=2000.
                           IF(IREPNUM.NE.1)THEN
                                        DO 913 IKJ=1,5
913                                     BS(IKJ)=B(IKJ)
                           ENDIF
             ENDIF
             GOTO 9865
13     FORMAT (/,10(' ',A72/))
23     FORMAT (/' ',15X,'INPUT DATA'//'    FORMAT FOR DATA POINTS'
     *          ,5X,A8)
14     FORMAT ('    NUMBER OF PARAMETERS',10X,I4/
     #         '    MAXIMUM NO. OF ITERATIONS',5X,I4/
     #         '    NUMBER OF DATA POINTS',9X,I4/
     #         '    FREQUENCY OF PRINTING',9X,I4//' COEFFICIENT',
     #              6X,'STARTING',9X,'LOWER',11X,'UPPER'/'   NUMBER',
     #              10X,'VALUE',2(11X,'LIMIT')//30(I7,7X,3(1PE16.8)/))
9865         ioutlim=0
9866         continue
             IREDUCE=0
      DO 200 ITER = ISTART,MAXIT

*  SUBROUTINE SOLVE IS A CANNED MARQUARDT'S METHOD SUBROUTINE
*   PROVIDED BY Dr. Dana Knox OF NJIT Chem.Eng. DEPT
*   A LISTING OF THIS CODE IS NOT PROVIDED IN THIS THESIS

         CALL SOLVE (PARM,DATA,ERROR,ERROR2,B,BMIN,BMAX,Y,Z,
     &               SUMSQ,CONV,NORMAL,SCALE,
     &               WORK,GRAD,BI,ZI,DIAG,X )
             IF(SUMSQ.LT.CONV(1)) GOTO 600
C
C            CHECK FOR TERMINAL ERROR CODES FROM SUBROUTINE
C            SOLVE.
C
```

```
          IF (ERROR .EQ. -1)   GO TO 900
          IF (ERROR .EQ. -2)   GO TO 900
          IF (ERROR .EQ. -3)  then
                            if(ioutlim.eq.1)goto 900
                            ioutlim=1
                            BMIN(1)=0.
                            BMIN(2)=0.
                            BMIN(3)=75.
                            BMIN(4)=150.
                            BMIN(5)=300.
                            BMAX(1)=CONST(2)
                            BMAX(2)=CONST(2)
                            BMAX(3)=1500.
                            BMAX(4)=3500.
                            BMAX(5)=4500.
                            GOTO 9866
              endif
          IF (ERROR .EQ. 0)    GO TO 600
          IF (ITER/FREQ*FREQ .NE. ITER)      GO TO 20001
          WRITE (OUTPUT,18) ITER,SUMSQ
18        FORMAT('0'/' ITERATION',I4/'OSUM OF SQUARES',1PE17.8)
20001         continue
200       CONTINUE
      WRITE (OUTPUT,19)
19    FORMAT (/' MAXIMUM NUMBER OF ITERATIONS EXCEEDED')
      ITER=MAXIT
600           CONTINUE
              AI(1)=B(1)
              AI(2)=B(2)
              AI(3)=CONST(2)-(B(1)+B(2))
              AI(4)=B(3)
              AI(5)=B(4)
              AI(6)=B(5)
20    FORMAT (' F I N A L  R E S U L T S',I8,' ITERATIONS PERFORMED')
45    FORMAT (/' SUM OF SQUARES = ',1PE17.8)
8764          FORMAT(1X,/,5X,'Cvvib: sum of ',i2,' terms'/)
8763          format(1x,'coef: A(i)          exp:    A(i+n)')
8765          format(1x,/,1PE10.3,'               ',1PE10.3)
8990          FORMAT(1X,/,'          Cp = 4.0 + C vib   (non-linear molecule)'
     +/)
8991          format(1x,/,'          Cp = 7/2 + C vib   (    linear molecule)'
     +/)
*             format(/,'  C vib =
8992    format(1x,'                  n')
8993    format(1x,'               _______')
8994    format(1x,'               \       ')
8995    format(1x,'                \                         2       ex
     +p(A(i+n)*(hc/kT))')
8996    format(1x,'       C vib =  /    A(i)*[A(i+n)*(hc/kT)]  .  _____
     _______________________')
8997    format(1x,'               /______
     +                    2')
8998    format(1x,24x,'i',24x,'[1 - exp(A(i+n)*hc/kT)]'//)
```

```
      IF (NPRT .LT. NIND)  NPRT = NIND
      IF (NPRT .GT. 5)       NPRT = 5
9987  FORMAT(1X,'X',13X,'Y',13X,'YC',11X,'YC-Y',10X,'PCT',//)
      SUMP = 0.0
      DO 630 I = 1,DATA
         ZI(I) = Z(I) - Y(I)
         IF (Y(I) .EQ. 0.0)  GO TO 620
         PCT = 100.0 * ZI(I) / Y(I)
         GO TO 625
620      PCT = 0.0
625           CONTINUE
              IF(ABS(PCT).GT.4.5.AND.IREPEAT.NE.2)THEN
                            IREPEAT=1
                            DELPCT(I)=PCT
              ELSEIF(ABS(PCT).GT.5.5.AND.IREPEAT.EQ.2)THEN
                            IREPEAT=1
                            IREPNUM=7
                            GOTO 9268
              ENDIF
         SUMP = SUMP + ABS(PCT)
630           CONTINUE
      SUMSQR = SUMSQ
      FDATA = DATA
      FPARM = PARM
      SEE = SQRT (SUMSQR/ FDATA)
      AVP = SUMP / FDATA
      IRDF = DATA - PARM
      RDF = IRDF
      RMS = SUMSQR / RDF
      RRMS = SQRT(RMS)
22    FORMAT('0    CONSTANTS : ',1P8D12.5/)
21    FORMAT (/' STANDARD ERROR OF ESTIMATE =',1PE15.6/
     @        ' ABSOLUTE AVERAGE PERCENT ERROR = ',0PF7.2/
     @        ' RESIDUAL DEGREES OF FREEDOM =',I4/
     @        ' RESIDUAL MEAN SQUARE =',1PE15.6/
     @        ' RESIDUAL ROOT MEAN SQUARE =',E15.6)
              IF(IREPEAT.NE.0)THEN
                            CALL RESET
                            GOTO 9268
              ENDIF
              RETURN
900           WRITE(*,*)' ERROR ENCOUNTERED IN SUBROUTINE CPFIT'
              WRITE(*,*)' ERROR CODE = ',ERROR
              WRITE(*,*)'      B(I)      BMIN(I)      BMAX(I)'
              DO 912 IK1=1,5
912           WRITE(*,'(1X,3(F10.3,1X))')B(IK1),BMIN(IK1),BMAX(IK1)
              STOP ' CPFIT TERMINATED DUE TO ERROR'
      END
```

APPENDIX IV - E

CALCULATION OF HEAT CAPACITY

AND EFFECT OF HINDERED INTERNAL ROTATION

JANAF : HEXAFLUOROETHANE : $C2F6$

NON - LINEAR MOLECULE
Cpinf (cal/mol K) = 37.75
NUMBER OF OSCILLATORS = 17
Arithmetic mean frequency = 745.765 (1/cm)
Geometric mean frequency = 637.699 (1/cm)
Root Mean Square frequency = 838.446 (1/cm)
Mean Square frequency = 7.030E+05 (1/cm^2)
Squre of Mean frequency = 5.562E+05 (1/cm^2)
modified frequency dispersion parameter = 1.190E+00

TEMP(K)	Cp(cal/mol K)	Cp(J/mol K)	1/Qf	V/RT	Cp_{ir} (J/mol K)	Cp_t (J/mol K)	Cp JANAF
100.00	10.812	45.236					
200.00	17.588	73.589					
298.15	23.160	96.900					
300.00	23.253	97.289	0.0725	6.643	9.414	106.703	106.809
400.00	27.660	115.729	0.0628	4.982	9.76	125.49	125.545
500.00	30.954	129.510	0.0561	3.986	9.506	139.01	139.139
600.00	33.356	139.561	0.0521	3.322	9.109	148.67	148.708
700.00	35.105	146.879					
800.00	36.393	152.268	0.0444	2.49	8.03	160.3	160.31
900.00	37.357	156.302					
1000.00	38.092	159.377	0.0397	1.993	7.037	166.414	166.486
1100.00	38.662	161.761					
1200.00	39.111	163.641					
1300.00	39.471	165.145					
1400.00	39.762	166.365					
1500.00	40.002	167.368	0.0324	1.33	5.742	173.110	173.114
1600.00	40.201	168.200					
1700.00	40.368	168.898					
1800.00	40.509	169.489					
1900.00	40.629	169.994					
2000.00	40.733	170.427	0.028	0.996	5.104	175.53	175.548
2500.00	41.084	171.894					
3000.00	41.278	172.705					
3500.00	41.396	173.199					
4000.00	41.473	173.521					
4500.00	41.526	173.743					
5000.00	41.564	173.903	0.0178	0.3986	4.322	178.22	178.155
5500.00	41.592	174.021					
6000.00	41.613	174.111					

BIPHENYL: BURCAT, ZELEZNIK, McBride: NASA TECH MEMO# 83800: JAN 1985.

NON - LINEAR MOLECULE
Cpinf (Kcal/mol) = 121.207
NUMBER OF OSCILLATORS = 59
Arithmetic mean frequency = 1308.508 (1/cm)
Geometric mean frequency = 1038.418 (1/cm)
Root Mean Square frequency = 1575.540 (1/cm)
Mean Square frequency = 2.482E+06 (1/cm^2)
Squre of Mean frequency = 1.712E+06 (1/cm^2)
modified frequency dispersion parameter = 1.425E+00

TEMP(K)	Cp(cal/mol K)	Cp(J/mol K)	1/Qf	V/RT	Cp_{ir} (J/mol K)	Cp_t (J/mol K)
100.00	12.869	53.843				
200.00	24.017	100.489				
298.15	37.752	157.953				
300.00	38.016	159.059	0.0488	2.516	8.02	167.08
400.00	51.612	215.944	0.0423	1.887	6.85	222.79
500.00	63.049	263.799	0.038	1.51	6.067	269.87
600.00	72.232	302.220	0.0345	1.258	5.012	307.23
700.00	79.608	333.079				
800.00	85.622	358.241	0.0299	0.944	4.725	362.97
900.00	90.602	379.078				
1000.00	94.776	396.542	0.0267	0.755	4.72	401.26
1100.00	98.305	411.308				
1200.00	101.308	423.874				
1300.00	103.878	434.627				
1400.00	106.088	443.873				
1500.00	107.997	451.859	0.0218	0.503	4.42	456.28
1600.00	109.653	458.787				
1700.00	111.095	464.823				
1800.00	112.357	470.104				
1900.00	113.466	474.743				
2000.00	114.444	478.834	0.0189	0.377	4.286	483.12
2500.00	117.932	493.427				
3000.00	119.991	502.042				
3500.00	121.295	507.498				
4000.00	122.168	511.153				
4500.00	122.780	513.711				
5000.00	123.224	515.569	0.012	0.151	4.088	519.66
5500.00	123.556	516.958				
6000.00	123.811	518.024				

```
             PROGRAM CPIDEAL
***********************************************************************
*      THIS PROGRAM CALCULATES THE VIBRATIONAL CONTRIBUTION TO Cp     *
*      GIVEN VIBRATIONAL FREQUENCIES FOR EACH OSCILATOR               *
*                                                                     *
*      WRITTEN BY:   EDWARD RITTER      11/26/88                      *
***********************************************************************
             IMPLICIT REAL*8 (A-H,O-Z)
             INTEGER NS, ND
             DIMENSION FREQ(100),T(30),CVI(100),THETA(100),GI(100)
             REAL*8 CP,CVV,MS,RMS,SMS
             nrot=0
             T(1)=100.
             T(2)=200.
             T(3)=298.15
             T(4)=300.0
             T(5)=400.0
             T(6)=500.0
             T(7)=600.
        T(8)=700.
             T(9)=800.0
             T(10)=900.
             T(11)=1000.0
             T(12)=1100.
             T(13)=1200.
             T(14)=1300.
             T(15)=1400.
             T(16)=1500.0
             T(17)=1600.
             T(18)=1700.
             T(19)=1800.
             T(20)=1900.
             T(21)=2000.0
             T(22)=2500.
             T(23)=3000.0
             T(24)=3500.
             T(25)=4000.
             T(26)=4500.
             T(27)=5000.
             T(28)=5500.
             T(29)=6000.0
             CALL INPUT(NS,ND,GI,FREQ)
             CALL CLS
             WRITE(*,*)'  enter :  0 - non-linear molecule'
             write(*,*)'           1 - linear molecule'
             read(*,*)lin
             call lines(30,2)
             if(lin.eq.0)then
                          write(30,*)'    NON - LINEAR MOLECULE '
             else
                          WRITE(30,*)'          LINEAR MOLECULE'
             endif
             IF(LIN.EQ.0)THEN
                          write(*,*)' enter # of free rotors in this molecule '
```

```
                        read(*,*)nrot
                        NAT=(NS+6+nrot)/3
                        CPINF=(3.0*FLOAT(NAT)-(2.0+(nrot/2.)))*1.987
            ELSE
                        NAT=(NS+5)/3
                        CPINF=(3.0*FLOAT(NAT)-(3./2.))*1.987
            ENDIF
            WRITE(30,*)'  Cpinf (Kcal/mol) = ',CPINF
            FARM=0.0
            FGM=0.0
            SS=0.0
            SQM=0.0
            DO 1999 I1=1,ND
            SS=SS+(GI(I1)*(FREQ(I1)**2))
            FGM=FGM+(GI(I1)*LOG(FREQ(I1)))
            FARM=FARM+(GI(I1)*FREQ(I1))
1999        CONTINUE
            MS=SS/FLOAT(NS)
            FARM=FARM/FLOAT(NS)
            FARMSQ=FARM**2
            BETA=(FLOAT(NS-1)/FLOAT(NS))*(MS/FARMSQ)
            FGM=EXP(FGM/FLOAT(NS))
            RMS=MS**0.5
            WRITE(30,*)' NUMBER OF OSCILLATORS = ',NS
            WRITE(30,17000)FARM
17000       FORMAT(1X,5X,'Arithmetic mean frequency = ',f10.3,' ( 1/cm )')
            WRITE(30,17001)FGM
17001   format(1x,5x,'Geometric mean frequency = ',f10.3,' ( 1/cm )')
            WRITE(30,17004)RMS
17004  FORMAT(6X,'Root Mean Square frequency = ',f10.3,' ( 1/cm )')
            WRITE(30,17002)MS
17002  format(6x,'Mean Square frequency = ',1pe15.3,' ( 1/cm^2 )')
            WRITE(30,17003)FARMSQ
17003  format(6x,'Squre of Mean frequency = ',1pe15.3,' ( 1/cm^2 )')
            WRITE(30,17005)BETA
17005  format(6x,'modified frequency dispersion parameter = ', 1pe15.3)
            WRITE(*,17000)FARM
            WRITE(*,17001)FGM
            WRITE(*,17004)RMS
            WRITE(*,17002)MS
            WRITE(*,17003)FARMSQ
            WRITE(*,17005)BETA
            WRITE(30,7003)
10          CONTINUE

            call lines(30,1)
            call cls
            WRITE(*,*)'  enter :  1 - default temp range ( 300 - infin K)'
            write(*,*)'           2 - enter a specific temperature'
            write(*,*)'           3 - end program'
            read(*,*)iop
            if(iop.eq.1)then
                        it=1
                        nt=29
```

```
          elseif(iop.eq.2)then
                    call cls
                    write(*,*)'   enter Temperature (K)'
                    read(*,*)T(1)
                    it=1
                    nt=1
          else
                    goto 301
          endif
          DO 300 J=IT,NT
          CVV=0.0
          CP=0.0
          CPCAL=0.0
          CPJ=0.0
          CALL LINES(20,3)
          WRITE(20,7000)T(J)
          WRITE(20,7001)
          DO 100 I=1,ND
          CALL CVVIB(FREQ(I),T(J),CVI(I),THETA(I))
          CVCONT=(CVI(I)*GI(I))
          CVV=CVV+CVCONT
          WRITE(20,7002)I,FREQ(I),THETA(I),GI(I),CVCONT
100       CONTINUE
          WRITE(20,7005)T(J),CVV
          if(lin.eq.0)then
                    CP=CVV+4.0+(nrot/2.)
          else
                    CP=CVV+(7./2.)
          endif
          CPCAL=CP*1.987
          CPJ=CPCAL*4.184
          if(lin.eq.0)then
                    WRITE(20,7007)
          else
                    write(20,7008)
          endif
          WRITE(20,7006)CP
          WRITE(30,7004)T(J),CPCAL,CPJ
300       CONTINUE
          WRITE(30,7004)99999.,CPINF,CPINF*4.184
          if(iop.eq.2)goto 10
301       continue
          CLOSE(20,STATUS='KEEP')
          CLOSE(30,STATUS='KEEP')
          WRITE(*,17000)FARM
          WRITE(*,17001)FGM
          WRITE(*,*)' NUMBER OF OSCILLATORS = ',NS
          STOP '    have a nice day !'
7000      FORMAT(1X,' VIBRATIONAL CONTRIBUTION TO Cv AT ',F7.1,' deg K')
7001      FORMAT(1X,' i     v (1/cm)     THETAi     gi    Cv,vib i ')

7002  FORMAT(1X,I3,5X,F8.3,5X,F8.2,5X,F6.2,5X,F8.3)

7003      FORMAT(1X,' TEMP(K)  Cp(cal/mol K)  Cp(J/mol K)')
```

```
7004    FORMAT(2X,F8.2,5X,F8.3,6X,F8.3)
7005        FORMAT(1X,9x,' T = ',F8.2,' K',3X,'Cv,vib = ',F5.2,' * R')
7006    FORMAT(1X,28X,'Cp,tot = ',F5.2,' * R')
7007        FORMAT(3X,'Trans and rot + convert Cv-> Cp =  4.00 * R')
7008    FORMAT(3X,'Trans and rot + convert Cv-> Cp =  7/2 * R')
            END



            SUBROUTINE INPUT(NS,ND,GI,FREQ)
            IMPLICIT REAL*8(A-H,O-Z)
            INTEGER NS,ND
            CHARACTER*70 SPEC,FN,FIN,FOUT,FTBL,spec2
            DIMENSION FREQ(100),GI(100)
            CALL CLS
            WRITE(*,*)'ENTER INPUT FILENAME'
            READ(*,70)FN
70          FORMAT(A70)
71          FORMAT(1X,A70)
            FIN=FN
            FOUT=FN
            FTBL=FN
            CALL FNAME(FIN,'INP')
            CALL FNAME(FOUT,'OUT')
            CALL FNAME(FTBL,'TBL')
            OPEN(10,FILE=FIN,STATUS='OLD')
            read(10,70)spec2
            if(spec2(:5).eq.'SHORT')FOUT='NUL'
            OPEN(20,FILE=FOUT,STATUS='UNKNOWN')
            OPEN(30,FILE=FTBL,STATUS='UNKNOWN')
            READ(10,70)SPEC
            WRITE(20,71)SPEC
            WRITE(30,71)SPEC
            CALL LINES(20,3)

            I=0
            S=0.
            NS=0
            ND=0
100         CONTINUE
            READ(10,*,END=200)FNUIN,DEGEN
            I=I+1
            FREQ(I)=FNUIN
            GI(I)=DEGEN
            S=S+DEGEN
            ND=I
            GOTO 100
200         CONTINUE
            NS=IDINT(S+.1)
            write(20,7001)
7001        format(5x,' FREQ(i)',3x,'gi')
            DO 300 I=1,ND
            WRITE(20,7000)FREQ(I),GI(I)
7000        FORMAT(5X,F8.3,2X,'(',F8.3,')')
```

```
300         CONTINUE
            CALL LINES(20,2)
            WRITE(20,7002)NS
7002        FORMAT(1X,' number of vibrational modes = ',I3)
            call lines(20,2)
            WRITE(30,*)NS
            CALL LINES(30,3)
            return
            end


            SUBROUTINE CVVIB(FREQ,T,CV,THETA)
            IMPLICIT REAL*8(A-H,O-Z)
            REAL*8 HP,KB,C,NAVAG
            HP=6.6256E-34
            KB=1.3805E-23
            C=3.0E+10
            NAVAG=6.023E+23
*
            THETA=((HP*C)/KB)*FREQ
            X=THETA/T
            TOP=EXP(X)
            BOTTOM=(EXP(X)-1)**2
            CV=(X**2)*(TOP/BOTTOM)
            RETURN
            END
```

APPENDIX IV - F

FORTRAN SOURCE CODE LISTING FOR

CPFIT

EXAMPLE INPUT FILE FOR CPFIT

```
 .
 .
 .        C2H4O   : DATA FROM THERM
 .
 .   REGRESSION OF Cp DATA vs. T TO A 5 PARAMETER HARMONIC OSCILLATOR
 .                SERIES ( 3 TERMS)     : C2H4O
 .

REPORT EVERY # ITER:    MAX-ITER:           CONV( 1 - 5 )
      100,                75,      1.0E-3, 1.0E-3, 0.05, 0.0, 0.0
R CONVERSION:   # VIB MODES:   GEOMETRY  0.(NON-LIN) 1.(LIN)
   4.184            15.                    0.
   300.00       48.45
   400.00       64.14
   500.00       75.56
   600.00       85.10
   800.00       102.09
  1000.00       114.47
```

EXAMPLE OUTPUT FILE FROM CPFIT

```
 .
 .
 .        C2H4O   : DATA FROM THERM
 .
 .   REGRESSION OF Cp DATA vs. T TO A 5 PARAMETER HARMONIC OSCILLATOR
 .                SERIES ( 3 TERMS)     : C2H4O
 .


    NUMBER OF PARAMETERS            5
    MAXIMUM NO. OF ITERATIONS      75
    NUMBER OF DATA POINTS           7
    FREQUENCY OF PRINTING         100

 COEFFICIENT      STARTING         LOWER           UPPER
   NUMBER           VALUE          LIMIT           LIMIT

      1        4.50000018E+00  0.00000000E+00  1.50000000E+01
      2        6.00000009E+00  0.00000000E+00  1.50000000E+01
      3        6.00000000E+02  3.50000000E+02  9.99000000E+02
      4        1.20000000E+03  9.00000000E+02  1.99900000E+03
      5        2.40000000E+03  1.50000000E+03  3.60000000E+03


 .
 .
 .        C2H4O   : DATA FROM THERM
 .
 .   REGRESSION OF Cp DATA vs. T TO A 5 PARAMETER HARMONIC OSCILLATOR
 .                SERIES ( 3 TERMS)     : C2H4O
 .


F I N A L  R E S U L T S      8 ITERATIONS PERFORMED

SUM OF SQUARES =    1.47886716E+00

      B( 1)  =   4.5332389E+00
      B( 2)  =   4.5309800E+00
      B( 3)  =   9.7201365E+02
      B( 4)  =   9.7201365E+02
      B( 5)  =   2.6244423E+03

         Cp = 4.0 + C vib   (non-linear molecule)

                  n
                ______
                \
                 \                        2       exp(A(i+n)*(hc/kT))
       C vib =   /    A(i)*[A(i+n)*(hc/kT)]  .  __________________________
                /_____                                                   2
                        i                         [1 - exp(A(i+n)*hc/kT)]
   Cvvib: sum of  3 terms


coef: A(i)          exp:    A(i+n)

4.533E+00             9.720E+02

4.531E+00             9.720E+02

5.936E+00             2.624E+03
X             Y              YC            YC-Y          PCT


 3.000E+02    4.845E+01    4.905E+01    6.008E-01        1.240
 4.000E+02    6.414E+01    6.329E+01   -8.469E-01       -1.320
 5.000E+02    7.556E+01    7.551E+01   -5.066E-02        -.067
 6.000E+02    8.510E+01    8.571E+01    6.087E-01         .715
 8.000E+02    1.021E+02    1.020E+02   -8.404E-02        -.082
 1.000E+03    1.145E+02    1.143E+02   -1.433E-01        -.125
 1.000E+05    1.580E+02    1.580E+02   -7.092E-03        -.004
0    CONSTANTS :  4.18400D+00 1.50000D+01 0.00000D+00

STANDARD ERROR OF ESTIMATE =   4.596376E-01
ABSOLUTE AVERAGE PERCENT ERROR =      .51
RESIDUAL DEGREES OF FREEDOM =   2
RESIDUAL MEAN SQUARE =   7.394336E-01
RESIDUAL ROOT MEAN SQUARE =   8.599032E-01
 NUMBER OF OSCILLATORS =        15.000000000000000

  ESTIMATED ARITHMETIC MEAN FREQUENCY
      1625.911


  ESTIMATED GEOMETRIC MEAN FREQUENCY
      1440.030

     Root Mean Square frequency =   1815.633 ( 1/cm )
     Mean Square frequency =      3.297E+06 ( 1/cm^2 )
     Square of Mean frequency =      2.644E+06 ( 1/cm^2 )
     modified frequency dispersion parameter =      1.164E+00
```

```
      PROGRAM CPFIT
$LARGE
             IMPLICIT REAL*8(A-H,O-Z)
      CHARACTER*8 FMT1
      CHARACTER*39 FMT2
      CHARACTER*33 FMT3A(5)
      CHARACTER*44 FMT3B
      CHARACTER*21 FMT4A(5)
      CHARACTER*13 FMT4B
      CHARACTER*1 PLTCHR,DIGIT(10)
      CHARACTER*72 TITLE(20),TEXT
      CHARACTER*4 BLANK,TEST
      CHARACTER*72 FIN, FOUT
      LOGICAL DONE, NEG
      REAL*8 NORML, MEAN
      DIMENSION B(20),BMIN(20),BMAX(20),Y(400),Z(400),PDERIV(20,400),
     @          BI(20),ZI(400),CONV(5),X(10,400),NORMAL(21,21),WORK(21),
     @          SCALE(20),GRAD(20),DIAG(21),CONST(40),VAR(6),STD(6),
     @          IPIVOT(20), COVAR(20,20), NORML(21,21), DIAG2(20),
     @          IX(800), XTEMP(8000),DELPCT(20),BS(20),BSMIN(20),
     @          BSMAX(20)
      DOUBLE PRECISION SUMSQ,B,BMIN,BMAX,BI,NORMAL,GRAD,WORK,DIAG,CONST
     @                ,SCALE,BTEMP,MLARGE,PLARGE,AVG,COEF2,COEF3,
     @                BS,BSMIN,BSMAX,BTMP
      INTEGER PARM,DATA,ERROR,ERROR2,INPUT,OUTPUT,
     %        OPTION,FREQ,MAXPAR,MAXIND,
     %        MAXDAT,NTITLE,MAXIT,NIND,OPT2,INTMSK,
     %        PRTOPT,PLTOPT,MAXCON,BFLAG
      EQUIVALENCE (IPIVOT(1), SCALE(1)), (COVAR(1,1), PDERIV(1,1)),
     @            (NORML(1,1), NORMAL(1,1)), (DIAG2(1), DIAG(1)),
     @            (IOVER1, OVERF1), (IOVER2, OVERF2),
     @            (IX(1), PDERIV(1,1)), (XTEMP(1), PDERIV(1,1))
      COMMON /DINFO/ IPARM1, IDATA, IPARM, INDPT
      COMMON /CONST/ CONST,NTERMS,NROT
      COMMON /PDV/ PDERIV
      DATA INPUT/5/,OUTPUT/6/,BLANK/'    '/,MAXPAR/20/,MAXIND/10/
      DATA MAXDAT/400/,MAXCON/40/
      DATA MLARGE/-1.6D308/,PLARGE/1.6D308/
      DATA FMT1/'(8F10.5)'/,DONE/.TRUE./
      DATA FMT2/'(3X  /(9H       B(,I2,4H)  =,1PE16.7))'/
      DATA FMT3A/'(3X  /8H       ,1(1HX,I1,12X)\)',
     &           '(3X  /8H       ,2(1HX,I1,12X)\)',
     &           '(3X  /8H       ,3(1HX,I1,12X)\)',
     &           '(3X  /8H       ,4(1HX,I1,12X)\)',
     &           '(3X  /8H       ,5(1HX,I1,12X)\)'/
      DATA FMT3B/'(2HX,13X,2HY,13X,2HYC,11X,4HYC-Y,10X,3HPCT//) '/
      DATA FMT4A/'(5H    ,4(1PE14.5)\)','(5H    ,5(1PE14.5)\)',
     &           '(5H    ,6(1PE14.5)\)','(5H    ,7(1PE14.5)\)',
     &           '(5H    ,8(1PE14.5)\)'/, FMT4B/'(1H ,0PF10.2)'/
      DATA DIGIT/'0','1','2','3','4','5','6','7','8','9'/,PLTCHR/'O'/
      WRITE(*,693)
      READ(*,1) FIN
      OPEN(5,FILE=FIN,STATUS='OLD')
      WRITE(*,694)
      READ(*,1) FOUT
      OPEN(6,FILE=FOUT,STATUS='UNKNOWN')
  693 FORMAT(' Enter the filename for the data points       :')
  694 FORMAT(' Enter the filename for the output            :')
             WRITE(*,*)'ENTER # OF HIGH TEMP FREE ROTORS'
             READ(*,*)NROT
             NREPEAT=10
             RMXPCT=3.75
  695 FORMAT(A10)
       CALL RESET
      IPARM = MAXPAR
      IDATA = MAXDAT
      IPARM1 = MAXPAR + 1
      INDPT = MAXIND
100   NTITLE = 1
110   READ (INPUT,1,END=900) TITLE(NTITLE)
1     FORMAT(A72)
      IF (TITLE(NTITLE).EQ.' ')   GO TO 120
      IF (NTITLE .EQ. 9)    GO TO 500
      NTITLE = NTITLE + 1
      GO TO 110
120          CONTINUE
             READ(INPUT,1,END=900)TEXT
             READ(INPUT,*)FREQ,MAXIT,CONV(1),CONV(2),CONV(3),CONV(4),CONV(5)
*            DEFINE  DEFAULTS
             IREPNUM=0
             IREPEAT=0
             option=1
             opt2=1
             nprt=1
             prtopt=1
             pltopt=1
             icall=0
             parm=5
             nterms=3
             nind=1
             ISTART=1
8     FORMAT(3D10.8)
             iconst=3
             fvalue=0.
             tvalue=0.
32    FORMAT(7E10.0,8X,I2)
11    FORMAT(8D10.0)
             READ(INPUT,1,END=900)TEXT
      READ (INPUT,*) (CONST(I), I = 1, ICONST)
*            CONST(1): GAS CONSTANT CONVERSION FACTOR;   CONST(1)=1.0 (CAL)
*                                                   CONST(1)=4.184 (J)
*        CONST(2) = TOTAL NUMBER OF VIBRATIONAL DEGREES OF FREEDOM
*
*            CONST(3) = 0., IF NON-LINEAR MOLECULE; = 1., IF LINEAR
*

170   NIND1 = NIND + 1
C
```

```
C            READ IN THE DATA POINTS.
C
      DATA = 1
180   READ (INPUT,*,END=190) X(1,DATA),Y(DATA)
      DATA = DATA + 1
      IF (DATA .GT. MAXDAT)  GO TO 565
      GO TO 180
190   CONTINUE
195   IF (FREQ .LE. 0)       FREQ = 1
      IF (MAXIT .LE. 0)       GO TO 530
      IF (PARM .LE. 0)        GO TO 520
      IF (NIND .LE. 0)        GO TO 540
      IF (PARM .GT. MAXPAR)   GO TO 550
      IF (NIND .GT. MAXIND)   GO TO 560
*
*    DETERMINE TYPE OF MOLECULE:  LOW, MID, HIGH FREQUENCY
*    FROM CP(300)/CONST(2) : HEAT CAPACITY PER OSCILLATOR
*           FIRST DATA POINT MUST BE CP(300)
            IF(CONST(3).EQ.0.)THEN
*                       non-linear molecule

                        RNATOM=((CONST(2)+6.0)/3.0)
                        CPINF=(3.0*RNATOM-(2.+(NROT/4.)))*1.987*CONST(1)
*           empirically determined correction for # of rotors.
*    theory gives nrot/2  but nrot/4 works better.
*                       CPINF=( 3.0 * NATOM - (2.0+(NROT/2.)))*1.987*CONST(1)*
                        CONST(2)=CONST(2)-float(NROT/2.)
            ELSE
*                       linear molecule
                        NATOM=IDINT((CONST(2)+5.)/3.)
                        CPINF=(3.0* NATOM - (3./2.))*1.987*CONST(1)
            ENDIF
**
**                                  set last data point equal to cpinf
**                                  at last temperature = 99999. K
**
                                    X(1,DATA)=99999.
                                    Y(DATA)=CPINF
**                      continue test for frequency range
*
            BFLAG=0
*           CPRATIO=Y(1)/CPINF
            CPRATIO=0.5
9268        CONTINUE
            IF(IREPEAT.EQ.1)THEN
                        IREPNUM=IREPNUM+1
                        IREPEAT=0
                        IF(IREPNUM.GE.NREPEAT)THEN
                                    STOP ' FIT PROCEDURE FAILED '
                        ENDIF
                        IF(IREPNUM.GT.1)THEN
                                    IBS=0
                                    DO 9655 I=1,5
                        if(bs(i).EQ.0.0)goto 9655
```

```
                                DELTAB=(ABS(BS(I)-B(I)))/BS(I)
                                IF(DELTAB.GT.0.01)IBS=1
9655                            CONTINUE
                                IF(IBS.EQ.0)THEN
                                          IREPNUM=NREPEAT
                                          IREPEAT=2
                                          GOTO 600
                                ENDIF
                      ENDIF
                      IF(DELPCT(1).LT.0.0.AND.IREPNUM.EQ.1)THEN
                                CPRATIO=0.75
                                DO 9651 I=1,5
                                BS(I)=B(I)
                                BSMIN(I)=BMIN(I)
                                BSMAX(I)=BMAX(I)
9651                            CONTINUE
                                BFLAG=0
                      ELSE IF(DELPCT(1).GT.0.0.AND.IREPNUM.EQ.1)THEN
                                CPRATIO=0.25
                                DO 9652 I=1,5
                                BS(I)=B(I)
                                BSMIN(I)=BMIN(I)
                                BSMAX(I)=BMAX(I)
9652                            CONTINUE
                                BFLAG=1
                      ELSEIF((DELPCT(1).LT.0.0.AND.DELPCT(2).LT.0.0)
     +              .AND.IREPNUM.GT.1)THEN
                                IF(ABS(B(3)-BMIN(3)).LE.0.1)THEN
                                          BMIN(3)=BMIN(3)-50.
                                ELSE
                                          IF(BFLAG.EQ.1)THEN
                                                    DO 9653 I=1,5
                                                    BTMP=B(I)
                                                    B(I)=(B(I)+BS(I))/2.0
                                                    BS(I)=BTMP
                                          IF(BMAX(I).LT.BSMAX(I))BMAX(I)=BSMAX(I)
                                          IF(BMIN(I).GT.BSMIN(I))BMIN(I)=BSMIN(I)
9653                                      CONTINUE
                                          ENDIF
                                ENDIF
                                BFLAG=0
                                GOTO 9238
                      ELSEIF((DELPCT(1).GT.0.0.AND.DELPCT(2).GT.0.0)
     +              .AND.IREPNUM.GT.1)THEN
                                IF(ABS(B(3)-BMIN(3)).LE.0.1)THEN
                                          BMIN(3)=BMIN(3)+50.
                                          B(3)=BMIN(3)+50.
                                ELSE
                                          IF(BFLAG.EQ.0)THEN
                                                    DO 9654 I=1,5
                                                    BTMP=B(I)
                                                    B(I)=(B(I)+BS(I))/2.0
                                                    BS(I)=BTMP
                                          IF(BMAX(I).LT.BSMAX(I))BMAX(I)=BSMAX(I)
```

```
                                                     IF(BMIN(I).GT.BSMIN(I))BMIN(I)=BSMIN(I)
9654                                                 CONTINUE
                                                     BLFAG=1
                                                     ELSE
                                                                   BMIN(1)=0.
                                                                   BMIN(2)=0.
                                                                   BMIN(3)=100.
                                                                   BMIN(4)=300.
                                                                   BMIN(5)=700.
                                                                   BMAX(1)=CONST(2)
                                                                   BMAX(2)=CONST(2)
                                                                   BMAX(3)=999.
                                                                   BMAX(4)=1999.
                                                                   BMAX(5)=3600.
                                                     ENDIF

                                        ENDIF
                                        GOTO 9238
                          ENDIF

             ENDIF
*            set initial parameter values, max & min
*            RANGE DETERMINED BY CPRATIO
*                         CPRATIO < 0.4  :  VERY HIGH FREQUENCY WEIGHTED
*              0.4 < CPRATIO < 0.65 :  MID FREQUENCY WEIGHTED
*                    CPRATIO > 0.65 :  LOW FREQUENCY WEIGHTED
             IF(CPRATIO.LT.0.4)THEN
*                         starting point : high frequency weighted
                          B(1)=0.2*CONST(2)
                          B(2)=0.5*CONST(2)
                          B(3)=600.
                          B(4)=1500.
                          B(5)=3000.
                          BMIN(1)=0.
                          BMIN(2)=0.
                          BMIN(3)=350.
                          BMIN(4)=900.
                          BMIN(5)=1500.
                          BMAX(1)=0.6*CONST(2)
                          BMAX(2)=CONST(2)
                          BMAX(3)=1500.
                          BMAX(4)=2500.
                          BMAX(5)=3600.
             ELSE IF(CPRATIO.GE.0.4.AND.CPRATIO.LE.0.63)THEN
*                         starting point : spread over range
                          B(1)=0.3*CONST(2)
                          B(2)=0.4*CONST(2)
                          B(3)=600.
                          B(4)=1200.
                          B(5)=2400.
                          BMIN(1)=0.
                          BMIN(2)=0.
                          BMIN(3)=350.
                          BMIN(4)=900.
```

```
                           BMIN(5)=1500.
                           BMAX(1)=CONST(2)
                           BMAX(2)=CONST(2)
                           BMAX(3)=999.
                           BMAX(4)=1999.
                           BMAX(5)=3600.
              ELSE IF(CPRATIO.GT.0.63)THEN
*                          starting point: low frequency weighted
*                          parameters are constrained to low frequencies;
*                          with lower minimum frequencies acceptable.
                           B(1)=0.5*CONST(2)
                           B(2)=0.4*CONST(2)
                           B(3)=300.
                           B(4)=500.
                           B(5)=700.
                           BMIN(1)=CONST(2)/10.
                           BMIN(2)=CONST(2)/10.
                           BMIN(3)=100.
                           BMIN(4)=401.
                           BMIN(5)=601.
                           BMAX(1)=CONST(2)
                           BMAX(2)=CONST(2)
                           BMAX(3)=450.
                           BMAX(4)=1000.
                           BMAX(5)=2000.
             ENDIF
9238         CONTINUE

      WRITE (OUTPUT,13) (TITLE(J),J=1,NTITLE)
*      WRITE (OUTPUT,23) FMT1
      WRITE (OUTPUT,14) PARM,MAXIT,DATA,FREQ,(J,B(J),BMIN(J),BMAX(J),
     ?                  J=1,PARM )
13    FORMAT (/,10(' ',A72/))
23    FORMAT (/' ',15X,'INPUT DATA'//'    FORMAT FOR DATA POINTS'
     *        ,5X,A8)
14    FORMAT ('    NUMBER OF PARAMETERS',10X,I4/
     #         '    MAXIMUM NO. OF ITERATIONS',5X,I4/
     #         '    NUMBER OF DATA POINTS',9X,I4/
     #         '    FREQUENCY OF PRINTING',9X,I4//' COEFFICIENT',
     #             6X,'STARTING',9X,'LOWER',11X,'UPPER'/'   NUMBER',
     #             10X,'VALUE',2(11X,'LIMIT')//30(I7,7X,3(1PE16.8)/))
9865         continue
             IREDUCE=0
      DO 200 ITER = ISTART,MAXIT
         CALL SOLVE (PARM,DATA,ERROR,ERROR2,B,BMIN,BMAX,Y,Z,
     &               SUMSQ,CONV,NORMAL,SCALE,
     &               WORK,GRAD,BI,ZI,DIAG,X )
             IF(SUMSQ.LT.CONV(1)) GOTO 600
C
C             CHECK FOR TERMINAL ERROR CODES FROM SUBROUTINE
C             SOLVE.
C
         IF (ERROR .EQ. -1)   GO TO 570
         IF (ERROR .EQ. -2)   GO TO 590
         IF (ERROR .EQ. -3)   GO TO 580
         IF (ERROR .EQ. 0)    GO TO 600
         IF (ITER/FREQ*FREQ .NE. ITER)      GO TO 20001
         WRITE (OUTPUT,18) ITER,SUMSQ
18       FORMAT('0'/' ITERATION',I4/'0SUM OF SQUARES',1PE17.8)
         WRITE (OUTPUT,FMT2) (J,B(J),J=1,PARM)

*
*       CHECK TO SEE IF TWO FREQUENCIES ARE APPROACHING ONE ANOTHER.
*

20001       continue

200      CONTINUE
      WRITE (OUTPUT,19)
19    FORMAT (/' MAXIMUM NUMBER OF ITERATIONS EXCEEDED')
      ITER=MAXIT
      GO TO 600
C
C             ERROR GENERATION
C
500   WRITE (OUTPUT,2)
2     FORMAT(/' MAXIMUM NUMBER OF TITLE CARDS EXCEEDED, ONLY ',
     $        'THE FIRST TEN WILL BE RETAINED')
510   READ (INPUT,3) TEST
3     FORMAT (72X,A4)
      IF (TEST .EQ. BLANK)   GO TO 510
      GO TO 120
515   WRITE(OUTPUT,49) MAXCON
49    FORMAT(/' NUMBER OF CONSTANTS SPECIFIED EXCEEDS MAXIMUM (',
     &       I4,') ALLOWED.')
520   WRITE (OUTPUT,5)
5     FORMAT(/' NUMBER OF PARAMETERS SPECIFIED LESS THAN OR ',
     $        'EQUAL TO ZERO')
      GO TO 900
530   WRITE (OUTPUT,6)
6     FORMAT (/' MAXIMUM NUMBER OF ITERATIONS SPECIFIED LESS ',
     %        'THAN OR EQUAL TO 0')
      GO TO 900
540   WRITE (OUTPUT,7)
7     FORMAT (/' NUMBER OF INDEPENDENT VARIABLES SPECIFIED ',
     *        'IS LESS THAN OR EQUAL TO 0')
      GO TO 900
550   WRITE (OUTPUT,9)
9     FORMAT (/' NUMBER OF PARAMETERS EXCEEDS MAXIMUM')
      GO TO 900
560   WRITE (OUTPUT,10)
10    FORMAT (/' NUMBER OF INDEPENDENT VARIABLES EXCEEDS MAXIMUM')
      GO TO 900
565   WRITE (OUTPUT,24)
24    FORMAT(/' MAXIMUM NUMBER OF DATA POINTS EXCEEDED')
      GO TO 900
570   WRITE (OUTPUT,15)
15    FORMAT(/' NUMBER OF PARAMETERS EXCEEDS NUMBER OF DATA POINTS')
```

```
      GO TO 700
580   WRITE (OUTPUT,16)
16    FORMAT (/' INITIAL B VALUES NOT BETWEEN BMIN AND BMAX')
      GO TO 700
590   WRITE (OUTPUT,17) ERROR2
17    FORMAT (/' SINGULAR MATRIX ENCOUNTERED IN SUBROUTINE SOLVE, ',
     %       'DIAGONAL ELEMENT ',I3)
      GO TO 700
C
C            SOLUTION FOUND
C
600         CONTINUE
      WRITE (OUTPUT,13) (TITLE(J),J=1,NTITLE)
      WRITE (OUTPUT,20) ITER
20    FORMAT (' F I N A L   R E S U L T S',I8,' ITERATIONS PERFORMED')
      WRITE (OUTPUT, 45) SUMSQ
45    FORMAT (/' SUM OF SQUARES = ',1PE17.8)
      WRITE (OUTPUT,FMT2) (J,B(J),J=1,PARM)
            IF(CONST(3).EQ.0)THEN
                        write(output,8990)
            ELSE
                        write(output,8991)
            ENDIF
            write(output,8992)
            write(output,8993)
            write(output,8994)
            write(output,8995)
            write(output,8996)
            write(output,8997)
            write(output,8998)
            WRITE(OUTPUT,*)' '
            WRITE(OUTPUT,8764)NTERMS
8764        FORMAT(1X,/,5X,'Cvvib: sum of ',i2,' terms'/)
                        WRITE(OUTPUT,*)' '
                        write(output,8763)
            IF(NTERMS.EQ.1)THEN
                        WRITE(OUTPUT,8765)B(1),B(2)
            ELSE IF(NTERMS.EQ.2)THEN
                        COEF2=CONST(2)-B(1)
                        WRITE(OUTPUT,8765)B(1),B(2)
                        WRITE(OUTPUT,8765)COEF2,B(3)
            ELSE IF(NTERMS.EQ.3)THEN
                        COEF3=CONST(2)-(B(1)+B(2))
                        WRITE(OUTPUT,8765)B(1),B(3)
                        WRITE(OUTPUT,8765)B(2),B(4)
                        WRITE(OUTPUT,8765)COEF3,B(5)
            ENDIF
8763        format(1x,'coef: A(i)          exp:    A(i+n)')
8765        format(1x,/,1PE10.3,'            ',1PE10.3)
8990        FORMAT(1X,/,'          Cp = 4.0 + C vib   (non-linear molecule)'
     +/)
8991        format(1x,/,'          Cp = 7/2 + C vib   (    linear molecule)'
     +/)
8992   format(1x,'                 n')
```

```
8993    format(1x,'               _______')
8994    format(1x,'               \        ')
8995    format(1x,'                \                          2       ex
     +p(A(i+n)*(hc/kT))')
8996    format(1x,'      C vib =  /     A(i)*[A(i+n)*(hc/kT)]  .  _____
     _______________________')
8997    format(1x,'               /_______
     +                   2')
8998    format(1x,24x,'i',24x,'[1 - exp(A(i+n)*hc/kT)]'//)
      IF (NPRT .LT. NIND)  NPRT = NIND
      IF (NPRT .GT. 5)      NPRT = 5
      WRITE (OUTPUT,9987)
9987  FORMAT(1X,'X',13X,'Y',13X,'YC',11X,'YC-Y',10X,'PCT',//)
      SUMP = 0.0
      DO 630 I = 1,DATA
         ZI(I) = Z(I) - Y(I)
         IF (Y(I) .EQ. 0.0)  GO TO 620
         PCT = 100.0 * ZI(I) / Y(I)
         GO TO 625
620      PCT = 0.0
625          CONTINUE
             IF(ABS(PCT).GT.3.75.AND.IREPEAT.NE.2)THEN
                           IREPEAT=1
                           DELPCT(I)=PCT
             ENDIF
         SUMP = SUMP + ABS(PCT)
         WRITE (OUTPUT,'(1X,4(1PE10.3,3X),0PF9.3)')
     + X(1,I),Y(I),Z(I),ZI(I),PCT
630          CONTINUE
      SUMSQR = SUMSQ
      FDATA = DATA
      FPARM = PARM
      SEE = SQRT (SUMSQR/ FDATA)
      AVP = SUMP / FDATA
      IRDF = DATA - PARM
      RDF = IRDF
      RMS = SUMSQR / RDF
      RRMS = SQRT(RMS)
      IF (ICONST .LE. 0)  GO TO 632
      WRITE (OUTPUT,22) (CONST(I), I = 1, ICONST)
22    FORMAT('0    CONSTANTS : ',1P8D12.5/)
632   WRITE (OUTPUT,21) SEE,AVP,IRDF,RMS,RRMS
21    FORMAT (/' STANDARD ERROR OF ESTIMATE =',1PE15.6/
     @        ' ABSOLUTE AVERAGE PERCENT ERROR = ',0PF7.2/
     @        ' RESIDUAL DEGREES OF FREEDOM =',I4/
     @        ' RESIDUAL MEAN SQUARE =',1PE15.6/
     @        ' RESIDUAL ROOT MEAN SQUARE =',E15.6)
             IF(IREPEAT.EQ.1)THEN
                           CALL RESET
                           WRITE(OUTPUT,*)' '
                           WRITE(OUTPUT,*)' ********  REPEAT REGRESSION *******'
                           WRITE(OUTPUT,*)'    REPEAT # ',IREPNUM+1
                           WRITE(OUTPUT,*)' '
                           GOTO 9268
```

```
              ENDIF
              IF(NTERMS.EQ.3)THEN
      GM=((B(1)*LOG(B(3)))+(B(2)*LOG(B(4)))+(COEF3*LOG(B(5))))/CONST(2)
      AM=((B(1)*B(3))+(B(2)*B(4))+(COEF3*B(5)))/CONST(2)
              SS=((B(1)*(B(3)**2))+(B(2)*(B(4)**2))+(COEF3*(B(5)**2)))
              ELSEIF(NTERMS.EQ.2)THEN
      GM=((B(1)*LOG(B(2)))+(COEF2*LOG(B(3))))/CONST(2)
      AM=((B(1)*B(2))+(COEF2*B(3)))/CONST(2)
              SS=((B(1)*(B(2)**2))+(COEF2*(B(3)**2)))
              ELSE
      GM=LOG(B(2))
      AM=B(2)
              SS=B(2)**2
              ENDIF
              GM=EXP(GM)
************

              SMS=SS/CONST(2)
              FARMSQ=AM**2
              BETA=((CONST(2)-1)/CONST(2))*(SMS/FARMSQ)
              RMS=SMS**0.5
              WRITE(OUTPUT,*)' NUMBER OF OSCILLATORS = ',CONST(2)
              WRITE(OUTPUT,*)' '
              WRITE(OUTPUT,*)'  ESTIMATED ARITHMETIC MEAN FREQUENCY'
              WRITE(OUTPUT,'(5X,F10.3)')AM
              WRITE(OUTPUT,*)' '
              WRITE(*,*)' '
              WRITE(*,*)'  ESTIMATED ARITHMETIC MEAN FREQUENCY'
              WRITE(*,'(5X,F10.3)')AM
              WRITE(*,*)' '
              WRITE(OUTPUT,*)' '
              WRITE(OUTPUT,*)'  ESTIMATED GEOMETRIC MEAN FREQUENCY'
              WRITE(OUTPUT,'(5X,F10.3)')GM
              WRITE(OUTPUT,*)' '
              WRITE(*,*)' '
              WRITE(*,*)'  ESTIMATED GEOMETRIC MEAN FREQUENCY'
              WRITE(*,'(5X,F10.3)')GM
              WRITE(*,*)' '

              WRITE(OUTPUT,17004)RMS
17004   FORMAT(6X,'Root Mean Square frequency = ',f10.3,' ( 1/cm )')
              WRITE(OUTPUT,17002)SMS
17002   format(6x,'Mean Square frequency = ',1pe15.3,' ( 1/cm^2 )')
              WRITE(OUTPUT,17003)FARMSQ
17003   format(6x,'Square of Mean frequency = ',1pe15.3,' ( 1/cm^2 )')
              WRITE(OUTPUT,17005)BETA
17005   format(6x,'modified frequency dispersion parameter = ', 1pe15.3)
              WRITE(*,17004)RMS
              WRITE(*,17002)SMS
              WRITE(*,17003)FARMSQ
              WRITE(*,17005)BETA
              CALL SIGMACP(B)
              STOP ' END OF Cp FIT'
```

```
             FUNCTION AVG(A,B)
             IMPLICIT REAL*8 (A-H,O-Z)
             DIMENSION A(1),B(1)
             AVG=(A(1)+B(1))/2.0
             RETURN
             END


      SUBROUTINE FOFX(I,X,B,Y,Z)
$LARGE
             IMPLICIT REAL*8(A-H,O-Z)
      DOUBLE PRECISION B,CONST,XIN,THETA,CV,TERM(3),SUM
      DIMENSION X(1),B(1),CONST(40)
      COMMON /CONST/ CONST,NTERMS,NROT
             XIN=X(1)
*            write(*,*)x(1),y
             Z=0.0
             DO 100 II=1,NTERMS
             THETA=0.0
             IF(NTERMS.EQ.1)THEN
                          CALL CVVIB(B(2),XIN,CV,THETA)
             ELSE
                          CALL CVVIB(B(NTERMS+II-1),XIN,CV,THETA)
             ENDIF
             IF(II.NE.NTERMS.OR.NTERMS.EQ.1)THEN
                          TERM(II)=B(II)*CV
             ELSE IF(II.EQ.NTERMS.AND.NTERMS.GT.1)THEN
                          SUM=0.0
                          DO 234 IK=1,NTERMS-1
234                       SUM=SUM+B(IK)
                          TERM(II)=(CONST(2)-SUM)*CV
             ENDIF
             Z=Z+TERM(II)
100          CONTINUE
             IF(CONST(3).EQ.0.)THEN
*                         non-linear molecule
*                         Z=(Z+4.0+(NROT/2.))*1.987*CONST(1)
*   theory tells us to add (nrot/2) to Cp/R, but fits better
*    if we ignore this.
*                         Z=(Z+4.0)*1.987*CONST(1)
                          Z=(Z+4.0)*1.987*CONST(1)
             ELSE
*                         linear molecule
                          Z=(Z+(7./2.))*1.987*CONST(1)
             ENDIF
*            write(*,*)'xin ',xin,' z= ',z
*            write(*,*)'theta= ',theta
      RETURN
      END
```

```
             SUBROUTINE CVVIB(FREQ,T,CV,THETA)
$LARGE
             IMPLICIT REAL*8(A-H,O-Z)
             REAL*8 HP,KB,C,X,THETA,TEST
             HP=6.62517E-34
             KB=1.3804E-23
             C=2.998E+10
             IF(FREQ.GT.25000.0.OR.FREQ.LT.1.)THEN
                          CV=1000.
                          RETURN
             ENDIF
             IF(FREQ.NE.0.0)THEN
                          THETA=((HP*C)/KB)*FREQ
             ELSE
                          IF(THETA.EQ.0.)THEN
                                       CV=1000.
                                       RETURN
                          ENDIF
             ENDIF
             X=THETA/T
             IF(X.GT.230.)THEN
                          CV=1000.
                          RETURN
             ENDIF
             TEST=DEXP(X)
             IF(TEST-1.0.LT.10.0E-6)THEN
                          CV=100.
                          RETURN
             ENDIF
             TOP=EXP(X)
             BOTTOM=(TOP-1.0)**2
             CV=(X**2)*(TOP/BOTTOM)
             RETURN
             END
```

```
        SUBROUTINE SIGMACP(B)
        IMPLICIT REAL*8(A-H,O-Z)
        CHARACTER*70 FILPRN
        DIMENSION B(1),CONST(40),T(30)
        COMMON/CONST/CONST,NTERMS,NROT
        WRITE(*,*)'ENTER PRN FILENAME'
        READ(*,70)FILPRN
70      FORMAT(A70)
        OPEN(12,FILE=FILPRN,STATUS='UNKNOWN')
        WRITE(12,'(1X,A70)')FILPRN
        T(1)=100.
        T(2)=200.
        T(3)=298.15
        T(4)=300.0
        T(5)=400.0
        T(6)=500.0
        T(7)=600.
        T(8)=700.
        T(9)=800.0
        T(10)=900.
        T(11)=1000.0
        T(12)=1100.
        T(13)=1200.
        T(14)=1300.
        T(15)=1400.
        T(16)=1500.0
        T(17)=1600.
        T(18)=1700.
        T(19)=1800.
        T(20)=1900.
        T(21)=2000.0
        T(22)=2500.
        T(23)=3000.0
        T(24)=3500.
        T(25)=4000.
        T(26)=4500.
        T(27)=5000.
        T(28)=5500.
        T(29)=6000.0
        DO 100 I=1,29
        CP=0.
        DO 200 J=1,NTERMS
        THETA=0.0
        IF(NTERMS.EQ.1)THEN
                CALL CVVIB(B(2),T(I),CV,THETA)
        ELSE
                CALL CVVIB(B(NTERMS+J-1),T(I),CV,THETA)
        ENDIF
        IF(J.NE.NTERMS.OR.NTERMS.EQ.1)THEN
                TERM=B(J)*CV
        ELSE IF(J.EQ.NTERMS.AND.NTERMS.NE.1)THEN
                SUM=0.0
                DO 234 IK=1,NTERMS-1
234             SUM=SUM+B(IK)
```

```
                          TERM=(CONST(2)-SUM)*CV
                 ENDIF
                 CP=CP+TERM
200              CONTINUE
                 CP=(CP+4.0)*1.987*CONST(1)
                 WRITE(12,'(1X,2(F10.3,2X))')T(I),CP
100              CONTINUE
                 CLOSE(12,STATUS='KEEP')
                 RETURN
                 END
```

```
             SUBROUTINE PD(NDATA,X,NPARM,B,Y,ZZ,Z,PDERIV,NORMAL,WORK,POIN01)
$LARGE
             IMPLICIT REAL*8(A-H,O-Z)
             DOUBLE PRECISION NORMAL(21,21),WORK(21),B(20),DEL,POIN01,
     +CONST(40),U,U1,DU,DU1,V,V1,DV,DV1,ALPHA,PRD,TERM1,TERM2,TERM3,
     +TERM4,HP,KB,C
             DIMENSION X(10,400),Y(400),PDERIV(20,400),Z(400)
             INTEGER NDATA,NPARM
             COMMON/CONST/CONST,NTERMS,NROT
             HP=6.62517E-34
             KB=1.3804E-23
             C=2.998E+10

C             INITIALIZE NORMAL TO 0.0,  ONLY THE PART OF
C             THE ARRAY THAT WILL BE USED
C
              DO 101 I = 1,NPARM+1
         DO 100 J = I,NPARM+1
         NORMAL(J,I) = 0.0
100          CONTINUE
101          CONTINUE
*************************************************************************
*          REPLACE THIS SECTION WITH EXPLICIT JACOBIAN
*
C
C             CALCULATE THE PARTIAL DERIVATIVES OF THE FUNCTION
C             VALUES WITH RESPECT TO THE PARAMETER VALUES FOR
C             EACH SET OF DATA POINTS.  THESE DERIVATIVES WILL BE
C             STORED IN PDERIV.
C
         DO 200 J = 1,NDATA
             ALPHA=(HP*C)/(KB*X(1,J))
             U2=EXP(ALPHA*B(5))
             TERM3=(ALPHA*B(5))**2
             TERM4=(U2/((U2-1.0)**2))
             PRD=TERM3*TERM4
             DO 150 I=1,2
             U1=EXP(ALPHA*B(I+2))
             TERM1=(ALPHA*B(I+2))**2
             TERM2=(U1/((U1-1.0)**2))
             PDERIV(I,J)=(TERM1*TERM2-PRD)*1.987*CONST(1)
150          CONTINUE
             DO 1000 I=3,5
             U=(ALPHA*B(I))**2
             DU=2.0*(ALPHA**2)*B(I)
             U1=EXP(ALPHA*B(I))
             DU1=ALPHA*U1
             V1=(U1-1.0)**2
             DV1=(2.0*(U1-1.0)*DU1)
             V=U1/V1
             DV=((1.0/V1)*DU1)-((U1/(V1**2))*DV1)
             PDERIV(I,J)=(U*DV)+(V*DU)
             IF(I.LT.5)THEN
                           PDERIV(I,J)=PDERIV(I,J)*B(I-2)*1.987*CONST(1)
             ELSE
                          PDERIV(I,J)=PDERIV(I,J)*(CONST(2)-(B(1)+B(2)))*1.987*
     +CONST(1)
             ENDIF
1000         CONTINUE

*                     PDERIV(J,I) = (ZZ - Z(I)) / DEL
200          CONTINUE
*
*
************************************************************************
C
C             COMPUTE THE  NORMAL EQUATIONS
C
      DO 240 J1 = 1,NDATA
         DO 230 J2 = 1,NPARM
230         WORK(J2) = PDERIV(J2,J1)
         WORK(NPARM+1) = Y(J1) - Z(J1)
         DO 240 J3 = 1,NPARM+1
            DO 240 J4 = J3,NPARM+1
240            NORMAL(J4,J3) = NORMAL(J4,J3) + WORK(J3) * WORK(J4)
             RETURN
             END
```

APPENDIX V

SOURCE CODE LISTING

THERM COMPUTER PACKAGE

APPENDIX V-A

FORTRAN SOURCE CODE FOR

THERM MAIN PROGRAM

```
$LARGE
C
C      PROGRAM THERM
C**************************************************************
C*                                                            *
C*                                                            *
C*      THERMODYNAMIC PROPERTY ESTIMATION BY BENSON'S         *
C*                                                            *
C*               GROUP ADDITIVITY METHOD                      *
C*                                                            *
C*                 program developed at:                      *
C*                                                            *
C*          NEW JERSEY INSTITUTE of TECHNOLOGY                *
C*                                                            *
C*              KINETICS RESEARCH LABORATORY                  *
C*                                                            *
C*       Under the direction of Dr. Joseph W. Bozzelli        *
C*                                                            *
C*                                                            *
C*                                                            *
C*    VERSION :  4.0                                          *
C*    DATE :  4/22/89                                         *
C*                                                            *
C*                                                            *
C*                                                            *
C*    WRITTEN BY : EDWARD R. RITTER                           *
C*                                                            *
C*                                                            *
C*    OTHER CONTRIBUTORS:                                     *
C*          S. C. CHUANG   (PC BASIC VERSION 1)               *
C*          J. W. BOZZELLI (PC BASIC VERSION 2)               *
C*                                                            *
C*                                                            *
C**************************************************************
$INCLUDE: 'forexec.inc'
             INTEGER*2 SYSTEM, SPAWN, MODE
             CHARACTER STRING
      INTEGER IX(50), E(50), N, NGRPT, Q,EI,FIL,Z1,I9(50),NSAVE,
     +I1CH(50),I2CH(35),INDX(70),GRTOT,NLROT(400)
      INTEGER FLAG, BFLAG, SYMM,SYMS, SYMR, CTRL, Z, NGRP, NATOM,
     +NED,ISORT(500),KEY(10),I3(25)
      INTEGER IG, Y, NG, P, X, O, D,NE(4),NE1(4,400),FLG,IFIRST(25),
     +ILAST(25),EP(50),NATOMP,NP(4),IOP,IDOC(500),NSPEC,KDOCT,ITMP(25),
     +IGRL2(400)
             INTEGER*2 IMON,IDAY,IYR
      REAL A(400,9),S(400,9),SCOR, CPINF, CP2K,DELTA(9)
      REAL*4 B(2,9),BSAVE(9),BO(9)
      CHARACTER*70 TEXT(50),TEXTS(50),EXEFIL,NUMIN1,GRDOC(400),
     +GRLIN2(100)
             CHARACTER*14 DOC(400)
        CHARACTER*70 OPTION,KS(25),NASAFI,NASAFO,FILNM
        CHARACTER*70 DS1,DUMMY,FORM,FORMP,S1,S2,HLP(10)
      CHARACTER*14 AS(400),SS(400),NUMIN,TSS,SPECP,FS2,THERMFIL,SPECL
      CHARACTER*14 CS(2,60), WS(5) , LS, RS,CP, TS, FS,DS
             CHARACTER*2 ELEM(50),DRIVE1,DRIVE2,DRIVE3,ERRORCK
      CHARACTER ES, ANSS,PHASE,PHSE(400),TITLE*30,HEADIN*126
      CHARACTER*14 NUL,NUL1(2,50),DOCFIL,RS22,DOCTYPE,COMMAND,OUTPUT,
     +UNITS,GRUNITS,FOUNITS,FIUNITS
      CHARACTER*4 IE(4),IP(4),IE1(4,400),REF(2),REF1(2,400)
      CHARACTER*6 REF3,REF2(400)
             LOGICAL EXST,OPNED
        COMMON/CONTRL/FIL
             COMMON/CTRL2/COMMAND,OUTPUT
        COMMON/DELETE/NDEL,IDEL(500)
             COMMON/NASANM/NASAFI,NASAFO
             COMMON/CONFG1/ELEM
             COMMON/CONFG2/NELEM,N1CH,N2CH,I1CH,I2CH
             COMMON/CONFG3/HLP
             COMMON/CONFG4/DRIVE1,DRIVE2,DRIVE3
             COMMON/CONFG5/NKEYS
             COMMON/CONFG6/UNITS
             COMMON/CONFG7/ERRORCK
             COMMON/GRFNM1/NFILES,I3
             COMMON/GRFNM2/KS
             COMMON/UNITSET/FOUNITS,FIUNITS
             COMMON/ELEM1/NE1
             COMMON/ELEM2/IE1
             COMMON/FILEFLG/NCALL
             COMMON/GROUPD/GRDOC,GRLIN2
             COMMON/GROUPI/IGRL2
             COMMON/DATE/IMON,IDAY,IYR
             COMMON/NROTOR/NROT,NROTS
             COMMON/NROTLST/NLROT
*            SET CREATION DATE
                         CALL GETDAT(IYR,IMON,IDAY)
                         IF(INT(IYR/100).EQ.19)THEN
                                     IYR=IYR-1900
                         ELSE
                                     IYR=IYR-2000
                         ENDIF
*
             IXFER=0
             ERRORCK=' '
             GRUNITS='KCAL'
             FOUNITS='KCAL'
             FIUNITS='KCAL'
591          CONTINUE
      CALL CLS
        DO 6677 I=1,9
        KEY(I)=0
        BSAVE(I)=0.0
6677    DELTA(I)=0.0
        ICNT=0
        NED=0
        RXFIL=0
        FILNM=' '
      WRITE (*,6645)
```

```
6645  FORMAT('0',7X,'                         THERM')
      CALL LINES (0,1)
      WRITE(*,6646)
6646  FORMAT('0',18X,'THermo Estimation for Radicals & Molecules')
      CALL LINES (0,2)
      WRITE (*,666)
666   FORMAT('0', 32X,'written by')
      write(*,667)
667   format('0',7X,'               E. R. Ritter & J. W. Bozzelli ')
      write(*,668)
668   format(' ',15x,'Department of Chemical Engineering and Chemistry')
      write(*,669)
669   format(' ',21x,'New Jersey Institute of Technology')
      CALL LINES(0,2)
      WRITE(*,6647)
6647    FORMAT('0',7X,'                    revision 4.0 : 4/22/89')
             INQUIRE(FILE='THERM.CFG',EXIST=EXST)
             IF(EXST)GOTO 670
680          continue
               STOP '"THERM.CFG" is missing'

670          CONTINUE
             CALL LINES(0,2)
      write(*,671)
671          FORMAT('+',20(' '),' **** Loading Configuration File ****')

             CALL CONFIG(ELEM,NFILES,KS,ISTAT,OUTPUT)
             IF(ISTAT.NE.0)GOTO 680
C      WRITE(*,*)' '
91827 CONTINUE
      NGRPT = 0
      D = 0
      DO 8 I1 = 1, NFILES
*----------------------------------------------------------------*
*        inquire if group data file specified in THERM.CFG exists   *
*----------------------------------------------------------------*
92827        continue
1       FORMAT(A14)
             INQUIRE(FILE=KS(I1),EXIST=EXST)
             IF(EXST)GOTO 88
             CALL CLS
             WRITE(*,*)'  CANNOT FIND GROUP DATA FILE'
             WRITE(*,*)'            ',KS(I1)
             CALL LINES(0,2)
             WRITE(*,*)'  enter a new path and filename { "Q" to quit }'

             READ(*,'(A)')DUMMY
             CALL READCHAR(DUMMY)
             IF(dummy.EQ.'Q')STOP 'THERM4 aborted by user'
             KS(I1)=DUMMY
             GOTO 92827
*-----------------------------------------------------*
*         end inquire file block                      *
*-----------------------------------------------------*
```

```
88           CONTINUE
             OPEN(I1,FILE=KS(I1),STATUS='OLD')
             READ(I1,52)DUMMY
             CALL LRDCHAR(DUMMY)
             IF(DUMMY.EQ.'BOND_DISSOCIATION')I3(4)=I1
             IF(DUMMY.EQ.'RADICAL_GROUPS')I3(3)=I1

             CLOSE(I1,STATUS='KEEP')
8            CONTINUE
             IF(I3(4).EQ.0)THEN
                          IFIRST(4)=0
                          ILAST(4)=0
             ENDIF
             IF(I3(3).EQ.0)THEN
                          IFIRST(3)=0
                          ILAST(3)=0
             ENDIF
             I=0
             DO 33650 I2=1,NFILES-2
33651        I=I+1
             IF(I.EQ.I3(3).OR.I.EQ.I3(4))GOTO 33651
             ITMP(I2)=I
33650        CONTINUE
             I3(1)=ITMP(1)
             I3(2)=ITMP(2)
             DO 511 I=5,NFILES
511          I3(I)=ITMP(I-2)
             WRITE(*,33652)
33652 FORMAT('+',22(' '),'***** LOADING GROUP DATAFILES *****',5(' '))
             N2LIN=0
             DO 300 I1=1,NFILES
             grunits='KCAL'
             IF(I3(I1).EQ.0)GOTO 300
        OPEN(I1,FILE=KS(I3(I1)),STATUS='OLD')
             IF(I1.EQ.3.OR.I1.EQ.4)READ(I1,52)DUMMY
*    determine units for group data
             read(i1,52)dummy
             call lrdchar(dummy)
             call deblnk(dummy,s1,len1,0)
             if(dummy(:6).eq.'UNITS:')then
                          if(dummy(7:8).eq.'KJ')grunits='KJ'
             else if(dummy(:6).ne.'UNITS:')then
                          backspace(i1)
             endif
*
        READ (I1,*) Q
        IFIRST(I1)=NGRPT+1
        ILAST(I1)=NGRPT+Q
        DO 400 J = 1, Q
          D = D + 1
             GRDOC(D)=' '
             GRLIN2(N2LIN+1)=' '
             if(grunits.ne.'KJ')grunits='KCAL'
             if(grunits.eq.'KCAL')then
          READ(I1, 2) AS(D), (A(D,ID9),ID9=1,9),GRDOC(D),DUMMY
2        FORMAT(A,9F7.2,2A70)
             else if(grunits.eq.'KJ')then
          READ(I1,39) AS(D), (A(D,ID9),ID9=1,9),GRDOC(D),DUMMY
39           FORMAT(A,9F8.2,2A70)
             endif
             IF(DUMMY.NE.' ')THEN
                         N2LIN=N2LIN+1
*                        A(D,9)=FLOAT(N2LIN)
                         IGRL2(D)=FLOAT(N2LIN)
                         GRLIN2(N2LIN)=DUMMY
             ELSE
                         IGRL2(D)=0
             ENDIF
*       convert units to kcal if necessary
             if(grunits.eq.'KJ')then
                         do 445 j1=1,9
445                      a(d,j1)=a(d,j1)/4.184
             endif
*
        IN3=1
C    CHANGE COMMA AT END OF GROUP NAME TO A BLANK
 398    IF(AS(D)(IN3:IN3).EQ.',')THEN
              AS(D)(IN3:14)=' '
              GOTO 399
        ENDIF
        IN3=IN3+1
        IF(IN3.GT.14)GOTO 399
        GOTO 398
399     CONTINUE
400     CONTINUE
      NGRPT = NGRPT + Q
      CLOSE(I1)
300   CONTINUE
      write(*,909)ngrpt
909   FORMAT('+',27X,I3,'  Groups in the dataBase.          ')
      WRITE(*,*)' '
      WRITE(*,*)'                             { hit return to continue }'
      read(*,3)NUMIN
        call readchar(numin)
C
C
************************ END OF GROUP DB INPUT ********************
C
C
        IF(NUMIN.EQ.' '.OR.NUMIN.EQ.'1'.OR.NUMIN.EQ.'L')THEN
                IHELP=1
                GOTO 27
        ENDIF
        IOP=0
        IF(NUMIN.EQ.'0'.OR.NUMIN.EQ.'S')THEN
                IHELP=0
                GOTO 27
        ENDIF
```

```
C
C****************************************************************
C****************** THERM MAIN MENU*********************************
C*****
C
 27     CALL CLS
             COMMAND=' '
C      RESET ALL APPROPRIATE FLAGS HERE
C
             SPECL=' '
             IIICT=0
        ISOURCE=0
        BLFAG=0
        BLFLAG=0
        FLAG=0
             NROT=0
             NROTS=0
        BD=0.0
        DELB=0.0
        DO 28 I=1,9
28      DELTA(I)=0.0
        DOCTYPE=' '
        KDOCT=0
        NED=0
C

      WRITE(*,4300)
4300  FORMAT(1X,30X,'THERM MAIN MENU')
             CALL LINES(0,1)
             if(units.ne.'KJ')units='KCAL'
             if(units.eq.'KCAL')then
                          write(*,98256)IMON,IDAY,IYR
98256 FORMAT(6X,'   Units:  Kcal/mol  ( cal/mol K )',19X,I2,2('/',I2))
             else if(units.eq.'KJ')then
                          write(*,98257)IMON,IDAY,IYR
98257 FORMAT(6X,'   Units:  KJ/mol  (J/mol K)',25X,I2,2('/',I2))
             endif
      WRITE(*,4310)
4310  FORMAT(1X,5X,75('_'))
        CALL LINES(0,1)
      WRITE(*,4320)
c                                                          column 72 ]

4320  FORMAT(6X,'1 - ENTER/ESTIMATE SPECIES',5X,'5 - REVIEW (LST FORMAT
     +FILE)')
      WRITE(*,4330)
4330  FORMAT(6X,'2 - VIEW GROUPS',16X,'6 - GET SPECIES FROM FILE {*.DOC}
     +')
      IF(IHELP.EQ.1)WRITE(*,4335)'SHORT'
      IF(IHELP.EQ.0)WRITE(*,4335)'LONG'
4335  FORMAT(6X,'3 - SELECT ',A5,' MENU',10X,'7 - ReWrite THERMO FILES {
     +*.DOC, *.LST}')
      WRITE(*,4337)
4337  FORMAT(6X,'4 - SELECT LIST FILE {*.LST}',3X,'8 - RUN THERMFIT crea
     +te NASA {*.DAT}')
        WRITE(*,4437)
4437  FORMAT(1X,16X,'9 - RUN THERMRXN (calculate dU, dH, dS, dG for reac
     +tion)')
        WRITE(*,4438)
4438  FORMAT(1X,15X,'10 - Sort & ReWrite THERMO FILES { *.DOC, *.LST }')
        WRITE(*,4439)
4439  FORMAT(16X,'11 - Sort/Write "LST" file or Write w/ new Units }')
             WRITE(*,44391)
44391 FORMAT(1X,15X,'             { no "DOC" format file is needed }')
             write(*,4444)
4444  format(1x,15x,'12 - RUN THERMLST (LIST NASA FORMAT FILE) /create *
     +.LST')
             WRITE(*,4440)
4440  FORMAT(1X,15X,'13 - DOS utilities')
             if(units.eq.'KCAL')then
                          write(*,4840)
4840         format(1x,16x,'U - CHANGE CURRENT UNITS TO : KJ/mol  (J/mol K)')
             else if(units.eq.'KJ')then
                           write(*,4841)
4841         format(1x,16x,'U - CHANGE CURRENT UNITS TO : Kcal/mol  (cal/mol
     +K)')
             endif
c         CALL LINES(0,1)
             WRITE(*,4843)
4843         FORMAT(1X,25X,'*  -  CLOSE ALL OPENED FILES')
*            write(*,4842)
*4842  format(1x,25x,'C  -  review / set configuration')
      WRITE(*,4336)
4336  FORMAT(1X,25X,'X  -  EXIT PROGRAM ')
             WRITE(*,4342)
4342         FORMAT(1X,25X,'?  -  HELP')
      CALL LINES(0,1)
321   CONTINUE
      WRITE(*,4341)
4341   FORMAT(1X,27X,'enter option : { 1 }')
      READ(*,3) NUMIN
      CALL READCHAR(NUMIN)
69696   CONTINUE
             IF(NUMIN(:1).EQ.'*')THEN
                          NCALL = 0
                          FIL=0
                          IOP=0
                          ICNT=0
                          INQUIRE(7,OPENED=OPNED)
                          IF(OPNED)THEN
                                       ENDFILE(7)
                                  CLOSE(7)
                          ENDIF
                          INQUIRE(25,OPENED=OPNED)
                          IF(OPNED)CLOSE(25)
                          INQUIRE(41,OPENED=OPNED)
                          IF(OPNED)THEN
                                       ENDFILE(41)
                                  CLOSE(41)
                          ENDIF
                          INQUIRE(52,OPENED=OPNED)
                          IF(OPNED)THEN
69992                                  read(52,51,end=69993)nul
                                       goto 69992
69993                                  BACKSPACE(52)
                                       NUL=' '
                                       WRITE(52,'(A70)')NUL
                                       WRITE(52,'(A70)')NUL
                                       ENDFILE(52)
                                       CLOSE(52)
                          ENDIF
                          GOTO 27
             ENDIF
             if(numin(:1).eq.'U')then
                          if(units.eq.'KCAL')then
                                       units='KJ'
                                       goto 27
                          else if(units.eq.'KJ')then
                                       units='KCAL'
                                       goto 27
                          endif
             goto 27
             endif
             IF(NUMIN.EQ.'?')THEN
                          WRITE(*,4343)
4343                      FORMAT('+',10X,' HELP ON WHICH OPTION ?')
                          READ(*,52)OPTION
                          CALL READCHAR(OPTION)
                          IF(OPTION.EQ.' '.OR.OPTION.EQ.'Q')GOTO 27
             IF(OPTION.EQ.'1'.OR.OPTION.EQ.'2'.OR.OPTION.EQ.'3'.OR.OPTION.EQ.
     +'4'.OR.OPTION.EQ.'5'.OR.OPTION.EQ.'6'.OR.OPTION.EQ.'7')THEN
                          CALL HELP(HLP(1),OPTION)
                          GOTO 27
             ELSE IF(OPTION.EQ.'8'.OR.OPTION.EQ.'9'.OR.OPTION.EQ.'10'.OR.
     +OPTION.EQ.'11'.OR.OPTION.EQ.'12'.OR.OPTION.EQ.'13'.OR.
     +OPTION(:1).EQ.'X'.OR.OPTION(:1).EQ.'*'.OR.OPTION(:1).EQ.'U')THEN
                          CALL HELP(HLP(2),OPTION)
                          GOTO 27
             ENDIF
             ENDIF

             IF(NUMIN.EQ.'7'.OR.NUMIN.EQ.'10'.OR.NUMIN.EQ.'11')THEN
       FIL=0
             INQUIRE(52,OPENED=OPNED)
             IF(OPNED)THEN
                    CLOSE(52,STATUS='KEEP')
             ENDIF
        CLOSE(7,STATUS='KEEP')
        IOP=0
             IF(NUMIN(:2).EQ.'10'.OR.NUMIN(:1).EQ.'7')GOTO 2109
        ENDIF
        IF(NUMIN(:1).EQ.'Q')THEN
                CALL CLS
        WRITE(*,*)'           TYPE   "X"  TO EXIT THERM '
        WRITE(*,*)'      or hit RETURN to return to main menu'
        read(*,3)NUMIN
        CALL READCHAR(NUMIN)
        IF(NUMIN(:1).NE.'X')GOTO 27
        ENDIF


      IF(NUMIN(:1).EQ.'X') THEN
         CALL CLS
         GOTO 9200
      ENDIF
        IF(NUMIN(:2).EQ.'12')THEN
516          CONTINUE
             EXEFIL=' '
             EXEFIL(:2)=DRIVE2
             EXEFIL(3:15)='THERMLST.EXE'
             INQUIRE(FILE=EXEFIL,EXIST=EXST)
             IF(EXST)GOTO 517
                          CALL CLS
                          WRITE(*,*)'CANNOT FIND:'
                          WRITE(*,*)EXEFIL
                          WRITE(*,*)'NASA to "LST" CONVERSION PROCEDURE'
                          CALL LINES(0,2)
                          WRITE(*,*)'CURRENT DRIVE AND DIRECTORY :'
                          I = SYSTEM('cd'C)
                          PAUSE 'change directory or drive specification'
                          CALL CLS
                          WRITE(*,*)'CURRENT DRIVE AND DIRECTORY :'
                          I = SYSTEM('cd'C)
             WRITE(*,*)'{hit return to continue or "Q" to main menu }'
                          read(*,1)NUMIN
                          CALL READCHAR(NUMIN)
                          IF(NUMIN(:1).EQ.'Q')GOTO 27
                          GOTO 516
517          CONTINUE
             OPEN(13,FILE='THUNITS.TMP',STATUS='UNKNOWN')
                          WRITE(13,1)UNITS
             CLOSE(13,STATUS='KEEP')
             IF(DRIVE2.EQ.'A:')THEN
             I = SPAWN(0,LOC('a:thermlst'C),INT4(0))
             ELSE IF(DRIVE2.EQ.'B:')THEN
             I = SPAWN(0,LOC('b:thermlst'C),INT4(0))
             ELSE IF(DRIVE2.EQ.'C:')THEN
             I = SPAWN(0,LOC('c:thermlst'C),INT4(0))
             ELSE IF(DRIVE2.EQ.'D:')THEN
             I = SPAWN(0,LOC('d:thermlst'C),INT4(0))
             ELSE IF(DRIVE2.EQ.' ')THEN
             I = SPAWN(0,LOC('thermlst'C),INT4(0))
             ENDIF
             IF(I.NE.0)THEN
                          CALL CLS
                          WRITE(*,*)'  ERROR ENCOUNTERED IN NASALST'
                          WRITE(*,*)'  NASA to "LST" CONVERSION PROCEDURE'
                          CALL LINES(0,2)
                          WRITE(*,*)'    ERROR CODE RETURNED = ',I
                          CALL LINES(0,2)
             IF(I.EQ.-1)THEN
                          WRITE(*,*)' insufficient memory available to spawn'
                          write(*,*)'        this procedure.'
                          write(*,*)' use a system w/ 640K RAM or run this'
                          write(*,*)' procedure off line (from DOS directly)'
                          call lines(0,1)
             endif
                          WRITE(*,*)'  { hit return to continue }'
                          read(*,3)numin
             ENDIF
        GOTO 27
      ENDIF

             IF(NUMIN(:2).EQ.'13')THEN
18455        CONTINUE
             KE=KE
             KF=0
10371        IF(KF.EQ.0)THEN
*          DOS OPTION MENU
                          CALL CLS
                          WRITE(*,7300)
7300         FORMAT(1X,30X,' DOS UTILITIES MENU')
                          WRITE(*,7301)
7301         FORMAT(1X,25X,40('_'))
                          CALL LINES(0,1)
                          WRITE(*,7302)
7302         FORMAT(1X,25X,' 1 - SHELL TO DOS { COMMAND.COM }')
                          WRITE(*,7303)
7303         FORMAT(1X,25X,' 2 - CHANGE DEFAULT:   DISK DRIVE')
                          WRITE(*,7304)
7304         FORMAT(1X,25X,' 3 -                   DIRECTORY')
                          write(*,7305)
7305         FORMAT(1X,25X,' 4 -                   PATH')
                          WRITE(*,7306)
7306         FORMAT(1X,25X,' 5 - DIRECTORY:        { *.LST } FILES')
                          WRITE(*,7307)
7307         FORMAT(1X,25X,' 6 -                   { *.DOC } FILES')
                          WRITE(*,7308)
7308         FORMAT(1X,25X,' 7 -                   { *.DAT } FILES')
                          WRITE(*,7309)
7309         FORMAT(1X,25X,' 8 - ISSUE ANOTHER DOS COMMAND')
                          CALL LINES(0,2)
                          WRITE(*,7350)
7350         FORMAT(1X,25X,' enter option {return to previous menu}')
                          read(*,83)numin
83           format(a14)
                          call readchar(numin)
```

```
                         CALL CLS
             ENDIF
c
             if(numin.eq.'1'.or.kf.eq.1)then

             CALL LINES(0,2)
             WRITE(*,*)'  THERM3....type EXIT to return to THERM3'
             call lines(0,1)
                         I = SPAWN(0,LOC('\\command'C),INT4(0))
                         if(i.ne.0.OR.ke.ne.0)GOTO 27
                         if(ke.eq.0)then
                                     kf=ke
                                     goto 10371
                         endif
             CALL CLS
             KF=0
             GOTO 10371
             endif
c
             if(numin.eq.'2'.or.kf.eq.2)then
                         WRITE(*,*)'current disk'
                         WRITE(*,*)'and directory:'
                         CALL LINES(0,1)
                         I = SYSTEM('cd'C)
                         CALL LINES(0,1)
                         PAUSE 'enter new drive ( ex. "A:" )'
                         IF(KE.NE.0)GOTO 27
                         goto 10371
             endif
c
             if(numin.eq.'3'.or.kf.eq.3)then
                         WRITE(*,*)'current disk'
                         WRITE(*,*)'and directory:'
                         CALL LINES(0,1)
                         I = SYSTEM('cd'C)
                         CALL LINES(0,1)
                         PAUSE 'enter new directory ( ex. "cd\" )'
                         IF(KE.NE.0)GOTO 27
                         goto 10371
             endif
c
             if(numin.eq.'4'.or.kf.eq.4)then
                         WRITE(*,*)'current path'
                         CALL LINES(0,1)
                         I = SYSTEM('path'C)
                         CALL LINES(0,1)
                         PAUSE 'enter new path ( ex. "path=\;\dos;\therm" )'
                         IF(KE.NE.0)GOTO 27
                         goto 10371
             endif
c
             if(numin.eq.'5'.or.kf.eq.5)then
                         I = SYSTEM('dir *.lst'C)
                         call lines(0,1)
                         PAUSE '  { hit RETURN to continue }'
                         IF(KE.NE.0)GOTO 27
                         goto 10371
            endif
c
            if(numin.eq.'6'.or.kf.eq.6)then
                         I = SYSTEM('dir *.doc'C)
                         call lines(0,1)
                         PAUSE '  { hit RETURN to continue }'
                         IF(KE.NE.0)GOTO 27
                         goto 10371
            endif
c
            if(numin.eq.'7'.or.kf.eq.7)then
                         I = SYSTEM('dir *.dat'C)
                         call lines(0,1)
                         PAUSE '  { hit RETURN to continue }'
                         IF(KE.NE.0)GOTO 27
                         goto 10371
            endif
c
            if(numin.eq.'8'.or.kf.eq.8)then
                         CALL LINES(0,1)
                         PAUSE
                         PAUSE ' { hit return to continue }'
                         IF(KE.NE.0)GOTO 27
                         goto 10371
            endif
            NUMIN='Q'

                         IF(NUMIN.eq.'q'.OR.NUMIN.EQ.'Q')GOTO 27
                         GOTO 18455
            ENDIF


124     IF((NUMIN(:2).EQ.'10'.AND.NSPEC.NE.0)
     +.OR.(NUMIN(:2).EQ.'11'.AND.ICNT.NE.0))THEN

        DO 7856 LOOP=1,ICNT
7856    ISORT(LOOP)=LOOP
       DO 125 IH=1,10
125    KEY(IH)=0
12512       CALL CLS
            write(*,*)'  Sort database (Y/N/Q)?   :  (Y)'
            read(*,'(a70)')numin1
            call readchar(numin1)
            if(numin1(:1).eq.'Q')goto 27
            if(numin1(:1).eq.'N')goto 12312
            if(numin1(:1).eq.'?')then
                         OPTION='SORT/Y/N/Q'
                         CALL HELP(HLP(2),OPTION)
                         GOTO 12512
            endif
               CALL CLS
```

```
                WRITE(*,*)'    SORTING DATABASE BY COMPOSITION'
                WRITE(*,*)'             PLEASE WAIT'
                          IF(NUMIN(:2).EQ.'10')ICNT=NSPEC
                CALL ATOMSORT(ISORT,NE1,IE1,ICNT,KEY)
                CALL CLS
                IPG=0
                DO 123 KJI=1,ICNT
                IPG=IPG+1
                IF(IPG.GT.20)THEN
                WRITE(*,*)'  { PRESS RETURN TO CONTINUE }'
                READ(*,3)NUL
                IF(NUL(:1).EQ.'Q')GOTO 786
                IPG=0
                ENDIF
                CALL DELCHK(ISORT(KJI),ID)
                IF(ID.EQ.1.or.ss(isort(kji)).eq.' ')GOTO 123
             IF(NUMIN(:2).EQ.'10')THEN
       WRITE(*,876)DOC(ISORT(KJI))(:14),(IE1(KJ,ISORT(KJI)),
     +NE1(KJ,ISORT(KJI)),KJ=1,4)
             ELSE IF(NUMIN(:2).NE.'10')THEN
       WRITE(*,876)SS(ISORT(KJI)),(IE1(KJ,ISORT(KJI)),NE1(KJ,ISORT
     +(KJI)),KJ=1,4)
             ENDIF
123     CONTINUE
12312        CONTINUE
             IF(NUMIN(:2).EQ.'10')THEN
                          ICNT=0
                          GOTO 9786
             ENDIF
876   FORMAT(1X,A14,2X,4(2X,A4,I3))

        IF(NUMIN1(:1).NE.'N')THEN
                          WRITE(*,*)'   { press return to continue }'
                          READ(*,3)NUL
             ENDIF
             IF(NUL(:1).EQ.'Q'.OR.NUL(:1).EQ.'q')GOTO 27
786     CONTINUE
        IF(NUMIN(:2).EQ.'11')THEN
                FIL=2
                     IQR=0
        DO 23987 JN=1,ICNT
C       IGNORE SPECIES ON THE DELETED LIST
                DO 44441 IHJ=1,NDEL
44441           IF(ISORT(JN).EQ.IDEL(IHJ))GOTO 23987
             bflag=0
             blflag=0
             NROT=NLROT(ISORT(JN))
             NROTS=NROT
        LS=SS(ISORT(JN))
        DO 23997 JM=1,9
23997   B(1,JM)=S(ISORT(JN),JM)
             CP2K=B(1,8)
        DO 23999 JM=1,4
        NE(JM)=NE1(JM,ISORT(JN))
```

```
23999   IE(JM)=IE1(JM,ISORT(JN))
        REF(1)=REF1(1,ISORT(JN))
        REF(2)=REF1(2,ISORT(JN))
        REF3=REF2(ISORT(JN))
             COMMAND='AUTO_RECALC'
             IF(NUMIN(:2).EQ.'11')DOCTYPE='NOCALC'
      CALL FILE(LS,RS,B,CP2K,REF,REF3,IE,NE,CS,IX,MM1,E,FLAG,BFLAG
     +,TEXT,BLFLAG,FORM,FORMP,SPECP,RS22,CPINF,SYMR,SYMS,SYMM,BD,TS,FIL
     +,N,KOUNT,LENF,LENFP,DELB,DOCTYPE,KDOCT,DELTA,IOP)
             IF(FIL.EQ.-10)THEN
                          FIL=0
                          GOTO 27
             ENDIF
             COMMAND=' '
             DOCTYPE=' '
        IF(IQR.EQ.0)THEN
        WRITE(*,*)'  ReWriting Database. Please Wait'
        IQR=1
        ENDIF
23987   CONTINUE
             NUMIN(:1)='*'
             GOTO 69696
*        GOTO 27
        ENDIF
        GOTO 2109
        ENDIF
        IF(NUMIN(:2).EQ.'11'.AND.ICNT.EQ.0)GOTO 2108
      IF((NUMIN(:1).EQ.'1').OR.(NUMIN(:1).EQ.'E')) GOTO 10000
      IF((NUMIN(:1).EQ.'2').OR.(NUMIN(:1).EQ.'V')) THEN
                IENTRY=0
      CALL LISTGR(AS,A,NGRPT,IFIRST,ILAST,IENTRY,IHELP,NE1,IE1)
         GOTO 27
      ENDIF
        IF(NUMIN(:1).EQ.'3'.OR.NUMIN(:1).EQ.'M'.OR.NUMIN(:1)
     +.EQ.'?')THEN
                IF(IHELP.EQ.0)THEN
                        IHELP=1
                        GOTO 27
                ENDIF
                IF(IHELP.EQ.1)THEN
                        IHELP=0
                        GOTO 27
                ENDIF
        ENDIF


c                                                            column 72 ]
2108    CONTINUE
        IF(NUMIN(:1).EQ.'D'.OR.NUMIN(:1).EQ.'4'.OR.NUMIN(:1).EQ.'L'
     +.OR.((NUMIN(:1).EQ.'5'.OR.NUMIN(:2).EQ.'11').AND.ICNT.EQ.0))THEN
          ICNT=0
          NDEL=0
          CALL READSP(S,SS,ICNT,IE1,NE1,REF1,REF2,PHSE,IENTRY,IHELP)
```

```
               IF(ICNT.EQ.0)GOTO 27
          IF(NUMIN(:1).EQ.'8'.OR.NUMIN(:1).EQ.'9')GOTO 87
          IF(NUMIN(:2).EQ.'11')GOTO 124
           GOTO 27
         ENDIF


87       CONTINUE
         IF(NUMIN(:1).EQ.'8')THEN
               IF(IIICT.EQ.0)THEN
                              CALL CLS
                              WRITE(*,*)'  ENTER {*.LST} FILENAME FOR INPUT TO'
               WRITE(*,*)'       THERMFIT NASA TYPE POLYNOMIAL PROCEDURE '
                              read(*,52)NASAFI
                              CALL LRDCHAR(NASAFI)
               IF(NASAFI.EQ.' '.OR.NASAFI.EQ.'Q'.OR.NASAFI.EQ.'q')GOTO 27
               call fname(nasafi,'lst')
                              INQUIRE(FILE=NASAFI,EXIST=EXST,ERR=27)
                              IF(EXST)GOTO 193
                              CALL CLS
                              WRITE(*,*)NASAFI, 'DOES NOT EXIST'
                              CALL LINES(0,2)
                              WRITE(*,*)' { hit RETURN to continue }'
                              read(*,3) numin
                              goto 27
193                           CALL CLS
                              WRITE(*,*)'  ENTER {*.DAT} OUTPUT FILENAME'
                              WRITE(*,*)'    FOR NASA FORMAT COEFFICIENTS'
                              read(*,52)NASAFO
                              CALL LRDCHAR(NASAFO)
               call fname(nasafo,'dat')
               IF(NASAFO.EQ.' '.OR.NASAFO.EQ.'Q'.OR.NASAFO.EQ.'q')GOTO 27
               ENDIF
               CLOSE(7,STATUS='KEEP')

               INQUIRE(52,OPENED=OPNED)
               IF(OPNED)THEN
               DUMMY=' '
               WRITE(52,'(A70)')DUMMY
               WRITE(52,'(A70)')DUMMY
                              ENDFILE(52)
                              CLOSE(52,STATUS='KEEP')
               ENDIF

               CLOSE(34,STATUS='KEEP')
               FIL=0
               IOP=0
               ICNT=0
802            CONTINUE
               EXEFIL=' '
               EXEFIL(:2)=DRIVE1
               EXEFIL(3:15)='THERMFIT.EXE'
               INQUIRE(FILE=EXEFIL,EXIST=EXST)
```

```
              IF(EXST)GOTO 801
              CALL CLS
              WRITE(*,*)'  CANNOT FIND:'
              WRITE(*,*)EXEFIL
              CALL LINES(0,2)
              WRITE(*,*)'  CURRENT DEFAULT DISK AND DIRECTORY:'
              CALL LINES(0,1)
              I = SYSTEM('cd'C)
              CALL LINES(0,2)
       PAUSE ' enter new drive specification or change directory "cd\"'
              CALL CLS
              WRITE(*,*)'  NEW DEFAULT DISK AND DIRECTORY:'
              CALL LINES(0,1)
              I = SYSTEM('cd'C)
              CALL LINES(0,2)
       WRITE(*,*)' { hit RETURN to continue or Q return to main menu }'
              READ(*,3)NUMIN
              CALL READCHAR(NUMIN)
              IF(NUMIN.EQ.'Q')GOTO 27
              GOTO 802
801           CONTINUE
              OPEN(45,FILE='THERMFIT.TMP',STATUS='UNKNOWN')
              WRITE(45,*)'    '
              WRITE(45,52)NASAFI
              WRITE(45,52)NASAFO
64            FORMAT(1X,A14)
              CLOSE(45,STATUS='KEEP')
              IF(DRIVE1.EQ.'A:')THEN
              I = SPAWN(0,LOC('a:thermfit'C),INT4(0))
              ELSE IF(DRIVE1.EQ.'B:')THEN
              I = SPAWN(0,LOC('b:thermfit'C),INT4(0))
              ELSE IF(DRIVE1.EQ.'C:')THEN
              I = SPAWN(0,LOC('c:thermfit'C),INT4(0))
              ELSE IF(DRIVE1.EQ.'D:')THEN
              I = SPAWN(0,LOC('d:thermfit'C),INT4(0))
              ELSE IF(DRIVE1.EQ.' ')THEN
              I = SPAWN(0,LOC('thermfit'C),INT4(0))
              ENDIF
              IF(I.NE.0)THEN
                           CALL CLS
                           WRITE(*,*)'  ERROR ENCOUNTERED IN NASA FORMAT'
                           WRITE(*,*)'  DATABASE GENERATION PROCEDURE'
                           CALL LINES(0,2)
                           WRITE(*,*)'    ERROR CODE RETURNED = ',I
                           CALL LINES(0,2)
              if(i.eq.2)then
                           write(*,*)' INPUT FILES NOT FOUND'
              ENDIF
              IF(I.EQ.-1)THEN
                           WRITE(*,*)' insufficient memory available to spawn'
                           write(*,*)'        this procedure.'
                           write(*,*)' use a system w/ 640K RAM or run this'
                           write(*,*)' procedure off line (from DOS directly)'
                           call lines(0,1)
```

```
          endif
                    WRITE(*,*)'  { hit return to continue }'
                    read(*,3)numin
          ENDIF
      GOTO 27
      ENDIF



      IF(NUMIN(:1).EQ.'9')THEN
316       continue
          CLOSE(7,STATUS='KEEP')
          EXEFIL=' '
          EXEFIL(:2)=DRIVE3
          EXEFIL(3:15)='THERMRXN.EXE'
          INQUIRE(FILE=EXEFIL,EXIST=EXST)
          IF(EXST)GOTO 317
                    CALL CLS
                    WRITE(*,*)'CANNOT FIND:'
                    WRITE(*,*)EXEFIL
                    WRITE(*,*)'REACTION ANALYSIS PROCEDURE'
                    CALL LINES(0,2)
                    WRITE(*,*)'CURRENT DRIVE AND DIRECTORY :'
                    I = SYSTEM('cd'C)
                    PAUSE 'change directory or drive specification'
                    CALL CLS
                    WRITE(*,*)'CURRENT DRIVE AND DIRECTORY :'
                    I = SYSTEM('cd'C)
          WRITE(*,*)'{hit return to continue or "Q" to main menu }'
                    read(*,1)NUMIN
                    CALL READCHAR(NUMIN)
                    IF(NUMIN(:1).EQ.'Q')GOTO 27
                    GOTO 316
317       CONTINUE
          OPEN(13,FILE='THUNITS.TMP',STATUS='UNKNOWN')
          WRITE(13,1)UNITS
          CLOSE(13,STATUS='KEEP')
          IF(DRIVE3.EQ.'A:')THEN
          I = SPAWN(0,LOC('a:thermrxn'C),INT4(0))
          ELSE IF(DRIVE3.EQ.'B:')THEN
          I = SPAWN(0,LOC('b:thermrxn'C),INT4(0))
          ELSE IF(DRIVE3.EQ.'C:')THEN
          I = SPAWN(0,LOC('c:thermrxn'C),INT4(0))
          ELSE IF(DRIVE3.EQ.'D:')THEN
          I = SPAWN(0,LOC('d:thermrxn'C),INT4(0))
          ELSE IF(DRIVE3.EQ.' ')THEN
          I = SPAWN(0,LOC('thermrxn'C),INT4(0))
          ENDIF
          INQUIRE(13,EXIST=EXST)
          IF(EXST)THEN
                    OPEN(13,FILE='THUNITS.TMP',STATUS='OLD')
                    READ(13,1)UNITS
                    CLOSE(13,STATUS='DELETE')
          ENDIF
```

```
             IF(I.NE.0)THEN
                          CALL CLS
                          WRITE(*,*)'  ERROR ENCOUNTERED IN THERMRXN'
                          WRITE(*,*)'  REACTION ANALYSIS PROCEDURE'
                          CALL LINES(0,2)
                          WRITE(*,*)'    ERROR CODE RETURNED = ',I
                          CALL LINES(0,2)
             IF(I.EQ.-1)THEN
                          WRITE(*,*)' insufficient memory available to spawn'
                          write(*,*)'        this procedure.'
             write(*,*)' use a system w/ 640K RAM or run "THERMRXN.EXE"'
                          write(*,*)'      from DOS directly'
                          call lines(0,1)
             endif
                          WRITE(*,*)'  { hit return to continue }'
                          read(*,3)numin
             ENDIF
        GOTO 27
      ENDIF


        IF(NUMIN(:1).EQ.'5'.OR.NUMIN(:1).EQ.'T')THEN
        CALL LISTGR(SS,S,ICNT,IFIRST,ILAST,8,IHELP,NE1,IE1)
                NUMIN=' '
                GOTO 27
        ENDIF



2109  IF(NUMIN(:1).EQ.'6'.OR.NUMIN(:1).EQ.'S'.OR.((NUMIN(:1).EQ.'7'.OR.
     +NUMIN(:2).EQ.'10').AND.IOP.EQ.0))THEN
** FOR OPTION 10 AND 7 THIS BLOCK INITIALIZES BOTH LOADSP AND FILE
        IENTER=0
        REF(1)='    '
        REF(2)='    '
        REF3='      '
        SYMS=0
        SYMM=0
        SYMR=0
        FORM=' '
        FORMP=' '
        N=0
        NSAVE=0
        LS=' '
        RS=' '
        SPEC=' '
        SPECP=' '
        NATOM=0
        NATOMP=0
        FLAG=0
        FLG=0
        BFLAG=0
        BLFLAG=0
        SCOR=0.0
        SCORP=0.0
        BD=0.0
        DELB=0.0
        DO 32 I=1,60
        CS(1,I)=' '
32      CS(2,I)=' '
        DO 33 I=1,50
        E(I)=0
33      IX(I)=0
        DO 34 I=1,4
        IE(I)=' '
34      NE(I)=0
        KNT=0
        IF(NUMIN(:1).EQ.'7'.OR.NUMIN(:2).EQ.'10')THEN
                          IENTER=3
                          NCALL=0
             ENDIF
      CALL LOADSP(AS,A,IFIRST,ILAST,NGRPT,CS,B,N,NSAVE,SYMM,SYMS,
     +SYMR,SCOR,SCORP,FORM,FORMP,SPECP,IE,NE,NATOM,FLAG,BFLAG,FLG,LS,RS,
     +RS22,FILST,DOCFIL,IX,E,LENF,EP,FILNM,DOC,NDOC,TEXT,TEXTS,KOUNT,
     +IOP,IDOC,NSPEC,BD,LENFP,IENTER,JC,DELB,DOCTYPE,KDOCT,DELTA)
        IF(NUMIN(:1).EQ.'7'.OR.NUMIN(:2).EQ.'10')THEN
               IOP=1
               IENTER=2
                          IF(NUMIN(:2).EQ.'10')GOTO 124
               GOTO 9786
        ENDIF
               IF(FLG.EQ.1)GOTO 27
        CALL FORMINT(FORMP,LENFP,IP,NP,NATOMP,1)
        ISOURCE=1
           GOTO 8990
      ENDIF



9786      IF((NUMIN(:1).EQ.'7'.OR.NUMIN(:2).EQ.'10').AND.IOP.NE.0)THEN
        IQR=0
         DO 45632 JC=1,NSPEC
        IENTER=2
             IF(NUMIN(:1).EQ.'7')THEN
               CALL DELCHK(JC,ID)
               IF(ID.EQ.1)GOTO 45632
             ELSE IF(NUMIN(:2).EQ.'10')THEN
                          CALL DELCHK(ISORT(JC),ID)
               IF(ID.EQ.1)GOTO 45632
             ENDIF
        SYMS=0
        SYMM=0
        SYMR=0
        FORM=' '
        FORMP=' '
        N=0
        NSAVE=0
```

```
       LS=' '
       RS=' '
       SPEC=' '
       SPECP=' '
       NATOM=0
       NATOMP=0
       FLAG=0
       FLG=0
       BFLAG=0
       BLFLAG=0
       SCOR=0.0
       SCORP=0.0
       BD=0.0
       DELB=0.0
       KNT=0
       DO 69 I=1,60
       CS(1,I)=' '
69     CS(2,I)=' '
       DO 73 I=1,50
       E(I)=0
73     IX(I)=0
       DO 71 I=1,4
       IE(I)=' '
71     NE(I)=0
     IF(NUMIN(:2).EQ.'10')THEN
                        IF(ISORT(JC).EQ.0)THEN
C     ON SORT ERROR RETURN TO MAIN MENU
                                    WRITE(*,*)'SORT ERROR ON SPECIES# ',JC
                                    WRITE(*,*)' AUTO_RECALC ABORTED'
                                    PAUSE' hit return to continue'
                                    GOTO 27
                        ENDIF
                        JL=ISORT(JC)
            ENDIF
     IF(NUMIN(:1).EQ.'7')JL=JC
            COMMAND='AUTO_RECALC'
     CALL LOADSP(AS,A,IFIRST,ILAST,NGRPT,CS,B,N,NSAVE,SYMM,SYMS,
    +SYMR,SCOR,SCORP,FORM,FORMP,SPECP,IE,NE,NATOM,FLAG,BFLAG,FLG,LS,RS,
    +RS22,FILST,DOCFIL,IX,E,LENF,EP,FILNM,DOC,NDOC,TEXT,TEXTS,KOUNT,
    +IOP,IDOC,NSPEC,BD,LENFP,IENTER,JL,DELB,DOCTYPE,KDOCT,DELTA)
            COMMAND=' '
              IF(FLG.EQ.1)GOTO 27
C      CALL FORMINT(FORMP,LENFP,IP,NP,NATOMP,1)
       PHASE='G'
       REF(1)='    '
       REF(2)='    '
       REF3='      '
       DO 45631 JCOUNT=1,N
       IF((IX(JCOUNT).GE.IFIRST(4)).AND.(IX(JCOUNT).LE.ILAST(4)))THEN
              BFLAG=1
              KNT=KNT+1
              IF(KNT.EQ.2)BFLAG=2
              GOTO 45631
       ENDIF
```

```
        IF((IX(JCOUNT).GE.IFIRST(3)).AND.(IX(JCOUNT).LE.ILAST(3)))THEN
                FLAG=1
                GOTO 45631
        ENDIF
        IF(FLAG.EQ.1)GOTO 45631
      FLAG=0
45631  CONTINUE
       ISOURCE=1
          CPINF = ((3*NATOM) - (2.+(FLOAT(NROT)/2.)))*1.987
        CP2K = B(1,9) + .37*(CPINF - B(1, 9))
             COMMAND='AUTO_RECALC'
      CALL FILE(LS,RS,B,CP2K,REF,REF3,IE,NE,CS,IX,MM1,E,FLAG,BFLAG
     +,TEXT,BLFLAG,FORM,FORMP,SPECP,RS22,CPINF,SYMR,SYMS,SYMM,BD,TS,FIL
     +,N,KOUNT,LENF,LENFP,DELB,DOCTYPE,KDOCT,DELTA,IOP)
             IF(FIL.EQ.-10)THEN
                           FIL=0
                           GOTO 27
             ENDIF
             COMMAND=' '
      IF(IQR.EQ.0)THEN
        WRITE(*,*)'ReGenerating Database Files.  Please Wait'
        IQR=1
        ENDIF
45632   CONTINUE
             IIICT=1
        NUMIN='8'
        GOTO 87
c       GOTO 27
      ENDIF
```

```
C
C************* ENTER A NEW SPECIES ******************************
C
10000 FLAG = 0
             NTEMPS=7
        DO 29 I=1,9
29      DELTA(I)=0.0
        NED=0
      BFLAG = 0
        SCORP=0
        BLFLAG=0
        SYMS=0
      SCOR = 0
      SYMM = 0
      SYMR = 0
      GRTOT=0
      IERR=0
        ISOURCE=0
        BD=0.0
        DELB=0.0
        DOCTYPE=' '
        KDOCT=0
      DO 543 IZ1=1,10
      TEXTS(IZ1)=' '
543   TEXT(IZ1)=' '
      KOUNT=0
      KOUNTS=0
      DO 502 Z1=1,3
      DO 500 Z = 1, 9
        B(Z1,Z) = 0.0
500   CONTINUE
502   CONTINUE
501   CALL CLS
      WRITE(*,*) '    Enter species name (ID)'
      WRITE(*,*) '      9 characters maximum'
      WRITE(*,*) '{ "Q" return to opening menu }'
      READ(*,3) LS
3     FORMAT(A14)
      CALL READCHAR(LS)
      IF((LS(:1).EQ.'Q').AND.(LS(2:2).EQ.' '))GOTO 27
             IF(ls(:1).EQ.'?')THEN
                          OPTION='ENTERSPEC'
                          CALL HELP(HLP(3),OPTION)
                          GOTO 502
             ENDIF

      SPECP(:14)=LS(:14)
      M = 1

8803         CALL CLS
      write(*,*)' '
      WRITE(*,*)'enter the elemental formula for this species '
      write(*,*)'{ example: for PHENOL enter " C6H5OH " }'
      WRITE(*,*)'{          for TOLUENE enter " C6H5CH3 " }'
      WRITE(*,*)'{          for CHLOROPHENOL enter " C6H4CLOH " }'
        CALL LINES(0,2)
        WRITE(*,*)'        4 ELEMENTS MAXIMUM : 70 CHARACTERS MAX'
        CALL LINES(0,1)
             IF(IHELP.EQ.1)THEN
             WRITE(*,*)'  THE FOLLOWING ELEMENTS ARE DEFINED:'
             IF(NELEM.GT.10)THEN
                          ISTOP=10
             ELSE IF(NELEM.LE.10)THEN
                          ISTOP=NELEM
             ENDIF
             WRITE(*,1832)(ELEM(I5),I5=1,ISTOP)
1832  FORMAT(' ',10(2X,A2))
             IF(NELEM.LE.20.AND.NELEM.GT.10)WRITE(*,1832)(ELEM(I5),I5=11,
     +NELEM)
             IF(NELEM.GT.20)THEN
           WRITE(*,*)' MORE THAN 20 ELEMENTS ARE DEFINED'
                WRITE(*,*)' NOT ALL ARE SHOWN'
```

```
              ENDIF
              CALL LINES(0,1)
              WRITE(*,*)' ALL OTHER ELEMENTS OR NON-NUMERIC CHARACTERS'
              WRITE(*,*)'             WILL BE IGNORED !'
              CALL LINES(0,1)
              WRITE(*,*)' enter formula here: {Q=QUIT}'
              ENDIF
       read(*,52) FORM
52       FORMAT(A70)
         CALL LRDCHAR(FORM)
         IF(FORM.EQ.' '.OR.FORM.EQ.'Q')GOTO 10000
              IF(FORM(:1).EQ.'?')THEN
                           OPTION='FORM'
                           CALL HELP(HLP(3),OPTION)
                           CALL CLS
                           GOTO 8803
              ENDIF

         CALL DEBLNK(FORM,DUMMY,LENF,0)
   51 FORMAT(A14)
              CALL CLS
              IF(ERRORCK.EQ.' ')THEN
                           iform=0
              ELSE
                           IFORM=1
              ENDIF
      CALL FORMINT(FORM,LENF,IE,NE,NATOM,iform)
              if(iform.lt.0)goto 8803
      FORMP(:LENF)=FORM(:LENF)
        LENFP=LENF
        DO 568 KJ=1,4
                IP(KJ)=IE(KJ)
568             NP(KJ)=NE(KJ)
        NATOMP=NATOM
25001   CONTINUE
              IF(ERRORCK.EQ.' ')THEN
        CALL LINES(0,2)
      WRITE(*,*) ' number of Atoms for This Species =',NATOM
      WRITE(*,*)' '
        WRITE(*,*)'    IF THIS IS INCORRECT ENTER " Q "'
        WRITE(*,*)'           TO BEGIN AGAIN'
        WRITE(*,*)'  OTHERWISE HIT RETURN TO CONTINUE'
      READ(*,3)  FS
      CALL READCHAR(FS)
      IF (FS(1:1) .EQ. 'Q' ) GOTO 8803

2500  IF(IHELP.EQ.1)THEN
              CALL CLS
              WRITE(*,*)'  Radicals are estimated either by entering a'
              write(*,*)'  radical {CDOT} group such as " CB/. " on this'
              write(*,*)'  screen; or by first estimating the properties'
              write(*,*)'  for the stable molecule here { ex: benzene } and'
              write(*,*)'  then applying a bond increment {BD} group to '
              write(*,*)'  account for loss of H atom from the molecule.'
        call lines(0,1)
        write(*,*)'  Bond increments cannot be entered here.  They'
        write(*,*)'  are entered at the RADICAL menu { available'
        write(*,*)'  after the molecule''s properties are estimated }.'
        call lines(0,1)
        write(*,*)'  You must be sure that groups you enter adequately'
        write(*,*)'  define a molecule (no unbonded electrons).'
        write(*,*)'  THERM makes no attempt to check the context'
 write(*,*)'  of the groups entered (only whether they are defined
+).'
        call lines(0,2)
 ENDIF

 WRITE(*,*) '      Enter the Number of Groups in this species'
 WRITE(*,*)' (each Central Atom or Ring Correction counts as 1 grou
+p)'
        ELSE
 WRITE(*,*)' Enter 0 for literature data input or RETURN to enter g
+roups'
        ENDIF
 READ(*,3) NUMIN
 CALL READCHAR(NUMIN)
 IF(NUMIN(1:1).EQ.'Q')GOTO 10000
        IF(NUMIN.EQ.'0')THEN
*        transfer to literature data input block
                    DOCTYPE='NOCALC'
                    GOTO 1277
        ENDIF
        IF(NUMIN(:3).EQ.'REF'.OR.NUMIN(:3).EQ.'LIT')THEN
                    DOCTYPE='NOCALC'
        IF(NUMIN(:3).EQ.'REF'.OR.NUMIN(:3).EQ.'LIT')THEN
                    call cls
                    WRITE(*,*)' enter literature data: record groups'
                    write(*,*)'enter total number of groups:  { Q=QUIT }'
                    read(*,3)numin
                    call readchar(numin)
                    if(numin.eq.' '.or.numin.eq.'Q')goto 10000
                    call readnum(numin,rnum,ngrp,flg)
                    if (flg.eq.1)then
                              flg=0
                              goto 10000
                    endif
                    goto 8802
        ENDIF
                    IF(NUMIN.EQ.'0')THEN
                              N=0
                              NSAVE=0
                              NGRP=0
                    ENDIF
                    IF(NUMIN.EQ.' '.AND.ERRORCK.NE.' ')THEN
                              N=100
                              NSAVE=100
                              NGRP=100
                    ENDIF
```

```
                        GOTO 8802
                ENDIF
                IF(NUMIN(:1).EQ.'?'.AND.IHELP.NE.1.AND .ERRORCK.EQ.' ')THEN
                        IHELP=1
                        GOTO 2500
                ELSE IF (NUMIN(:1).EQ.'?'.AND.IHELP.EQ.1.AND.ERRORCK.EQ.' ')THEN
                        OPTION='NGROUPS'
                        CALL HELP(HLP(3),OPTION)
                        CALL CLS
                        GOTO 2500
                ELSE IF(NUMIN(:1).EQ.'?'.AND.ERRORCK.NE.' ')THEN
                        OPTION='LIT/REF'
                        CALL HELP(HLP(3),OPTION)
                        CALL CLS
                        GOTO 25001
                ENDIF
      CALL READNUM(NUMIN,RNUM,NGRP,FLG)
      IF(FLG.EQ.1)THEN
         FLG=0
         GOTO 10000
      ENDIF
8802  CONTINUE
                IF(NGRP.NE.O.AND.ERRORCK.EQ.' ')THEN
      WRITE(*,*)' '
      WRITE(*,*) 'How many different groups are to be entered?'
      READ(*,3) NUMIN
      CALL READCHAR(NUMIN)

      IF(NUMIN(1:1).EQ.'Q')GOTO 10000
                IF(NUMIN(:1).EQ.'?')THEN
                        OPTION='DGROUPS'
                        CALL HELP(HLP(3),OPTION)
                        CALL CLS
                        GOTO 8802
                ENDIF
                IF(NUMIN.EQ.' '.OR.NUMIN.EQ.'0')THEN
                        N=0
                        NSAVE=0
                        DOCTYPE='NOCALC'
                        GOTO 459
                ENDIF
      CALL READNUM(NUMIN,RNUM,N,FLG)
      IF (FLG.EQ.1)THEN
        FLG=0
        GOTO 10000
      ENDIF
      NSAVE=N
                ENDIF
459             CONTINUE
*------------------ NUMBER FREE ROTORS ( AT  5000 K )--------

22501           CALL CLS
                WRITE(*,*)' how many rotors in this species ( 0 )'
                read(*,'(a)')numin
```

```
            call readchar(numin)
            call dblnk(numin,lnum,0)
            if(numin.eq.'Q')goto 22501
            if(numin.eq.'?')then
                        option='NROTOR'
                        CALL HELP(HLP(3),OPTION)
                        GOTO 22501
            ENDIF
            if(numin.eq.' '.or.numin.eq.'0')then
                        nrot=0
            else
                        call readnum(numin,rnum,NROT,kflg)
                        if(kflg.ne.0)goto 22501
            endif
            nrots=nrot

*-------------------------------------------------------------
45     CALL CLS
            IF((NGRP.NE.0.AND.N.NE.0).OR.(ERRORCK.NE.' '))THEN
      write(*,*)ngrpt,' groups in the database'
      WRITE(*,*)' '
      IF(ERRORCK.EQ.' ')WRITE(*,50)N
  50  FORMAT(1X,'You must enter a total of ',I3,' different groups')
      WRITE(*,*)'  {enter " D " to re-enter the last group}'
      WRITE(*,*)'  {enter " Q " to start a new species}'
      WRITE(*,*)'  {enter " L " to list group database}'
            IF(ERRORCK.EQ.'NO')THEN
      WRITE(*,*)'  {enter "   " ( blank ) when finished}'
            ENDIF
        CALL LINES(0,2)
      WRITE(*,*)' ENTER: "GROUP ID,QUANTITY"'
      IF(IACT.EQ.3)THEN
          IACT=0
          IGM1=IG-1
          CALL LINES(0,2)
          IF(ERRORCK.EQ.' ')WRITE(*,1812)IGM1,N
1812    FORMAT('0',10X,I3,' out of ',I3,' groups have been entered')
            IF(IGM1.LE.0)GOTO 1778
            CALL WRITEGR(1,IGM1,CS,E,IG,6,LS)
            CALL LINES(0,2)
1778        IACT=0
            GOTO 3010
      ENDIF
      GRTOT=0
      IG=0
  600 IG = IG+1
      IF(ERRORCK.EQ.' ')THEN
                      IACT=1
            ELSE
                      IACT=0
            ENDIF
      IF(IG.GT.N.AND.ERRORCK.EQ.' ')GOTO 601
 3010 CALL READGR(IG,CS,EI,IACT,MM1)
        E(IG)=EI
```

```
              IF(IACT.EQ.-10)THEN
                           IG=IG-1
                           N=IG
                           NSAVE=N
                           GOTO 601
              ENDIF
        IF((IACT.EQ.1).AND.(IG.GE.2))IG=IG-2
        IF (IACT.EQ.1)GOTO 600
        IF(IACT.EQ.2) GOTO 10000
        IF(E(IG).EQ.0)E(IG)=1
        IF(IACT.EQ.3)THEN
                IENTRY=0
      CALL LISTGR(AS,A,NGRPT,IFIRST,ILAST,IENTRY,IHELP,NE1,IE1)
            GOTO 45
        ENDIF
              IF(IACT.EQ.4)GOTO 45
      CALL FINDGR(AS,NGRPT,CS,IX,IG,ISTAT,MM1)
      IF(ISTAT.EQ.1)THEN
            ISTAT=0
                GOTO 3010
      ENDIF
      IF((IX(IG).LE.ILAST(3)).AND.(IX(IG).GE.IFIRST(3)))FLAG=1
      IF((IX(IG).LE.ILAST(4)).AND.(IX(IG).GE.IFIRST(4))
     +.AND.(ERRORCK.EQ.' ')THEN
        CALL CLS
        WRITE(*,*)'        ERROR : YOU CANNOT ENTER'
        WRITE(*,*)'        BOND DISSOCIATION GROUPS'
        WRITE(*,*)'            ON THIS SCREEN'
        CALL LINES(0,3)
        WRITE(*,*)'        { hit return to continue }'
        call lines(0,3)
        READ(*,3)NUL
        IACT=3
        GOTO 45
      ENDIF
      GOTO 600
601   CONTINUE
             IF(N.NE.0)THEN

      DO 700 Y = 1, 9
        DO 800 NG = 1, N
             IF(A(IX(NG),9).EQ.0.0.AND.A(IX(NG),8).NE.0.)NTEMPS=6
          B(1, Y) = E(NG) * A(IX(NG), Y) + B(1,Y)
          B(2,Y) = E(NG)*A(IX(NG),Y)+B(2,Y)
          EP(NG)=E(NG)
          CS(2,NG)=CS(1,NG)
800     CONTINUE
700   CONTINUE
             GRTOT=0
             DO 7801 Y=1,N
7801     GRTOT=E(Y)+GRTOT
        IF(GRTOT.NE.NGRP.AND.ERRORCK.EQ.' ')THEN
                          CALL CLS
             WRITE(*,*)' WARNING!  You said there were ',NGRP,' total'
```

```
              WRITE(*,*)'          but ',GRTOT,' groups were identified'
                            CALL LINES (0,2)
              WRITE(*,*)' hit RETURN to continue or enter "Q" to begin again'
                            read(*,1)nul
                            call readchar(nul)
                            if(nul.eq.'Q')goto 10000
              ENDIF
              ENDIF
8804          CONTINUE
              ENDIF
      IF (FLAG .EQ. 0) THEN
          WRITE(*,*) '     Enter symmetry # for molecule'
          READ(*,3) NUMIN
          CALL READCHAR(NUMIN)
          IF(NUMIN(1:1).EQ.'Q')GOTO 45
              IF(numin(:1).EQ.'?')THEN
                            OPTION='SYMMETRY'
                            CALL HELP(HLP(3),OPTION)
                            CALL CLS
                            GOTO 8804
              ENDIF
          CALL READNUM(NUMIN,RNUM,SYMM,FLG)
          SYMS=SYMM
C         WE SAVE PARENTS SYMMETRY NUMBER FOR LATER PROCESSING
          IF(FLG.EQ.1)THEN
             FLG=0
             GOTO 45
          ENDIF
      ENDIF
      IF (FLAG .EQ. 1) THEN
          WRITE(*,*) '     Enter symmetry # for radical'
          READ(*,3) NUMIN
          CALL READCHAR(NUMIN)
          IF(NUMIN(1:1).EQ.'Q')GOTO 45
              IF(numin(:1).EQ.'?')THEN
                            OPTION='SYMMETRY'
                            CALL HELP(HLP(3),OPTION)
                            CALL CLS
                            GOTO 8804
              ENDIF
          CALL READNUM(NUMIN,RNUM,SYMM,FLG)
          IF(FLG.EQ.1)THEN
              FLG=0
              GOTO 45
          ENDIF
      ENDIF
      IF (FLAG .EQ. 1) THEN
        SYMR = SYMM
      ENDIF
              IF(DOCTYPE.NE.'NOCALC')THEN
              REF3(:6)=' THERM'
      SCOR = 1.98717 * ALOG(SYMM*1.0)
      B(1, 2) = B(1, 2) - SCOR
      SCORP=SCOR
      P = NGRP
             ENDIF

************************** LITERATURE INPUT FOLLOWS *********************
1277    CONTINUE
             IF(DOCTYPE.EQ.'NOCALC')THEN
19991        fiunits='KCAL'
             CALL CLS
             WRITE(*,*)'enter literature data { Q=QUIT }'
             call lines(0,1)
             write(*,*)'species: ',LS
             write(*,*)form
             call lines(0,2)
      write(*,*)'input:  Hf, S, Cp300, Cp400, Cp500, Cp600, Cp800, Cp100
     +0, (Cp1500)'
             write(*,*)'        separate numbers with commas or blanks'
             read(*,52)numin1
             do 9765 ij9=1,70
9765         if(numin1(ij9:ij9).eq.',')numin1(ij9:ij9)=' '
             call deblnk(numin1,s2,len,1)
             if(numin1.eq.' '.or.numin1.eq.'q'.or.numin1.eq.'Q'.OR.
     +len.eq.0)goto 10000
             IF(NUMIN1(:1).EQ.'?')THEN
                          OPTION='LIT/INPUT'
                          CALL HELP(HLP(3),OPTION)
                          GOTO 1277
             ENDIF
97651        CALL LINES(0,1)
             write(*,*)' enter units:  1- Kcal/mol (cal/mol K)'
             write(*,*)'               2- KJ/mol   (J/mol K)'
             CALL LINES(0,1)
             write(*,*)'        enter choice { 1 }'
             read(*,3)nul
             IF(NUL(:1).EQ.'?')THEN
                          OPTION='LIT/UNITS'
                          CALL HELP(HLP(3),OPTION)
                          GOTO 97651
             ENDIF
             if(nul(:1).eq.'2')fiunits='KJ'
             call indxst(numin1,' ',noccur,indx,len,1,flg)
             NUMIN=numin1(:indx(1)-1)
             call readnum(NUMIN,rnum,num,flg)
             if(fiunits.eq.'KJ')rnum=rnum/4.184
             B(1,1)=rnum
             b(2,1)=rnum
             do 9766 ij9=2,noccur
             NUMIN=numin1(indx(ij9-1)+1:indx(ij9)-1)
             call readnum(NUMIN,rnum,num,flg)
             if(fiunits.eq.'KJ')rnum=rnum/4.184
             b(1,ij9)=rnum
             b(2,ij9)=rnum
9766         continue
             NUMIN=numin1(indx(noccur)+1:len)
             call readnum(NUMIN,rnum,num,flg)
```

```
            if(fiunits.eq.'KJ')rnum=rnum/4.184
            b(1,noccur+1)=rnum
            b(2,noccur+1)=rnum
            if(units.eq.'KJ')then
                    do 446 j1=1,9
446                 bo(j1)=b(1,j1)*4.184
                    write(*,11)(bo(j1),j1=1,9)
            else if(units.eq.'KCAL')then
                    write(*,10)(b(1,ij9),ij9=1,9)
            endif
*-----------------  NUMBER FREE ROTORS ( AT  5000 K )--------

62501       CONTINUE
            WRITE(*,*)' how many rotors in this species ( 0 )'
            read(*,'(a)')numin
            call readchar(numin)
            call dblnk(numin,lnum,0)
            if(numin.eq.'Q')goto 62501
            if(numin.eq.'?')then
                    option='NROTOR'
                    CALL HELP(HLP(3),OPTION)
                    CALL CLS
                    GOTO 62501
            ENDIF
            if(numin.eq.' '.or.numin.eq.'0')then
                    nrot=0
            else
                    call readnum(numin,rnum,NROT,kflg)
                    if(kflg.ne.0)THEN
                            CALL CLS
                            goto 62501
                    ENDIF
            endif
            nrots=nrot

*--------------------------------------------------------
4661        call lines(0,2)
            write(*,*)'enter text for reference ( journal name, date, etc )'
            write(*,*)'      blank line or "Q" end text input'
            kount=0
9768        read(*,52)dummy
            IF(DUMMY(:1).EQ.'?')THEN
                    OPTION='TEXT'
                    CALL HELP(HLP(3),OPTION)
                    GOTO 4661
            ENDIF
            if(dummy.ne.' '.and.dummy.ne.'q'.and.dummy.ne.'Q')then
                    kount=kount+1
                    text(kount)=dummy
                    IF(KOUNT.LE.50)goto 9768
            endif
            IF(KOUNT.EQ.1.AND.TEXT(KOUNT).EQ.' ')KOUNT=0
            call lines(0,2)
46619       symm=1
```

```
             syms=1
             write(*,*)'  enter symmetry #  ( not used for calculation )'
             read(*,'(a)')numin
             call readchar(numin)
             if(numin.eq.'?')then
                          OPTION='SYMMETRYL'
                          CALL HELP(HLP(3),OPTION)
                          CALL CLS
                          GOTO 46619
             elseif(numin.eq.' '.or.numin.eq.'Q')THEN
                          goto 80009
             else
                          call readnum(numin,rnum,SYMM,kflg)
                          if(kflg.ne.0)goto 46619
                          syms=symm
             endif
80009        CALL CLS
             write(*,*)'enter reference string { "Q" or " " =  no string }'
             write(*,*)'                                  14 characters max'
             read(*,'(a14)')numin
             IF(numin.EQ.'?')THEN
                          OPTION='REF/STR'
                          CALL HELP(HLP(3),OPTION)
                          GOTO 80009
             ENDIF
             if (numin.eq.'Q'.or.numin.eq.' '.or.numin.eq.'q')then
                    REF(1)='    '
                    REF(2)='    '
                    REF3='      '
             else
                          ref(1)(:4)=numin(:4)
                          ref(2)(:4)=numin(5:8)
                          ref3(:6)=numin(9:14)
             endif
             goto 8990
             ENDIF
**************************** END OF LITERATURE INPUT *********************
C
C
C*******************   SPECIES SCREEN *************************
8990  CALL CLS
        X=1
        KNT=0
        DO 8991 JCOUNT=1,N
c                                                            column 72 ]

             IF(NTEMPS.EQ.6)THEN
                          B(1,9)=0.0
                          B(2,9)=0.0
             ENDIF
        IF((IX(JCOUNT).GE.IFIRST(4)).AND.(IX(JCOUNT).LE.ILAST(4)))THEN
               BFLAG=1
               KNT=KNT+1
               IF(KNT.EQ.2)BFLAG=2
```

```
                GOTO 8991
        ENDIF
        IF((IX(JCOUNT).GE.IFIRST(3)).AND.(IX(JCOUNT).LE.ILAST(3)))THEN
                FLAG=1
                GOTO 8991
        ENDIF
        IF(FLAG.EQ.1)GOTO 8991
      FLAG=0
8991  CONTINUE
             IF(DOCTYPE.EQ.'NOCALC')THEN
                           WRITE(*,*)'Thermo Data Entered:'
      ELSE IF(BFLAG.GE.2)THEN
             WRITE(*,*)'Thermo estimation for diradical'
      ELSE IF (FLAG .EQ. 1 .OR. BFLAG .EQ. 1 ) THEN
        WRITE(*,*) 'Thermo estimation for radical'
      ELSE IF (FLAG .EQ. 0 .AND. BFLAG .EQ. 0 ) THEN
        WRITE(*,*) 'Thermo estimation for molecule'
      ENDIF
        IF (BLFLAG .EQ. 1) THEN
          WRITE(*,22) RS22,FORM(:LENF)
22        FORMAT(1X,A14,3X,A50)
23        FORMAT(1X,'RADICAL BASED UPON PARENT  ',A14,3X,A70)
          GOTO 9003
        ENDIF
        IF (BFLAG.GE.1) THEN
               WRITE (*,22)RS,FORM(:LENF)
               WRITE(*,23)SPECP,FORMP(:LENFP)
        ENDIF
        IF (BFLAG.EQ.0) WRITE(*,22)  LS,FORM(:LENF)
        IF(BFLAG.GE.1)WRITE(*,*)'PARENT SYMMETRY',SYMS
          CALL LINES(0,1)
9003      CALL WRITEGR(1,N,CS,E,IG,6,LS)
          CALL LINES(0,1)
             if(units.eq.'KCAL')then
             WRITE(*,*)'UNITS:  Hf{Kcal/mol}   S & Cp {cal/mol K}'
        WRITE(*,*) '   Hf      S     Cp 300    400      500      600      8
     +00    1000    1500'
             else if(units.eq.'KJ')then
             WRITE(*,*)'UNITS:  Hf{KJ/mol}    S & Cp {J/mol K}'
        WRITE(*,*) '     Hf        S       Cp 300    400       500      600
     +      800      1000     1500'
             endif
          CPINF = (3.0*FLOAT(NATOM)-(2.0+FLOAT(NROT/2.0)))*1.987
             if(units.eq.'KCAL')then
        IF(B(1,9).NE.0.)WRITE(*,10) (B(X,ID9),ID9=1,9)
        IF(B(1,9).EQ.0.)WRITE(*,10) (B(X,ID9),ID9=1,8)
10      FORMAT(9F8.2)
11           format(1x,F8.2,8(1X,F8.2))
        WRITE(*,665) CPINF
665     FORMAT(' ',5X,'CPINF',2X,F7.2)
             else if(units.eq.'KJ')then
                           do 659 j1=1,9
659                        bo(j1)=b(x,j1)*4.184
```

```
                       IF(BO(9).NE.0.)write(*,11)(bo(j1),j1=1,9)
                       IF(BO(9).EQ.0.)write(*,11)(bo(j1),j1=1,8)
                       cpio=cpinf*4.184
                       write(*,665)cpio
             endif
        IF (BFLAG .GE. 1) THEN
           WRITE(*,*)'Symmetry number used to correct S value is ',SYMR
           WRITE(*,*) 'R ln(2) has been added to S to account for unpaire
     +d electron'
           IF((TS(:1).EQ.'Q').OR.(TS(:1).EQ.' ').OR.(TS(:1).EQ.'O'))
     +TS(:14)=TSS(:14)
             IF(UNITS.EQ.'KCAL')THEN
           WRITE(*,84848)BD
84848        FORMAT(5X,'bond dissociation energy used = ',f10.2)
             ELSE IF(UNITS.EQ.'KJ')THEN
             BDO=BD*4.184
           WRITE(*,84848)BDO
             ENDIF
        ENDIF
        IF (BFLAG .GE. 1 ) GOTO 900
        WRITE(*,*) 'Symmetry number used to correct S value is', SYMM
900   CONTINUE
             IF(KOUNT.GE.1)THEN
             WRITE(*,*)'              Notes :'
      DO 933 IZ1=1,KOUNT
      IF((BFLAG.EQ.0).OR.(FLAG.EQ.1)) TEXTS(IZ1)=TEXT(IZ1)
933   WRITE(*,74)TEXT(IZ1)
      IF((BFLAG.EQ.0).OR.(FLAG.EQ.1)) KOUNTS=KOUNT
74    FORMAT(1x,A70)
             CALL LINES(0,1)
             ENDIF
9192  WRITE(*,*)'  EDIT THESE RESULTS?     "Y" OR {"N"}'
      READ(*,3) DS
      CALL READCHAR(DS)
             IF(DS(:1).EQ.'?')THEN
                       OPTION='EDIT'
                       CALL HELP(HLP(3),OPTION)
                       CALL CLS
                       GOTO 8990
             ENDIF

             if(ds(:1).eq.'U')then
                       if(units.eq.'KCAL')then
                                 units='KJ'
                                 goto 8990
                       else if(units.eq.'KJ')then
                                 units='KCAL'
                                 goto 8990
                       endif
             goto 8990
             endif


      IF(DS(1:1).EQ.'Y')THEN
```

```
          CALL EDIT(CS,E,B,AS,A,N,RS22,LS,RS,TEXT,BD,TS,IX,FORM,FORMP,
     +    CPINF,SYMM,SYMR,BFLAG,FLAG,BLFLAG,SCOR,SCORP,NSAVE,IG,NE,
     +    IE,NGRPT,IFIRST,ILAST,NATOM,KOUNT,SYMS,TEXTS,KOUNTS,SPECP,
     +    IHELP,LENF,LENFP,EP,NATOMP,IP,NP,ISOURCE,DELB,DOCTYPE,KDOCT,
     +    DELTA,NED,BSAVE,DOCFIL,FILNM,DOC)
          GOTO 8990
      ENDIF
C************************************************************************
C
C SPECIES MENU OPTIONS
55      CALL CLS
        IF(IHELP.EQ.0.AND.BFLAG.EQ.0)THEN
        WRITE(*,*)'SPECIES OPTIONS MENU: R, F, P, L, H, Q, X, S, U'
        GOTO 56
        ENDIF
        IF(IHELP.EQ.0.AND.BFLAG.NE.0)THEN
      WRITE(*,*)'SPECIES OPTIONS MENU: R, F, M, P, L, H, Q, X, S, U'

        goto 56
        endif
      WRITE(*,*)'                    SPECIES ESTIMATION OPTIONS MENU'
             if(units.eq.'KCAL')then
                           write(*,*)'           Units:  Kcal/mol  ( cal/mol K )'
             else if(units.eq.'KJ')then
                           write(*,*)'           Units:  KJ/mol  (J/mol K)'
             endif
      WRITE(*,*)'           ___________________________________________
     +___________'
      write(*,*)' '
      WRITE(*,*)'           R      -  CALCULATE RADICAL BASED UPON PREV
     +IOUS SPECIES'
      WRITE(*,*)'           F      -  ADD CURRENT SPECIES TO OUTPUT FIL
     +E'

      WRITE(*,*)'           H      -  TOGGLE (SHORT/LONG) MENU'
      WRITE(*,*)'           P      -  RETURN TO PREVIOUS SPECIES SCREEN
     +'
      IF(BFLAG.NE.0)THEN
      WRITE(*,*)'           M      -  RETURN TO PREVIOUS PARENT MOLECUL
     +E'
      ENDIF
      WRITE(*,*)'           RETURN -  ENTER A NEW SPECIES'
      WRITE(*,*)'           L      -  LIST GROUPS'
      WRITE(*,*)'           Q      -  RETURN TO MAIN MENU {choose "F" '
      WRITE(*,*)'                     first; current species is discard
     +ed'
      write(*,*)'                     when you return to MAIN menu}'
      WRITE(*,*)'           X      -  EXIT PROGRAM'
      WRITE(*,*)'           S      -  GET SPECIES FROM { *.DOC } FILE'
      WRITE(*,*)'           ?      -  HELP  '
```

```
             if(units.eq.'KCAL')then
                         write(*,4940)
4940  format(1x,12x,'U',6x,'-  CHANGE CURRENT UNITS TO : KJ/mol  (J/mol
     +K)')
             else if(units.eq.'KJ')then
                          write(*,4941)
4941  format(1x,12x,'U',6x,'-  CHANGE CURRENT UNITS TO : Kcal/mol  (cal/
     +mol K)')
             endif
        CALL LINES(0,1)


56           CONTINUE
             INQUIRE(7,OPENED=OPNED)
             IF(OPNED)THEN
             CALL LINES(0,1)
             BACKSPACE(7)
             READ(7,82)SPECL
82           FORMAT(A10)
             WRITE(*,*)' LAST SPECIES FILED: ',SPECL
             CALL LINES(0,1)
             ENDIF
       WRITE(*,*)'          enter option : { ENTER NEW SPECIES }'
       READ(*,3) DS
       CALL READCHAR(DS)
         CALL DEBLNK(DS,NUL,LEN,0)

* change units
             if(ds(:1).eq.'U')then
                         if(units.eq.'KCAL!)then
                                     units='KJ'
                                     goto 55
                         else if(units.eq.'KJ')then
                                     units='KCAL'
                                     goto 55
                         endif
             goto 55
             endif


C*******************************************************************************
        IF(DS.EQ.' '.OR.DS.EQ.'NEW')GOTO 10000
C*******************************************************************************
************************ HELP **************************************************
             IF(DS.EQ.'?')THEN
                         WRITE(*,4343)
                         READ(*,52)OPTION
                         CALL LRDCHAR(OPTION)
                         IF(OPTION.EQ.' '.OR.OPTION.EQ.'Q')GOTO 55
            IF(OPTION(:1).EQ.'R'.OR.OPTION.EQ.'F'.OR.OPTION.EQ.'H'.OR.OPTION
     +.EQ.'P'.OR.OPTION.EQ.'M')THEN
                         CALL HELP(HLP(4),OPTION)
                         GOTO 55
            ELSE IF(OPTION.EQ.'L'.OR.OPTION.EQ.'Q'.OR.OPTION.EQ.'X'.OR.
     +OPTION.EQ.'S'.OR.OPTION.EQ.'O'.OR.OPTION.EQ.'?')THEN
                          CALL HELP(HLP(5),OPTION)
                          GOTO 55
             ENDIF
             GOTO 55
             ENDIF

****************************************************************************
        IF(DS.EQ.'S')THEN
                NUMIN(:1)='6'
                GOTO 87
        ENDIF
        IF(DS.EQ.'Q')THEN
                GOTO 27
        ENDIF

C*********************************************************************
        IF(DS.EQ.'H')THEN
                IF(IHELP.EQ.0)THEN
                IHELP=1
                GOTO 55
                ENDIF
                IF(IHELP.EQ.1)THEN
                IHELP=0
                GOTO 55
                ENDIF
        ENDIF
C*********************************************************************
      IF (DS.EQ.'P') GOTO 8990
        IENTRY=0
      IF (DS.EQ.'L') CALL LISTGR(AS,A,NGRPT,IFIRST,ILAST,IENTRY,IHELP,
     +NE1,IE1)
      IF (DS.EQ.'L')GOTO 55
      IF (DS .EQ. 'X' .OR. DS .EQ. 'EXIT') GOTO 9200
C*********************************************************************
C
C**************** M - RESET TO PARENT MOLECULE ************************
C
4567  CONTINUE
      IF ((DS .EQ.'M').AND.(BFLAG.NE.0))THEN
        IF(ISOURCE.EQ.0)THEN
             NROT=NROTS
          N=NSAVE
          BFLAG=0
          DO 857 NG=1,N
               CS(1,NG)=CS(2,NG)
857            E(NG)=EP(NG)
          DO 866 Y=1,9
866       B(1,Y)=B(2,Y)
          SYMM=SYMS
          SCOR=SCORP
          B(1,2)=B(1,2)-SCOR
          KOUNT=KOUNTS
          DO 867 IZ1=1,KOUNT
```

```
867        TEXT(IZ1)=TEXTS(IZ1)
           FORM=FORMP
           LENF=LENFP
           NATOM=NATOMP
                DO 548 KJ=1,4
                IE(KJ)=IP(KJ)
548             NE(KJ)=NP(KJ)
           GOTO 8990
        ENDIF
        IF(ISOURCE.EQ.1)THEN
        IENTER=1
        SYMS=0
        SYMM=0
        SYMR=0
        FORM=' '
        FORMP=' '
        N=0
        NSAVE=0
        LS=' '
        RS=' '
        SPEC=' '
        NATOM=0
        NATOMP=0
        FLAG=0
        FLG=0
        BFLAG=0
        BLFLAG=0
        SCOR=0.0
        SCORP=0.0
        BD=0.0
        DELB=0.0
        DO 869 I=1,60
        CS(1,I)=' '
869     CS(2,I)=' '
        DO 873 I=1,50
        E(I)=0
873     IX(I)=0
        DO 871 I=1,4
        IE(I)=' '
871     NE(I)=0
             COMMAND=' '
      CALL LOADSP(AS,A,IFIRST,ILAST,NGRPT,CS,B,N,NSAVE,SYMM,SYMS,
     +SYMR,SCOR,SCORP,FORM,FORMP,SPECP,IE,NE,NATOM,FLAG,BFLAG,FLG,LS,RS,
     +RS22,FILST,DOCFIL,IX,E,LENF,EP,FILNM,DOC,NDOC,TEXT,TEXTS,KOUNT,
     +IOP,IDOC,NSPEC,BD,LENFP,IENTER,JC,DELB,DOCTYPE,KDOCT,DELTA)
                IF(FLG.EQ.1)GOTO 55
        DO 54631 JCOUNT=1,N
        IF((IX(JCOUNT).GE.IFIRST(4)).AND.(IX(JCOUNT).LE.ILAST(4)))THEN
                BFLAG=1
                GOTO 54631
        ENDIF
        IF((IX(JCOUNT).GE.IFIRST(3)).AND.(IX(JCOUNT).LE.ILAST(3)))THEN
                FLAG=1
                GOTO 54631
```

```
         ENDIF
         IF(FLAG.EQ.1)GOTO 54631
       FLAG=0
54631  CONTINUE
        ISOURCE=1
           CPINF = ((3.0*NATOM) - (2.+(FLOAT(NROT)/2.0)))*1.987
         CP2K = B(1,7) + .37*(CPINF - B(1, 7))
                 GOTO 8990
         ENDIF
       ENDIF
C******************* END OF M BLOCK*************************************
C
C****************** F - FILE BLOCK *************************************
       IF (DS .EQ. 'F'.OR. DS .EQ. 'FILE') THEN

19020    X=1
         PHASE='G'
              COMMAND=' '
       CALL FILE(LS,RS,B,CP2K,REF,REF3,IE,NE,CS,IX,MM1,E,FLAG,BFLAG
     +,TEXT,BLFLAG,FORM,FORMP,SPECP,RS22,CPINF,SYMR,SYMS,SYMM,BD,TS,FIL
     +,N,KOUNT,LENF,LENFP,DELB,DOCTYPE,KDOCT,DELTA,IOP)
       ENDIF
       IF (DS(1:1) .EQ. 'F' .OR. DS .EQ. 'FILE') GOTO 55
C********************* END OF F - FILE BLOCK *******************************
C
C
C**************************************************************************
         IF(DS.EQ.' '.OR.DS.EQ.'NEW')GOTO 10000
C
C***************** BEGIN  R - CALCULATE RADICAL BLOCK **********************
656       NUM=0
       IF((DS.EQ.'R'.AND.BFLAG.GE.1).OR.(DS.EQ.'R'.AND.FLAG.EQ.1))THEN
           CALL CLS
         IF(IHELP.EQ.0.AND.FLAG.EQ.0)THEN
         WRITE(*,*)'RADICAL OPTIONS: 1 - DIRADICAL, 2 - NEW RADICAL'
         GOTO 456
         ENDIF
         IF(IHELP.EQ.0.AND.FLAG.EQ.1)THEN
         WRITE(*,*)'RADICAL OPTIONS: 1 - DIRADICAL'
         GOTO 456
         ENDIF
           WRITE(*,2525)
2525   FORMAT(35X,'RADICAL OPTIONS')
           WRITE(*,2526)
2526   FORMAT('0',30X,30('-'))
           WRITE(*,2527)
2527   FORMAT('0',30X,'1  -  CALCULATE DIRADICAL FROM LAST RADICAL')
         IF(FLAG.NE.1)THEN
           WRITE(*,2528)
         ENDIF
2528   FORMAT('0',30X,'2  -  CALCULATE DIFFERENT RADICAL FROM PARENT')
           WRITE(*,2529)
2529   FORMAT('0',30X,'Q  -  QUIT { RETURN TO PREVIOUS MENU }')
       WRITE(*,*)'                                 ?  -  HELP'
```

```
          CALL LINES(0,3)
          WRITE(*,*)'                             enter option : { Q }'
456          READ(*,51)NUMIN
          CALL READCHAR(NUMIN)
          IF((NUMIN(:1).EQ.'Q').OR.(NUMIN(:1).EQ.' '))GOTO 55
*********************** HELP *********************************************
             IF(NUMIN.EQ.'?')THEN
                           WRITE(*,4343)
                           READ(*,52)OPTION
                           CALL LRDCHAR(OPTION)
                           IF(OPTION.EQ.' '.OR.OPTION.EQ.'Q')GOTO 55
                           CALL HELP(HLP(6),OPTION)
                           GOTO 656
             ENDIF

*****************************************************************************
          CALL READNUM(NUMIN,RNUM,NUM,FLG)
        IF(FLAG.EQ.1.AND.NUM.EQ.2)FLG=1
              IF(FLG.EQ.1)THEN
                   FLG=0
                   GOTO 55
              ENDIF
c                                                            column 72 ]

          IF((NUM.EQ.1.AND.BFLAG.GE.2).OR.(NUM.EQ.1.AND.FLAG.EQ.1.AND.
     +BFLAG.NE.0).AND.ERRORCK.EQ.' ')THEN
                  CALL CLS
                  WRITE(*,*)'      YOU CANNOT CALCULATE A TRIRADICAL'
                  CALL LINES(0,5)
                  WRITE(*,*)'       { hit return to continue }'
                  read(*,51)nul
                  goto 55
          endif
          IF(NUM.EQ.1)THEN
              BFLAG=2
              GOTO 72
          ENDIF
          IF(NUM.EQ.2)THEN
              BFLAG=1
              DO 865 Y=1,9
865           B(1,Y)=B(2,Y)
              SYMM=SYMS
              SCOR=SCORP
              GOTO 72
          ENDIF
          IF ((NUM.GE.3).OR.(NUM.LE.0))GOTO 55
       ENDIF
72     IF (DS .NE. 'R') GOTO 55
        CALL CLS
        IF(IHELP.EQ.0)THEN
        WRITE(*,*)'RADICAL MENU OPTIONS:'
        GOTO 10100
        ENDIF
```

```
C ************* RADICAL MENU OPTIONS ****************************************

      WRITE(*,*) '             RADICAL CALCULATION MENU '
      WRITE(*,*) '         {radical by loss of H from parent}
    +'
      WRITE(*,*)'    _______________________________
    +_____'
      CALL LINES(0,1)

      WRITE(*,*)'       Q      -  RETURN TO PREVIOUS MENU'
          WRITE(*,*)'      ?      -  HELP'
      WRITE(*,*)' '
10100     continue
           IENTRY=4
      CALL LISTGR(AS,A,NGRPT,IFIRST,ILAST,IENTRY,IHELP,NE1,IE1)
          write(*,*)' '
          write(*,*)'  ENTER BOND INCREMENT NAME ONLY'
            WRITE(*,*)'      enter option : { Q }'
      READ(*,3) TS
      CALL READCHAR(TS)
C******************* END OF  RADICAL BD MENU ***************************
C*
*********************** HELP *********************************************
          IF(TS.EQ.'?')THEN
                    WRITE(*,4343)
                    READ(*,52)OPTION
                    CALL LRDCHAR(OPTION)
                    IF(OPTION.EQ.' '.OR.OPTION.EQ.'Q')GOTO 72
                    CALL HELP(HLP(6),OPTION)
                    DS='R'
                    GOTO 72
          ENDIF

***************************************************************************
C****************** Q - QUIT OPTION  RETURN TO PREVIUOS MENU *******
      IF((TS(:1).EQ.'Q').OR.(TS(:1).EQ.' '))GOTO 55

C****************** BEGIN OTHER INPUT ****************************
      IF (TS. EQ. 'O' .OR. TS .EQ. 'o') THEN
       WRITE(*,*)'OPTION NOT YET AVAILABLE'
       WRITE(*,*)' HIT RETURN TO CONTINUE'
       READ (*,3)NUMIN
       GOTO 10301
      ENDIF
C******************** END OTHER INPUT ***************************
C************ BEGIN SEARCH DATABASE FOR BOND DISSOCIATION GROUP **
     ISTAT=0
     MM1=14
     NUL1(1,1)(:MM1)=TS(:MM1)
     CALL FINDGR(AS,NGRPT,NUL1,I9,1,ISTAT,MM1)
     IF(ISTAT.EQ.1)THEN
         ISTAT=0
            GOTO 10100
     ENDIF
```

```
      IF((I9(1).GT.ILAST(4)).OR.(I9(1).LT.IFIRST(4)))THEN
        CALL CLS
        WRITE(*,*)'           ERROR :  ENTER ONLY'
        WRITE(*,*)'         BOND DISSOCIATION GROUPS'
        WRITE(*,*)'             ON THIS SCREEN'
        CALL LINES(0,3)
        WRITE(*,*)'         { hit return to continue }'
        call lines(0,3)
        READ(*,3)NUL
        DS='R'
        GOTO 72
      ENDIF
      IF(NUM.EQ.2)THEN
        N=NSAVE+1
        GOTO 365
      ENDIF
      N=N+1
365   CS(1,N)=AS(I9(1))
      E(N)=1
      BD=A(I9(1),1)

          DO 1300 K = 2,9
          B(1,K)=B(1,K)+A(I9(1),K)
               IF(A(I9(1),9).EQ.0.0.AND.A(I9(1),8).NE.0.0)THEN
                           B(1,9)=0.0
               ENDIF
1300    CONTINUE

C************ ESTIMATE PROPERTIES FOR THE RADICAL ****************
10300  CONTINUE
      B(1,1)=B(1,1)+BD-52.1
        CALL CLS
10302        WRITE(*,*)'PARENT =  ',FORMP(:LENFP)
        WRITE(*,*)' '
        WRITE(*,*)'BOND TYPE =  ',TS
        WRITE(*,*)' '
10301   WRITE(*,*)'enter the symmetry number for the radical'
        READ(*,3) NUMIN
        CALL READCHAR(NUMIN)
        IF(NUMIN(1:1).EQ.'Q')THEN
                DS='M'
                GOTO 4567
        ENDIF
             IF(NUMIN(:1).EQ.'?')THEN
                           OPTION='SYMMETRY'
                           CALL HELP(HLP(3),OPTION)
                           CALL CLS
                           GOTO 10302
             ENDIF
        CALL READNUM(NUMIN,RNUM,SYMR,FLG)
        IF(FLG.EQ.1)THEN
           FLG=0
                DS='M'
           GOTO 4567
        ENDIF
        B(1,2)=B(1,2)+SCOR
        SCOR=1.987*ALOG(SYMR*1.0)
10333        B(1,2)=B(1,2)-SCOR
        IF(FLAG.EQ.0.AND.BFLAG.EQ.0)THEN
                BFLAG=1
                GOTO 2345
        ENDIF
        IF(FLAG.EQ.1.AND.BFLAG.EQ.0)THEN
                BFLAG=2
                GOTO 2345
        ENDIF
      IF(BFLAG.LE.1)THEN
        BFLAG=1
        GOTO 2345
      ENDIF
2345  CONTINUE
      IF (DS(1:1) .EQ. 'R' .AND. FLAG .EQ. 0) THEN
        WRITE(*,*)' '
        WRITE(*,*) 'Enter radical name(ID)'
        READ(*,3) RS
        CALL READCHAR(RS)
        IF(RS.EQ.'Q'.OR.RS.EQ.'QUIT')THEN
                DS='M'
                GOTO 4567
        ENDIF
             IF(RS(:1).EQ.'?')THEN
                          OPTION='ENTERSPEC'
                          CALL HELP(HLP(3),OPTION)
                          CALL CLS
                          GOTO 2345
             ENDIF
        ENDIF
      IF (DS(1:1) .EQ. 'R' .AND. FLAG .EQ. 1) THEN
        WRITE(*,*)' '
        WRITE(*,*) 'Enter diradical name(ID)'
        READ(*,3) RS22
        CALL READCHAR(RS22)
        IF(RS22.EQ.'Q'.OR.RS.EQ.'QUIT')THEN
                DS='M'
                GOTO 4567
        ENDIF
             IF(RS22(:1).EQ.'?')THEN
                          OPTION='ENTERSPEC'
                          CALL HELP(HLP(3),OPTION)
                          CALL CLS
                          GOTO 2345
             ENDIF
        ENDIF
2346         CONTINUE
        write(*,*)' '
        WRITE(*,*)'enter the elemental formula for this radical'
        write(*,*)'( example: for PHENYL enter " C6H5 " )'
        WRITE(*,*)'(          for BENZYL enter " C6H5CH2 " )'
        WRITE(*,*)'(         FOR VINYL  enter " C2H3 " )'
        CALL LINES(0,2)
        WRITE(*,*)'              4 ELEMENTS MAXIMUM'
        CALL LINES(0,2)
        read(*,52)FORM
        CALL LRDCHAR(FORM)
             IF(FORM(:1).EQ.'?')THEN
                          OPTION='FORM'
                          CALL HELP(HLP(3),OPTION)
                          CALL CLS
                          GOTO 2346
             ENDIF
        CALL DEBLNK(FORM,DUMMY,LENF,0)
        IF(FORM.EQ.' '.OR.FORM(:1).EQ.'Q')THEN
                DS='M'
                GOTO 4567
        ENDIF
             IF(ERRORCK.EQ.' ')THEN
                          iform=0
             ELSE
                          IFORM=1
             ENDIF
        CALL FORMINT(FORM,LENF,IE,NE,NATOM,iform)
             if(iform.lt.0)goto 2346
             IF(ERRORCK.NE.' ')THEN

22601        CALL CLS
             WRITE(*,*)' how many rotors in this species ( 0 )'
             read(*,'(a)')numin
             call readchar(numin)
             call dblnk(numin,lnum,0)
             if(numin.eq.'Q')goto 22601
             if(numin.eq.'?')then
                          option='NROTOR'
                          CALL HELP(HLP(3),OPTION)
                          GOTO 22601
             ENDIF
             if(numin.eq.' '.or.numin.eq.'0')then
                          nrot=0
             else
                          call readnum(numin,rnum,NROT,kflg)
                          if(kflg.ne.0)goto 22601
             endif
             ENDIF

             IF(ERRORCK.EQ.' ')THEN
        CALL LINES(0,2)
      WRITE(*,*) ' number of Atoms for This Species =',NATOM
      WRITE(*,*)' '
        WRITE(*,*)'    IF THIS IS INCORRECT ENTER " Q "'
        WRITE(*,*)'           TO BEGIN AGAIN'
        WRITE(*,*)'  OTHERWISE HIT RETURN TO CONTINUE'
      READ(*,3)  FS
      CALL READCHAR(FS)
      IF (FS(1:1) .EQ. 'Q' )THEN
                DS='M'
                GOTO 4567
      ENDIF

22602        CALL CLS
             WRITE(*,*)' how many rotors in this species ( 0 )'
             read(*,'(a)')numin
             call readchar(numin)
             call dblnk(numin,lnum,0)
             if(numin.eq.'Q')goto 22602
             if(numin.eq.'?')then
                          option='NROTOR'
                          CALL HELP(HLP(3),OPTION)
                          GOTO 22602
             ENDIF
             if(numin.eq.' '.or.numin.eq.'0')then
                          nrot=0
             else
                          call readnum(numin,rnum,NROT,kflg)
                          if(kflg.ne.0)goto 22602
             endif
             ENDIF
      IF (DS .EQ. 'R' .AND. FLAG .EQ. 0 )GOTO 8990
C****************END RADICAL ESTIMATION BLOCK*******************
C*
C******************** IF RETURN IS SELECTED : NEW SPECIES********
      GOTO 10000
C******************** GOTO 10000 :ENTER NEW SPECIES *************
C*
C*************** IF  X - EXIT WAS SELECTED, FILES ARE CLOSED ****
C***************           AND THE PROGRAM TERMINATES ***********
99999 CONTINUE
9200  CONTINUE

             INQUIRE(7,OPENED=OPNED)
             IF(OPNED)THEN
                          ENDFILE(7)
                     CLOSE(7)
             ENDIF

             INQUIRE(25,OPENED=OPNED)
             IF(OPNED)CLOSE(25)

             INQUIRE(41,OPENED=OPNED)
             IF(OPNED)THEN
                          ENDFILE(41)
                     CLOSE(41)
             ENDIF

             INQUIRE(52,OPENED=OPNED)
             IF(OPNED)THEN
19992        read(52,51,end=19993)nul
             goto 19992
```

```
19993      BACKSPACE(52)
                      NUL=' '
                      WRITE(52,'(A70)')NUL
                      WRITE(52,'(A70)')NUL
                      ENDFILE(52)
                      CLOSE(52)
           ENDIF

      STOP ' '
      END
```

APPENDIX V-B

FORTRAN SOURCE CODE FOR

THERMLST

```
             PROGRAM THERMLST
c
        IMPLICIT REAL*8 (A-H,O-Z)
        DIMENSION  TRA(3),F(14),CPL(50)
        CHARACTER*1 PHASE
        CHARACTER*2 IE(4), BLK1
        CHARACTER*3 BLK2
             INTEGER NE(4)
        CHARACTER*4 IFLAG,INAME,ID(10),REF(2)
        CHARACTER*6 REF3
             CHARACTER*6 XNAME
             CHARACTER*1 CFLAG,CNROT
             CHARACTER*70 FILIN,FILOUT,DUMMY,S1,title
             character*14 nul,UNITS,FUNITS,FORM
             LOGICAL EXST1,EXST2,EXST
             COMMON/UNIT/UNITS,FORM,CFLAG
        DATA  IFLAG,BLK1,BLK2/'END ',' ' ,'   '/

c   *****    ***********
C       THE INPUT IS TAKEN  DIRECTLY FROM THE THERMO DATA USED FOR
C       CHEMKIN
C
C
C       NPTS= NUMBER OF TEMPERATURE POINTS FOR A GIVEN SET
C       INAME= NAME OF SPECIES (FIRST 4 LETTERS)
C       XNAME= NAME OF SPECIES (NEXT 6 LETTERS)
C       GMW= GRAM MOLECULAR WEIGHT OF SPECIES
C       HF298= HEAT OF FORMATION AT 298 K (KCAL/MOLE) (JANAF INPUT)
C       T (I) = TEMP IN DEGREE K
C       CPL(I) = HEAT CAPACITY
C
C
C       FILE ASSIGNMENTS ARE
C           FILE 1 - INPUT DATA FITTED IN NASAFIT FORMAT FOR
C                    CHECKING 9SAME FORMAT AS USED FOR INTERPRETER)
C           FILE 3 - OUTPUT LISTING H AND S 298 K AND CP'S OVER TEMP
C                    RANGES
C
C
C****************************************************************************
C
C


*  DEFINE DEFAULT UNITS
             FUNITS='KCAL'
             UNITS='KCAL'
             FORM='LONG'
             INQUIRE (FILE='THUNITS.TMP',EXIST=EXST)
             IF(EXST)THEN
                          OPEN(13,FILE='THUNITS.TMP',STATUS='OLD')
                          READ(13,70)DUMMY
                          CALL DEBLNK(DUMMY,S1,LEN,0)
                          UNITS='KCAL'
                          IF(DUMMY(:2).EQ.'KJ')UNITS='KJ'
                          READ(13,70,END=987)DUMMY
                          CALL DEBLNK(DUMMY,S1,LEN,0)
                          IF(DUMMY(:5).EQ.'SHORT')THEN
                                    FORM='SHORT'
                          ENDIF
987                       CLOSE(13,STATUS='DELETE')
                ENDIF

110             CONTINUE
                DO 111 I=1,30
111             WRITE(*,*)' '
                WRITE(*,*)'ENTER INPUT FILENAME {.DAT}'
                READ(*,70)FILIN
                CALL FNAME(FILIN,'DAT')
                INQUIRE(FILE=FILIN,EXIST=EXST1)
                if(exst1)then
                          OPEN(1,FILE=FILIN,STATUS='OLD')
                          goto 120
                endif
                write(*,*)' cannot find input file:'
                write(*,*)filin
                write(*,*)' hit return to continue or "Q" to abort'
                read(*,3)nul
                if(nul.eq.'q'.or.nul.eq.'Q')stop 2
                goto 110

120             WRITE(*,*)'ENTER OUTPUT FILNAME {.LST}'
                READ(*,70)FILOUT
                CALL FNAME(FILOUT,'LST')
                INQUIRE(FILE=FILOUT,EXIST=EXST2)
                ktrl=0
                IF(EXST2)THEN
                          WRITE(*,*)'FILE ALREADY EXISTS'
                          WRITE(*,70)FILOUT
                          WRITE(*,*)' '
                          WRITE(*,*)' 1 - OVERWRITE'
                          WRITE(*,*)' 2 - APPEND'
                          WRITE(*,*)' 3 - ENTER DIFFERENT FILENAME'
                          WRITE(*,*)' '
                          WRITE(*,*)'     enter option { 1 }'
                          read(*,3) nul
                          if(nul.eq.'2')ktrl=1
                          if(nul.eq.'3')goto 120
                ENDIF

                if(units.ne.'KJ')units='KCAL'
                if(ktrl.eq.0)THEN
                          CALL CLS
                          WRITE(*,*)' what units do you want for output'
                          write(*,*)' 1- Kcal/mol   or  2- KJ/mol'
                          write(*,*)'   enter option: {1}'
                          read(*,70)dummy
                          call lrdchar(dummy)
```

```
                   call dblnk(dummy,len,0)
                   if(dummy(:2).eq.'KJ'.or.dummy(:1).eq.'2')units='KJ'
                   OPEN(3,FILE=FILOUT,STATUS='UNKNOWN')
                   if(units.eq.'KCAL')THEN
                             write(3,*)'UNITS:KCAL'
                   else if(units.eq.'KJ')then
                             write(3,*)'UNITS:KJ'
                   endif

          else if(ktrl.eq.1)then
          open(3,file=filout,status='old')
                   UNITS='KCAL'
                   READ(3,3,END=140)NUL
                             IF(NUL(:8).EQ.'UNITS:KJ'.or.NUL(:9).eq.
    +             ' UNITS:KJ')UNITS='KJ'
130                read(3,3,end=140)nul
                   goto 130
140                backspace(3)
                   backspace(3)
          endif

3         FORMAT(A14)
70        FORMAT(A70)

          CALL CLS
          write(*,*)' which table format do yo prefer?'
          write(*,*)' 1 - LONG format:  Cp 300 - 5000 K'
          write(*,*)' 2 - SHORT format: Cp 300 - 1500 K'
          write(*,*)'   enter option: {1}'
          read(*,70)dummy
          call lrdchar(dummy)
          call dblnk(dummy,len,0)
          if(dummy(:5).eq.'SHORT'.or.dummy(:1).eq.'2')THEN
                   FORM='SHORT'
          ELSE
                   FORM='LONG'
          ENDIF

  10  READ (1,71) INAME
  12  READ (1,71) TNAME
C
C

C  ** READ DATA IN FROM FILES USED BY CHEMKIN, NOT JANAFIT.**
C      OR THERCOEF

C    GET ID FOR PRINTOUT

      WRITE(*,100)
 100 FORMAT(1X,'enter a title line for the output file (blank = no
   + title)')
      READ(*,70) TITLE
 101  FORMAT(15A4)
 650  ICOUNT=1
```

```
        WRITE(3,'(1X,A70)') TITLE
  102   FORMAT (1H1,15A4)
        WRITE(3,119)
  119 FORMAT (1X,'SPECIES',4X,'HF(298)',2X,'S(298)',5X,'CP300',3X,
     +'CP400',3X,'CP500',3X,'CP600',3X,'CP800',3X,'CP1000',2X,'CP1500',
     + 2X,'DATE  REF  ',8X,' ELEMENTS ')
             IF(FORM.EQ.'LONG')WRITE(3,187)
187          FORMAT (1X,31X,'CP2000  CP2500  CP3000  CP3500  CP4000  CP5000 ')
655          CONTINUE
             CFLAG=' '
        READ(1,1010,END=900) INAME, XNAME, (REF(I),I=1,2),REF3,
     1 (IE(I),NE(I),I=1,4),PHASE,TRA(1),TRA(3),TRA(2),CNROT
*            TRA(2)=1500.0
             NROT=0
             IF(CNROT.EQ.'1')NROT=1
             IF(CNROT.EQ.'2')NROT=2
             IF(CNROT.EQ.'3')NROT=3
             IF(CNROT.EQ.'4')NROT=4
             IF(CNROT.EQ.'5')NROT=5
             IF(CNROT.EQ.'6')NROT=6
             IF(CNROT.EQ.'7')NROT=7
             IF(CNROT.EQ.'8')NROT=8
             IF(CNROT.EQ.'9')NROT=9
         IF (INAME.EQ.IFLAG) GOTO 900
         DO 658 J = 1,4
         IF (NE(J).EQ.0)  IE(J) = BLK1
  658    CONTINUE
C        TRA(1)=LOW TWMP   TRA(2)=MID-TEMP  TRA(3)=HIGH TEMP
  660   READ(1,1020,END=900) (F(I),I=1,14)
             ISAME=1
             DO 1959 KI=1,7
             IF(F(KI).NE.F(KI+7))ISAME=0
1959         CONTINUE
             IF(ISAME.EQ.1)CFLAG='*'
C       F(8)-F(14) ARE LOW TEMP COEFFICIENTS
C       F(1)-F(7)  ARE HIGH TEMP COEFFICIENTS
        CALL CHECK(INAME,XNAME,IE,NE,REF,REF3,PHASE,F,TRA,NROT)
        ICOUNT=ICOUNT+1
C       IF(ICOUNT.GE.55) GO TO 650
        GO TO 655

   71   FORMAT(A4)
 1010   FORMAT(A4,A6,2A4,A6,4(A2,I3),A1,2F10.3,1X,F8.0,4X,A1)
 1020   FORMAT(5E15.8,/,5E15.8,/,4E15.8)
  900   CONTINUE
             CLOSE(1,STATUS='KEEP')
             CLOSE(3,STATUS='KEEP')
             STOP
        END
C
```

```
      SUBROUTINE CHECK (INAME,XNAME,IE,NE,REF,REF3,PHASE,F,TRA,NROT)
          IMPLICIT REAL *8 (A-H,O-Z)
      DIMENSION TRA(3),F(14),CPL(50),CPJ(50),T1(20)
      CHARACTER*1 PHASE,CFLAG
      CHARACTER*2 IE(4)
      INTEGER NE(4)
      CHARACTER*4 REF(2),INAME
      CHARACTER*6 REF3
          CHARACTER*6 XNAME
          CHARACTER*14 UNITS,FORM
          COMMON/UNIT/UNITS,FORM,CFLAG
          T1(1)=300.
          T1(2)=400.
          T1(3)=500.
          T1(4)=600.
          T1(5)=800.
          T1(6)=1000.
          T1(7)=1500.
          T1(8)=2000.
          T1(9)=2500.
          T1(10)=3000.
          T1(11)=3500.
          T1(12)=4000.
          T1(13)=5000.
          NATOM=0
          ICARBON=0
          NCARBON=0
C
C-----FIRST COMPUTING HF298 AND S298 FROM THE INPUT PARAMETERS
C----  WANT TO COMPUTE H AND S AT 298 WITH THE LOW TEMP PARAMETERS
   7  T=298.16
      H=1.98717*T*((((F(12)*T/5.+F(11)/4.)*T+F(10)/3.)*T+F(9)/2.)*
     1 T+F(8)+F(13)/T)
       HF298=H/1000.
      S=1.98717*((((F(12)*T/4.+F(11)/3.)*T+F(10)/2.)*T +F(9))*T
     1 +F(8)*DLOG(T)+F(14))
      S298 = S
      ICP = 1
C
C


          DO 40  J= 1,13
          T = T1(J)
          IF(T.GT.TRA(3))THEN
                    CPL(J)=0.0
                    GOTO 40
          ENDIF
      TMID = TRA(2)
          IF(T.LE.TMID)THEN
                    CPL(J)=CPT(7,T1(J),F)*1.987
          ELSE
                    CPL(J)=CPT(0,T1(J),F)*1.987
          ENDIF
 40       CONTINUE
```

```
C
               IF(UNITS.EQ.'KJ')THEN
                              HF298=HF298*4.184
                              S298=S298*4.184
                              DO 87665 IK1=1,13
87665                         CPL(IK1)=CPL(IK1)*4.184
               ENDIF
          WRITE(3,165)CFLAG,INAME,XNAME,HF298,S298,(CPL(IJ),IJ=1,7),
     1    (REF(I),I=1,2),REF3,(IE(I),NE(I),I =1,4),PHASE,NROT
C
  165     FORMAT (A1,A4,A6,F8.2,1X,F8.2,2X,7(F7.2,1X),2A4,A6,
     1    3X,4(A2,I3,1X),A1,1X,I1)
               IF(FORM.EQ.'LONG')WRITE(3,167)(CPL(IJ),IJ=8,13)
167            FORMAT(30X,6(F7.2,1X))
   50     RETURN
          END
```

APPENDIX V-C

FORTRAN SOURCE CODE FOR

THERMRXN

```
             PROGRAM THERMRXN
$LARGE
*************************************************************************
*  WRITTEN BY:  ED RITTER
*
*  THERM REACTION INTERPRETER
*
*  CALCULATES THERMODYNAMIC PROPERTY CHANGES FOR REACTION
*************************************************************************
             IMPLICIT REAL*8(A-H,O-Z)
        REAL*8 DELH,DELS,DELG,SHJ(10),SSJ(10),KEQ(20),DNUM
             REAL*4 RNUM
       INTEGER NE1(4,400),LX(10),RX(10),IHELP,ICNT,
     +NLHS,NRHS,IERR,IND(70),JIND(70),RXFIL,FLG,NX(400)
             integer inlp(132),inbl(132),INEQ(132)
             DIMENSION ST1(20),STOT(20),HT1(20),HTOT(20),UTOT(20),T1(20),
     +RH298(10),SH298(10),RS298(10),SS298(10),AFAR(20),GORT(20),SOR(20),
     +COEF(10)
        CHARACTER*4 IE1(4,400)
        CHARACTER*14 RHS(10),LHS(10),SS(400),S(10),S3,OUTPUT
        CHARACTER*70 S1,S2,NUL,NUL2,NUL1,FILIN,FLNAME,DUMMY,KS(25),
     +CKFILE
             CHARACTER*127 S132
             CHARACTER PHSE(400)
             CHARACTER*4 REF1(2,400),S4(10)
             CHARACTER*6 REF2(400)
             CHARACTER*14 UNITS,NUMIN
             CHARACTER*2 ELEM(50)
        LOGICAL EXST, EXST1,OPNED,OPNED2,EXST2
             COMMON/CONFG6/UNITS
        COMMON/DELETE/NDEL,IDEL(500)
             COMMON/RTEMP/ST1,HT1,T1,NT
             COMMON/CONFG1/ELEM
             CALL CONFIG(ELEM,NFILES,KS,ISTAT,OUTPUT)
1023         RXFIL=0
             ICHEMKIN=0
             NCALL =0
             ICNT=0
             RC=1.987
             RJ=8.31361
*            DEFINE DEFAULT VECTOR OF TEMPERATURES
             T1(1)=298.15
             T1(2)=300.
             T1(3)=400.
             T1(4)=500.
             T1(5)=600.
             T1(6)=800.
             T1(7)=1000.
             T1(8)=1200.
             T1(9)=1500.
             T1(10)=2000.
             NT=10
C

             FLNAME='  '
             UNITS='KCAL'
             INQUIRE(FILE='THUNITS.TMP',EXIST=EXST1)
             IF(EXST1)THEN
                           OPEN(13,FILE='THUNITS.TMP',STATUS='OLD')
                           READ(13,14)UNITS
14                         format(a14)
                           CLOSE(13,STATUS='DELETE')
             ENDIF
             IHELP=1
             if(units.ne.'KJ')units='KCAL'
             CALL CLS
      WRITE(*,*)'        Thermodynamic Property Analysis for Reactions'
             call lines(0,3)
             CALL NASAINIT(SS,NX,IE1,NE1,ICNT,FLNAME)
99       CALL CLS
             IF(ICNT.EQ.0)THEN
                           OPEN(13,FILE='THUNITS.TMP',STATUS='UNKNOWN')
                           WRITE(13,14)UNITS
                           CLOSE(13,STATUS='KEEP')
                           STOP 'NO INPUT FILE SPECIFIED'
             ENDIF
100      CONTINUE
             ICHEMKIN=0
             ICR=0
             call cls
             do 7790 i1=1,10
7790         coef(i1)=1.0
             ITR=0
      IF(IHELP.EQ.0)WRITE(*,*)' enter Reaction expression: { "?" for hel
     +p }'
        IF(IHELP.EQ.1)THEN
             if(units.eq.'KCAL')then
                           write(*,*)' current units: Kcal/mol (cal/mol K)'
                           WRITE(*,*)' ___________________________________'
             else if(units.eq.'KJ')then
                           write(*,*)' current units: KJ/mol (J/mol K)'
                           WRITE(*,*)' _______________________________'
             endif

             CALL LINES(0,1)
        write(*,*)'     To Calculate Property Change of Reaction for'
        write(*,*)'     enthalpy (del H), entropy (del S), and '
        write(*,*)'     gibbs free energy (del G) '
        WRITE(*,*)'     enter a Reaction expression as follows:'
        call lines(0,1)
        write(*,*)'      " A+B=C+D " or " A+B=C " or " A=B+C "'
        write(*,*)'      (ad''n/abstr)   (ad''n/recomb)  (dissoc)'
        write(*,*)'      where A, B, C, etc.  are species defined '
        write(*,*)'     in the current Thermo D Base {*.DAT} file'
     write(*,*)' EXAMPLE:   C6H5CL=CYC6H5+ CL  (blanks are ignored)'
      call lines(0,1)
      write(*,*)'    NOTE:   Reaction expression MUST BALANCE'
      write(*,*)'            no quote marks are needed'
      call lines(0,1)
      write(*,*)'    enter " L " to list defined species'
      WRITE(*,*)'          " C " to process a "CHEMKIN" input file'
      write(*,*)'          " Q " to return to the MAIN MENU'
      WRITE(*,*)'          " ? " to supress this information'
          if(units.eq.'KJ')then
    write(*,*)'         " U " change units to KJ/mol (J/mol K)'
          else if(units.eq.'KCAL')then
    write(*,*)'         " U " change units to Kcal/mol (cal/mol K)'
          endif
      write(*,*)'            or enter Reaction expression here:'
      ENDIF
      READ(*,3)S1
3     FORMAT(A70)
      CALL READCHAR(S1)
          CALL SLEN(S1,LEN)
          CALL SUBSTIT(S1,CHAR(39),CHAR(32),LEN,1)
          CALL SUBSTIT(S1,CHAR(34),CHAR(32),LEN,1)
      CALL DBLNK(S1,LEN,0)
      IF(S1.EQ.' '.OR.S1.EQ.'Q')THEN
                   OPEN(13,FILE='THUNITS.TMP',STATUS='UNKNOWN')
                   WRITE(13,14)UNITS
                   CLOSE(13,STATUS='KEEP')
                   CLOSE(41)
                   STOP' '
      ENDIF
          IF(S1.EQ.'N')THEN
                   CLOSE(54)
                   GOTO 1023
          ENDIF
          if(S1.EQ.'?')THEN
            IF(IHELP.EQ.0)THEN
                  IHELP=1
                  GOTO 99
            ENDIF
            IF(IHELP.EQ.1)THEN
                  IHELP=0
                  GOTO 99
            ENDIF
      ENDIF
          IF(S1.EQ.'U')THEN
                   IF(UNITS.EQ.'KJ')THEN
                             UNITS='KCAL'
                             CALL CLS
                             GOTO 100
                   ELSE IF(UNITS.EQ.'KCAL')THEN
                             UNITS='KJ'
                             CALL CLS
                             GOTO 100
                   ENDIF
            ENDIF
       IF(S1.EQ.'L')THEN
            CALL CLS
            iline=0
            DO 2093 I=1,ICNT,3
            iline=iline+1
            WRITE(*,'(3(5X,A10))')(SS(I+K-1),K=1,3)
            if(iline.ge.21)then
                        iline=0
                        write(*,*)' '
                        pause '           { hit return to continue }'
            endif
2093        CONTINUE
            PAUSE 'hit return to continue'
              GOTO 100
       ENDIF
            IF(S1.EQ.'C')THEN
                        CALL CLS
                        WRITE(*,*)'    enter name of CHEMKIN format mechanism'
                        read(*,'(A70)')CKFILE
                        INQUIRE(FILE=CKFILE,EXIST=EXST2,OPENED=OPNED2)
                        IF(OPNED2)THEN
            PAUSE ' error : file is opened!   : hit return to continue'
                                    goto 100
                        ENDIF
                        if(exst2)then
                                    open(87,file=ckfile,status='old')
123                                 read(87,'(a70)',end=1324)nul
                                    if(nul(:5).ne.'REACT')goto 123
                                    ichemkin=1
                                    goto 10999
1324                                continue
                                    close(87)
      pause 'file error: "REACTIONS" not found: hit return to continue'
                                    goto 100
10999       IF(RXFIL.EQ.0)THEN
              CALL CLS
       WRITE(*,*)'   ENTER FILENAME FOR REACTION ANALYSIS OUTPUT'
              READ(*,3)NUL1
                        IF(NUL1.EQ.' '.OR.NUL1.EQ.'Q'.OR.NUL1.EQ.'q')THEN
                                                IU=0
                                                GOTO 100
                        ENDIF
              INQUIRE(FILE=NUL1,EXIST=EXST,OPENED=OPNED)
                        RXFIL=1
              IF(EXST)THEN
              WRITE(*,4)NUL1
              WRITE(*,*)'ALREADY EXISTS'
              CALL LINES(0,1)
              WRITE(*,*)'   1 - OVERWRITE'
              WRITE(*,*)'   2 - APPEND'
              WRITE(*,*)'   3 - ABORT'
              CALL LINES(0,1)
              WRITE(*,*)'        enter choice { 1 }'
```

```
                READ(*,3)NUL2
                CALL READCHAR(NUL2)
                            IF(NUL2.EQ.' '.OR.NUL2.EQ.'1')THEN
                                    OPEN(41,FILE=NUL1,STATUS='UNKNOWN')
                            ELSEIF (NUL2.EQ.'2')THEN
                                         OPEN(41,FILE=NUL1,STATUS='OLD')
                                                   IHJ=0
23451                                              CONTINUE
                                         IHJ=IHJ+1
                                         READ(41,3,END=14591)NUL2
                                                   GOTO 23451
14591                                    RXFIL=1
                                                   BACKSPACE(41)
                            ELSE
                                                   IU=0
                                                   GOTO 100
                                      ENDIF
                         ELSE
                       OPEN(41,FILE=NUL1,STATUS='UNKNOWN')
                         ENDIF
                         RXFIL=1
                         GOTO 10000
       ENDIF
                         else
                         PAUSE '  file does not exist: hit return to continue'
                         goto 100
                         endif
            ENDIF
            ICR=0
10000  CONTINUE
*_____________________________________________________________________
*
*       PROCESS A CHEMKIN INPUT FILE
*_____________________________________________________________________
*
            IF(ICHEMKIN.EQ.1)THEN
            ICR=ICR+1

                         READ(87,'(A)',END=10001)S132
                         call readchar(s132)
                         CALL DBLNK(S132,L132,1)
                         IF(S132(:1).EQ.'/')GOTO 10000
                         IF (S132(:3).EQ.'END')THEN
                                      CLOSE(87,STATUS='KEEP')
                                      CLOSE(41,STATUS='KEEP')
                                      RXFIL=0
                                      ICHEMKIN=0
                                      CALL CLS
                         WRITE(*,*)' CHEMKIN input file processing completed'
                         WRITE(*,*)ICR,' reactions processed'
                         PAUSE ' { hit return to continue }'
                                      GOTO 100
                         ENDIF
                         CALL INDXST(S132,'(',NLP,INLP,L132,1,FLG)
                           CALL INDXST(S132,CHAR(32),NBL,INBL,L132,1,FLG)
                           S1=' '
                           IF(NLP.EQ.1)THEN
                                        S1(:INLP(1)-1)=S132(:INLP(1)-1)
                           ELSE
                                        S1(:INBL(NBL-2))=S132(:INBL(NBL-2))
                           ENDIF
                           CALL DBLNK(S1,LS1,0)
              WRITE(*,*)' processing reaction # ',ICR
                           WRITE(*,'(1X,A)')S1(:LS1)
                           CALL SUBSTIT(S1,'-','=',LS1,1)
                           CALL INDXST(S1,'=',NEQ,INEQ,LS1,1,FLG)
                           IF(NEQ.NE.1)THEN
                                        CALL CLS
                                        WRITE(*,'(1X,A)')S1(:LS1)
                                        write(*,*) ' invalid reaction expression'
                                        WRITE(*,*)
      +                                 ' { hit return to continue or " Q " to quit }'
                                        read(*,'(a)')nul
                                        call readchar(nul)
                                        call dblnk(nul,lnul,0)
                                        if(nul.eq.'Q')then
                                                     rxfil=0
                                                     ichemkin=0
                                                     close(41,status='keep')
                                                     close(87,status='keep')
                                                     goto 100
                                        else
                                                     goto 10000
                                        endif
                           ENDIF
*                          INITIALIZE STOICHIOMETRIC COEFFICIENT ARRAY
                           do 97790 i1=1,10
97790                      coef(i1)=1.0
                           GOTO 10002
10001                      CALL CLS
                           ICHEMKIN=0
                           RXFIL=0
                           CLOSE(41,STATUS='KEEP')
                           CLOSE(87,STATUS='KEEP')
                           WRITE(*,*)' WARNING : " END " CARD MISSING'
                           WRITE(*,*)'  "END" CARD IS REQUIRED FOR CHEMKIN'
                           CALL LINES(0,2)
                           WRITE(*,*)ICR,' reactions processed'
                           PAUSE '{ hit return to continue }'
                           GOTO 100
              ENDIF
*______________________________________________________________________
*
10002   IKTL=0
             CALL DBLNK(S1,LEN,0)
        CALL INDXST(S1,'-',NOCCUR,IND,LEN,1,FLG)
        IF(NOCCUR.NE.0)THEN
                CALL CLS
                WRITE(*,4)S1
                WRITE(*,*)'    CONTAINS INVALID OPERATION "-" '
                WRITE(*,*)'      {hit return to continue}'
                read(*,3)nul
                GOTO 99
        ENDIF
        CALL INDXST(S1,'=',NOCCUR,IND,LEN,1,FLG)
        IF(NOCCUR.NE.1)THEN
                CALL CLS
                WRITE(*,4)S1
4               FORMAT(1X,A)
                WRITE(*,*)'    CONTAINS INVALID NUMBER OF "=" '
                WRITE(*,*)'      {hit return to continue}'
                read(*,3)nul
                GOTO 99
        ENDIF
        CALL INDXST(S1,'+',NOCR,JIND,LEN,1,FLG)
        NRHS=1
        NLHS=1
        DO 700 K=1,NOCR
                IF(JIND(K).LT.IND(1))NLHS=NLHS+1
                IF(JIND(K).GT.IND(1))NRHS=NRHS+1
700     CONTINUE
        IF(NLHS.GT.3)THEN
                WRITE(*,*)' TOO MANY REACTANTS : RE-ENTER REACTION'
                WRITE(*,*)'    {hit return to continue}'
                READ(*,3)NUL
                GOTO 99
        ENDIF
        IF(NRHS.GT.3)THEN
                WRITE(*,*)' TOO MANY PRODUCTS : RE-ENTER REACTION'
                WRITE(*,*)'    {hit return to continue}'
                READ(*,3)NUL
                GOTO 99
        ENDIF

C
C***********************************************************************
C             FIND LEFT HAND SIDE (LHS) SPECIES
C***********************************************************************
C
        IF(NLHS.EQ.1)LHS(1)=S1(:IND(1)-1)
        IF(NLHS.GE.2)THEN
                LHS(1)=S1(:JIND(1)-1)
                LHS(2)=S1(JIND(1)+1:IND(1)-1)
        ENDIF
        IF(NLHS.EQ.3)THEN
                LHS(2)=S1(JIND(1)+1:JIND(2)-1)
                LHS(3)=S1(JIND(2)+1:IND(1)-1)
        ENDIF
        IF(NLHS.GE.4)THEN
                LHS(3)=S1(JIND(2)+1:JIND(3)-1)
                LHS(4)=S1(JIND(3)+1:IND(1)-1)
        ENDIF
```

```
         IF(NLHS.EQ.5)THEN
                 LHS(4)=S1(JIND(3)+1:JIND(4)-1)
                 LHS(5)=S1(JIND(4)+1:IND(1)-1)
         ENDIF
C
C***********************************************************************
C                FIND RIGHT HAND SPECIES (RHS)
C***********************************************************************
C
         IF(NRHS.EQ.1)RHS(1)=S1(IND(1)+1:LEN)
         IF(NRHS.GE.2)THEN
                 RHS(1)=S1(IND(1)+1:JIND(NLHS)-1)
                 RHS(2)=S1(JIND(NLHS)+1:LEN)
         ENDIF
         IF(NRHS.EQ.3)THEN
                 RHS(2)=S1(JIND(NLHS)+1:JIND(NLHS+1)-1)
                 RHS(3)=S1(JIND(NLHS+1)+1:LEN)
         ENDIF
         IF(NRHS.GE.4)THEN
                 RHS(3)=S1(JIND(NLHS+1)+1:JIND(NLHS+2)-1)
                 RHS(4)=S1(JIND(NLHS+2)+1:LEN)
         ENDIF
         IF(NRHS.EQ.5)THEN
                 RHS(4)=S1(JIND(NLHS+2)+1:JIND(NLHS+3)-1)
                 RHS(5)=S1(JIND(NLHS+3)+1:LEN)
         ENDIF
C
C***********************************************************************
C
************    DETERMINE COEFFICIENTS IF ANY  ************
              DO 800 K=1,NLHS
              S(K)=' '
              S(K)=LHS(K)
              CALL INQNUM(S(K),IANS,NUMLEN,LSTR,0)
              IF(IANS.EQ.1.OR.IANS.EQ.2)THEN
                            NUMIN=' '
                            NUMIN(:LSTR)=S(K)(:LSTR)
                            CALL READNUM(NUMIN,RNUM,NUMM,IFLG)
                            IF(IFLG.NE.0)THEN
                                          WRITE(*,*)NUMIN
                                          STOP ' ILLEGAL STRING'
                            ENDIF
                            COEF(K)=COEF(K)*RNUM
              ELSEIF(IANS.EQ.3.OR.IANS.EQ.4)THEN
                            NUMIN=' '
                            NUMIN(:NUMLEN)=S(K)(:NUMLEN)
                            CALL READNUM(NUMIN,RNUM,NUMM,IFLG)
                            IF(IFLG.NE.0)THEN
                                          STOP 'ILLEGAL STRING'
                            ENDIF
                            COEF(K)=COEF(K)*DBLE(RNUM)
                            DUMMY=' '
                            DUMMY(:LSTR)=S(K)(:LSTR)
                            S(K)=' '
```

```
                        LEN10=LSTR-NUMLEN
                        S(K)(:LEN10)=DUMMY(NUMLEN+1:LSTR)
            ENDIF
800         CONTINUE
            DO 801 K=1,NRHS
            S(K+NLHS)=' '
            S(K+NLHS)=RHS(K)
            CALL INQNUM(S(K+NLHS),IANS,NUMLEN,LSTR,0)
            IF(IANS.EQ.1.OR.IANS.EQ.2)THEN
                        NUMIN=' '
                        NUMIN(:LSTR)=S(K+NLHS)(:LSTR)
                        CALL READNUM(NUMIN,RNUM,NUMM,IFLG)
                        IF(IFLG.NE.0)THEN
                                    STOP 'ILLEGAL STRING'
                        ENDIF
                        COEF(K+NLHS)=COEF(K+NLHS)*DBLE(RNUM)
            ELSEIF(IANS.EQ.3.OR.IANS.EQ.4)THEN
                        NUMIN=' '
                        NUMIN(:NUMLEN)=S(K+NLHS)(:NUMLEN)
                        CALL READNUM(NUMIN,RNUM,NUMM,IFLG)
                        IF(IFLG.NE.0)THEN
                                    STOP 'ILLEGAL STRING'
                        ENDIF
                        COEF(K+NLHS)=COEF(K+NLHS)*RNUM
                        DUMMY=' '
                        DUMMY(:LSTR)=S(K+NLHS)(:LSTR)
                        S(K+NLHS)=' '
                        LEN10=LSTR-NUMLEN
                        S(K+NLHS)(:LEN10)=DUMMY(NUMLEN+1:LSTR)
            ENDIF
801         CONTINUE
************************************************************************
*****************   REMOVE STRINGS WHICH APPEAR ON BOTH ****************
*****************         SIDES OF THE EQUAL SIGN       ****************
            DO 850 I1=1,NLHS
            DO 851 I2=1,NRHS
            IF(S(I1)(:10).EQ.S(I2+NLHS)(:10).AND.COEF(I1).EQ.COEF(I2+NLHS))
    +THEN
                        S(I1)=' '
                        S(I2+NLHS)=' '
                        LHS(I1)=' '
                        RHS(I2)=' '
                        COEF(I1)=0.0
                        COEF(I2+NLHS)=0.0
            ELSEIF(S(I1)(:10).EQ.S(I2+NLHS)(:10).AND.COEF(I1).NE.
    +COEF(I2+NLHS))THEN
                        DEL=COEF(I1)-COEF(I2+NLHS)
                        IF(DEL.GT.0.0)THEN
                                    COEF(I1)=DEL
                                    S(I2+NLHS)=' '
                                    RHS(I2)= ' '
                                    COEF(I2+NLHS)=0.0
                        ELSE
                                    COEF(I2+NLHS)=-1.0*DEL
```

```
                                        S(I1)=' '
                                        LHS(I1)=' '
                                        COEF(I1)=0.0
                                ENDIF
                        ENDIF
851                     CONTINUE
850                     CONTINUE
                        NSTR=NLHS+NRHS
                        NCOEF=NSTR
                        CALL DCREAL(COEF,NCOEF)
                        CALL CSTRING(S,NSTR)
                        CALL CSTRING(LHS,NLHS)
                        CALL CSTRING(RHS,NRHS)


**************************************************************************
*******  CHECK TO SEE IF ALL SPECIES IN THE REACTION ARE DEFINED *****
        DO 701 K=1,NLHS
        CALL DBLNK(S(K),LEN,0)
        J=0
750     J=J+1
        IF(J.GT.ICNT)THEN
                WRITE(*,*)'SPECIES NOT FOUND: ',LHS(K)
                WRITE(*,*)' {HIT RETURN TO CONTINUE}'
                READ(*,3)NUL
                GOTO 99
        ENDIF
        CALL DBLNK(SS(J),LEN2,0)
        IF(S(K)(:10).EQ.SS(J)(:10))THEN
                LX(K)=J
                GOTO 701
        ENDIF
        GOTO 750
701     CONTINUE
        DO 702 K=1,NRHS
        CALL DBLNK(S(K+NLHS),LEN,0)
        J=0
760     J=J+1
        IF(J.GT.ICNT)THEN
                WRITE(*,*)'SPECIES NOT FOUND: ' ,RHS(K)
                WRITE(*,*)' {HIT RETURN TO CONTINUE}'
                READ(*,3)NUL
                GOTO 99
        ENDIF
        CALL DBLNK(SS(J),LEN2,0)
        IF(S(K+NLHS)(:10).EQ.SS(J)(:10))THEN
                RX(K)=J
                GOTO 702
        ENDIF
        GOTO 760
702     CONTINUE
        IERR=0
        CALL MATBAL(NLHS,LX,NRHS,RX,IE1,NE1,COEF,IERR)
        IF(IERR.EQ.1)THEN
```

```
          WRITE(*,4)S1
     WRITE(*,*)' HIT RETURN TO RE-ENTER EXPRESSION OR "Q" TO QUIT'
          READ(*,3)NUL
          CALL READCHAR(NUL)
                IF(NUL.EQ.' '.AND.ICHEMKIN.EQ.1)THEN
                        GOTO 10000
                ELSEIF(NUL.EQ.'Q'.AND.ICHEMKIN.EQ.1)THEN
                        CLOSE(87,STATUS='KEEP')
                        CLOSE(41,STATUS='KEEP')
                        RXFIL=0
                        ICHEMKIN=0
                        GOTO 100
          ELSEIF(NUL.EQ.'Q')THEN
                OPEN(13,FILE='THUNITS.TMP',STATUS='UNKNOWN')
                WRITE(13,14)UNITS
                CLOSE(13,STATUS='KEEP')
                        STOP' '
                ENDIF
         GOTO 99
     ENDIF
C
C************************************************************************
C   CALCULATE PROPERTY CHANGE OF REACTION
C************************************************************************
C

8876 CONTINUE
        SMOLE=0.0
        RMOLE=0.0
        DO 7309 I1=1,NLHS
7309    SMOLE=SMOLE+COEF(I1)
        DO 7310 I1=1,NRHS
7310 RMOLE=RMOLE+COEF(I1+NLHS)
        DELTAN= RMOLE-SMOLE
        IF(ICHEMKIN.NE.1)THEN
                CALL CLS
        ENDIF
        DO 101 J=1,NT
        ST1(J)=0.
        STOT(J)=0.
        HT1(J)=0.
        HTOT(J)=0.
        UTOT(J)=0.
101     CONTINUE
*  REMEMBER HT1(I1) = (Hf(T1(I1))*T1(I1))/(1000.*R)
        DO 904 J=1,NLHS
        CALL READNASA(S(J),NX(LX(J)),FLNAME,IFLG)
        IF(IFLG.NE.0)THEN
                WRITE(*,*)S(J),' NOT FOUND'
                WRITE(*,*)' IN NASA FORMAT FILE'
                PAUSE '{ hit return to continue }'
                goto 99
        ENDIF
*            SAVING 1st TEMP'S PROPERTIES FOR ALL LHS SPECIES
```

```
                         SH298(J)=HT1(1)*COEF(J)
                         SS298(J)=ST1(1)*COEF(J)
             DO 905 J1=1,NT
*     REM SUBTRACT LEFT HAND SIDE FROM RIGHT  ( PROD - REACT )
*     REM : HTOT = H/RT
*           UTOT = U/RT
*           STOT = S/R
             HTOT(J1)=HTOT(J1)-(COEF(J)*HT1(J1))
             STOT(J1)=STOT(J1)-(COEF(J)*ST1(J1))
905          CONTINUE
904          CONTINUE

             DO 902 J=1,NRHS
             CALL READNASA(S(J+NLHS),NX(RX(J)),FLNAME,IFLG)
             IF(IFLG.NE.0)THEN
                         WRITE(*,*)S(J+NLHS),' NOT FOUND'
                         WRITE(*,*)' IN NASA FORMAT FILE'
                         PAUSE '{ hit return to continue }'
                         goto 99
             ENDIF
                         RH298(J)=HT1(1)*COEF(J+NLHS)
                         RS298(J)=ST1(1)*COEF(J+NLHS)
             DO 903 J1=1,NT
             HTOT(J1)=HTOT(J1)+(COEF(J+NLHS)*HT1(J1))
             STOT(J1)=STOT(J1)+(COEF(J+NLHS)*ST1(J1))
903          CONTINUE
902          CONTINUE
*   CONVERT DELTA H TO DELTA U  :  dU = dH - dn(RT)
             DO 9905 J1=1,NT
9905         UTOT(J1)=HTOT(J1)-(DELTAN*T1(J1)/1000.0)
             IF(ICHEMKIN.EQ.1)THEN
                         IU=41
             ELSE
                         IU=0
             ENDIF
************************  OUTPUT TO TERMINAL : IU = 0 ********************
************************        OR FILE     : IU =41 ********************
5762         CONTINUE
             IF(IU.EQ.0)THEN
                         CALL CLS
             ELSE
                         CALL LINES(IU,2)
             ENDIF
        WRITE(IU,*)'     THERMODYNAMIC ANALYSIS for REACTION'
             IF(NLHS.EQ.1.AND.NRHS.EQ.1)WRITE(IU,620)LHS(1),RHS(1)
             IF(NLHS.EQ.1.AND.NRHS.EQ.2)WRITE(IU,621)LHS(1),
     +(RHS(I2),I2=1,NRHS)
             IF(NLHS.EQ.1.AND.NRHS.EQ.3)WRITE(IU,622)LHS(1),
     +(RHS(I2),I2=1,NRHS)
             IF(NLHS.EQ.2.AND.NRHS.EQ.1)WRITE(IU,630)(LHS(I1),I1=1,NLHS),
     +RHS(1)
             IF(NLHS.EQ.2.AND.NRHS.EQ.2)WRITE(IU,631)(LHS(I1),I1=1,NLHS),
     +(RHS(I2),I2=1,NRHS)
             IF(NLHS.EQ.2.AND.NRHS.EQ.3)WRITE(IU,632)(LHS(I1),I1=1,NLHS),
     +(RHS(I2),I2=1,NRHS)
           IF(NLHS.EQ.3.AND.NRHS.EQ.1)WRITE(IU,640)(LHS(I1),I1=1,NLHS),
     +(RX(I2),I2=1,NRHS)
           IF(NLHS.EQ.3.AND.NRHS.EQ.2)WRITE(IU,641)(LHS(I1),I1=1,NLHS),
     +(RHS(I2),I2=1,NRHS)
           IF(NLHS.EQ.3.AND.NRHS.EQ.3)WRITE(IU,642)(LHS(I1),I1=1,NLHS),
     +(RHS(I2),I2=1,NRHS)
           IF(UNITS.EQ.'KCAL')THEN
       WRITE(IU,654)(SH298(I1)*RC,I1=1,NLHS),(RH298(I2)*RC,I2=1,NRHS)
       WRITE(IU,656)(SS298(I1)*RC,I1=1,NLHS),(RS298(I2)*RC,I2=1,NRHS)
           ELSE IF(UNITS.EQ.'KJ')THEN
                     DO 1126 I1=1,NLHS
                               SHJ(I1)=SH298(I1)*RJ
                               SSJ(I1)=SS298(I1)*RJ
1126                 CONTINUE
                     DO 1127 I1=1,NRHS
                               SHJ(NLHS+I1)=RH298(I1)*RJ
                               SSJ(NLHS+I1)=RS298(I1)*RJ
1127                 CONTINUE
       WRITE(IU,6542)(SHJ(I1),I1=1,NLHS),(SHJ(I2),I2=NLHS+1,NLHS+NRHS)
       WRITE(IU,6562)(SSJ(I1),I1=1,NLHS),(SSJ(I2),I2=NLHS+1,NLHS+NRHS)
           ENDIF
           DO 9977 I=1,NT
           SOR(I)=STOT(I)
           GORT(I)=(HTOT(I))-(STOT(I)*T1(I)/1000.0)
           AFAR(I)=EXP(STOT(I)-(DELTAN*LOG(EXP(1.0)*82.05*T1(I))))
9977       CONTINUE
       CALL LINES(IU,1)
           IF(UNITS.EQ.'KCAL')THEN
                      WRITE(IU,500)HTOT(1)*RC
                     WRITE(IU,504)UTOT(1)*RC
                     WRITE(IU,501)SOR(1)*RC
                     WRITE(IU,502)GORT(1)*RC
           ELSE
                     WRITE(IU,5001)HTOT(1)*RJ
                     WRITE(IU,505)UTOT(1)*RJ
                     WRITE(IU,5011)SOR(1)*RJ
                     WRITE(IU,5021)GORT(1)*RJ
           ENDIF
           WRITE(IU,503)AFAR(1)
           CALL LINES(IU,1)
           IF(UNITS.EQ.'KCAL')THEN
                     WRITE(IU,67000)
           ELSE IF(UNITS.EQ.'KJ')THEN
                     WRITE(IU,67800)
           ENDIF
           DO 29977 I=2,NT
           IF(UNITS.EQ.'KCAL')THEN
                     WRITE(IU,67001)T1(I),HTOT(I)*RC,UTOT(I)*RC,SOR(I)*RC,
     +AFAR(I),GORT(I)*RC
           ELSE IF (UNITS.EQ.'KJ')THEN
                     WRITE(IU,67001)T1(I),HTOT(I)*RJ,UTOT(I)*RJ,SOR(I)*RJ,
     +AFAR(I),GORT(I)*RJ
           ENDIF
```

```
29977       CONTINUE
            CALL LINES(0,1)
            IF(IU.NE.0.AND.ICHEMKIN.EQ.0)THEN
                        IU=0
                        GOTO 5762
            ELSEIF(ICHEMKIN.EQ.1)THEN
                        GOTO 10000
            ENDIF
*********************** OPTIONS HERE ********************************
            IF(RXFIL.EQ.0)THEN
                        write(*,8208)
            else
                        WRITE(*,8209)
            endif
            write(*,8300)
       READ(*,3)NUL
       CALL READCHAR(NUL)
            IF(NUL.EQ.' ')then
                        call cls
                        GOTO 100
       ELSEIF(NUL.EQ.'Q')THEN
                        CLOSE(41)
                        OPEN(13,FILE='THUNITS.TMP',STATUS='UNKNOWN')
                        WRITE(13,14)UNITS
                        CLOSE(13,STATUS='KEEP')
                        STOP ' '
            ELSEIF(NUL.EQ.'U')THEN
                        IF(UNITS.EQ.'KJ')THEN
                                    UNITS='KCAL'
                                    IU=0
                                    GOTO 5762
                        ELSE IF(UNITS.EQ.'KCAL')THEN
                                    UNITS='KJ'
                                    IU=0
                                    GOTO 5762
                        ENDIF
            ELSEIF(NUL.EQ.'N')THEN
                        CLOSE(54)
                        GOTO 1023
            ELSEIF(NUL.EQ.'*')THEN
                        RXFIL=0
                        CLOSE(41)
                        NUL='F'
            ENDIF
       IF(NUL.EQ.'F'.AND.RXFIL.EQ.0)THEN
              CALL CLS
       WRITE(*,*)'   ENTER FILENAME FOR REACTION ANALYSIS OUTPUT'
              READ(*,3)NUL1
                        IF(NUL1.EQ.' '.OR.NUL1.EQ.'Q'.OR.NUL1.EQ.'q')THEN
                                                IU=0
                                                GOTO 5762
                        ENDIF
              INQUIRE(FILE=NUL1,EXIST=EXST,OPENED=OPNED)
                        RXFIL=1
```

```
               IF(EXST)THEN
               WRITE(*,4)NUL1
               WRITE(*,*)'ALREADY EXISTS'
               CALL LINES(0,1)
               WRITE(*,*)'   1 - OVERWRITE'
               WRITE(*,*)'   2 - APPEND'
               WRITE(*,*)'   3 - ABORT'
               CALL LINES(0,1)
               WRITE(*,*)'       enter choice { 1 }'
               READ(*,3)NUL2
               CALL READCHAR(NUL2)
               IF(NUL2.EQ.' '.OR.NUL2.EQ.'1')THEN
                      OPEN(41,FILE=NUL1,STATUS='UNKNOWN')
               ELSEIF (NUL2.EQ.'2')THEN
                           OPEN(41,FILE=NUL1,STATUS='OLD')
                                    IHJ=0
12345                               CONTINUE
                           IHJ=IHJ+1
                           READ(41,3,END=4591)NUL2
                                    GOTO 12345
4591                   RXFIL=1
                                    BACKSPACE(41)
               ELSE
                                    IU=0
                                    GOTO 5762
                        ENDIF
                        ELSE
                      OPEN(41,FILE=NUL1,STATUS='UNKNOWN')
                        ENDIF
       ENDIF
       IF(NUL.EQ.'F')THEN
                        IU=41
                        GOTO 5762
8877   CONTINUE
       ENDIF

            GOTO 5762
       STOP
****************** FORMAT BLOCK ***************************************
620    FORMAT(1X,'Rx    ',6X,A10,1X,'=',1X,A10)
621    FORMAT(1X,'Rx    ',6X,A10,1X,'=',1X,A10,1X,'+',1X,A10)
622    FORMAT(1X,'Rx    ',6X,A10,1X,'=',1X,A10,2(1X,'+',1X,A10,1X))
630         FORMAT(1X,'Rx    ',6X,A10,1X,'+',1X,A10,1X,'=',1X,A10)
631         FORMAT(1X,'Rx    ',6X,A10,1X,'+',1X,A10,1X,'=',1X,A10,1X,'+'
    +,1X,A10)
632         FORMAT(1X,'Rx    ',6X,A10,1X,'+',1X,A10,1X,'=',1X,A10,2(1X,'+'
    +,1X,A10,1X))
640         FORMAT(1X,'Rx    ',6X,A10,2(1X,'+',1X,A10,1X),'=',1X,A10)
641         FORMAT(1X,'Rx    ',6X,A10,2(1X,'+',1X,A10,1X),'=',1X,A10,
    +1X,'+',1X,A10)
642         FORMAT(1X,'Rx    ',6X,A10,2(1X,'+',1X,A10,1X),'=',1X,A10,
    +2(1X,'+',1X,A10,1X))

654    FORMAT(1X,'Hf {Kcal/mol}',1X,6(F8.3,3x))
```

```
656      FORMAT(1X,'S {cal/mol K}',1X,6(F8.3,3x))
6542     FORMAT(1X,'Hf {KJ/mol}  ',1x,6(F8.3,3x))
6562     FORMAT(1X,'S {J/mol K}  ',1X,6(F8.3,3x))
500      FORMAT(1X,'  dHr   (298K) = ',1X,F10.3,1X,' Kcal/mol')
5001     FORMAT(1X,'  dHr   (298K) = ',1x,F10.3,1X,' KJ/mol')
501           FORMAT(1X,'  dSr   ( " )  = ',1X,F10.3,1X,' cal/mol K')
5011          FORMAT(1X,'  dSr   ( " )  = ',1X,F10.3,1X,' J/mol K')
502           FORMAT(1X,'  dGr   ( " )  = ',1X,F10.3,1x,' Kcal/mol')
5021          FORMAT(1X,'  dGr   ( " )  = ',1X,F10.3,1x,' KJ/mol')
503           FORMAT(1X,' Af/Ar  ( " )  = ',1X,1P,E10.3)
504      FORMAT(1X,'dU (dE) ( " )  = ',1X,F10.3,1X,' Kcal/mol')
505      FORMAT(1X,'dU (dE) ( " )  = ',1X,F10.3,1X,' KJ/mol')
8208   format(1x,'Q (quit)',3x,'F (file)',3x,'N (new input file)',3x,
     +'U (change units)')
8209   format(1x,'Q (quit)',2x,'F (file)',2x,'N (new input file)',2x,
     +'U (change units)',2X,'* (new output file)')
8300   format(1x,'   enter option  { enter new reaction }')
67000         FORMAT(6X,'T (K)',2X,'dH(Kcal/mol)',2x,'dU(Kcal/mol)',1X,
     +'dS(cal/mol K)',3X,'(Af/Ar)',1X,'dG(Kcal/mol)')
67800         FORMAT(6X,'T (K)',3X,'dH(KJ/mol)',3x,'dU(KJ/mol)',3X,
     +'dS(J/mol K)',4X,'(Af/Ar)',5X,'dG(KJ/mol)')
67001         FORMAT(1X,F10.3,5(3X,1P,E10.3))
*************************************************************************
        END
```

```
      SUBROUTINE NASAINIT(SS,NX,IE1,NE1,ICNT,THERMFIL)
              IMPLICIT REAL*8(A-H,O-Z)
      INTEGER IG, Y, NG, P, X, O, D,NE(4),NE1(4,400),FLG,IFIRST(7),
     +ILAST(7),U,NX(400)
              REAL*8 TRA(3),F(14)
        LOGICAL EXST,OPNED
      CHARACTER*14 SS(400)
              CHARACTER*70 THERMFIL,s1
              CHARACTER*80 DUMMY
      CHARACTER*14 NUL,NUL1(2,50),funits
              CHARACTER*4 IE1(4,400),REF(2)
              CHARACTER*6 REF3
              CHARACTER PHASE
*     THIS SUBROUTINE INITIALIZES A NASA FORMAT DATABASE FILE (*.DAT)
*     AS A DIRECT ACCESS FILE FOR USE ELSEWHERE
*     RETURNS A VECTORS OF SPECIES NAMES AND LINE# IN FILE
*             SS(400) :  SPECIES NAMES (UP TO 400 SPECIES SUPPORTED)
*             NX(400) :  THE LINE NUMBER IN THE NASA FILE WHICH STARTS SS(I)
*        ICNT     :  TOTAL NUMBER OF ENTRIES IN NASA FILE
*             THERMFIL:  NASA FORMAT FILENAME

70            FORMAT(A70)
14            FORMAT(A14)
10000       WRITE(*,*)'    ENTER FILENAME FOR NASA FORMAT COEFFICIENTS
     + { *.DAT }'
        CALL LINES (0,4)
        READ(*,70)THERMFIL
897     FORMAT(A14)
        CALL LRDCHAR(THERMFIL)
        IF(THERMFIL.EQ.' '.OR.THERMFIL.EQ.'Q')RETURN
        IIH=0
956      IIH=IIH+1
          IF (THERMFIL(IIH:IIH).EQ.'.'.OR.THERMFIL(IIH:IIH).EQ.' ')THEN
              LEN=IIH-1
              GOTO 987
          ENDIF
        GOTO 956
987     CONTINUE
        THERMFIL(LEN+1:LEN+4)='.DAT'
        INQUIRE(FILE=THERMFIL,EXIST=EXST,OPENED=OPNED)
        IF(EXST)GOTO 10002
        CALL CLS
        WRITE(*,7000)THERMFIL
7000    FORMAT(' ',5X,A70)
             WRITE(*,*)'          does not exist'
        CALL LINES(0,2)
        WRITE(*,7001)
7001    FORMAT(' ',20X,'please try again')
        goto 10000
10002   CONTINUE
        OPEN(54,FILE=THERMFIL,ACCESS='DIRECT',RECL=80,FORM='FORMATTED'
     +,STATUS='OLD')
```

```
**          INITIALIZE NASA FORMAT FILE
*           FIRST SPECIES NAME APPEARS ON LINE 3 OF NASA FORMAT FILE
*       SUBSEQUENT NAMES APPEAR EVERY 4 th LINE : FILE IS TERMINATED
*       BY THE WORD "END"
*           SET INITIAL VALUES
            READ(54,'(A)',REC=1)DUMMY
            READ(54,'(A)',REC=2)DUMMY
            ICNT=0
            J=3
1           CONTINUE
            IF(ICNT.NE.0)J=J+4
            ICNT=ICNT+1
            SS(ICNT)=' '
            READ(54,1010,REC=J,END=99999)SS(ICNT),(REF(I),I=1,2),REF3,
     +(IE1(I,ICNT),NE1(I,ICNT),I=1,4),PHASE,TRA(1),TRA(3),TRA(2)
            IF(SS(ICNT)(:3).NE.'END'.AND.SS(ICNT)(:3).NE.'end')THEN
            NX(ICNT)=J
                        GOTO 1
            ELSE
                        ICNT=ICNT-1
                        RETURN
            ENDIF
1010  FORMAT(A10,2A4,A6,4(A2,I3),A1,2F10.3,1X,F8.0)
1020  FORMAT(5E15.8,/,5E15.8,/,4E15.8)
99999       ICNT=ICNT-1
            CALL CLS
            WRITE(*,*)'   WARNING !   "END" card is missing'
            write(*,*)'   the last line in a NASA format file'
            write(*,*)'   MUST have  END  starting in column 1'
            call lines (0,1)
            write(*,*)'   an end of file was encountered'
            pause ' hit return to continue'
            return
98989       call cls
            write(*,*)' I/O ERROR encountered'
            pause '  hit return to continue'
            CLOSE(54)
            return
            end
```

```
             SUBROUTINE READNASA(SPEC,IADD,FILIN,IFLG)
             IMPLICIT REAL*8(A-H,O-Z)
        DIMENSION  TRA(3),F(14),ST1(20),HT1(20),T1(20)
             CHARACTER*10 NNAME
             CHARACTER*14 SPEC
        CHARACTER*1 PHASE
        CHARACTER*2 IE(4), BLK1
        CHARACTER*3 BLK2,NE(4)
        CHARACTER*4 IFLAG,INAME,ID(10),REF(2)
        CHARACTER*6 XNAME,REF3
             CHARACTER*14 UNITS
             CHARACTER*70 FILIN
             LOGICAL EXST,OPNED
             COMMON/RTEMP/ST1,HT1,T1,NT
             COMMON/CONFG6/UNITS
        DATA  IFLAG,BLK1,BLK2/'END ',' ' ,'  '/
110          CONTINUE
             IFLG=0
102          FORMAT(A70)
3            FORMAT(A14)
             INQUIRE(FILE=FILIN,OPENED=OPNED,EXIST=EXST,ERR=101)
             IF(OPNED)GOTO 100
             IF(EXST)THEN
                          WRITE(*,*)' FILE NOT OPENED ERROR'
                          PAUSE' hit return to continue'
                          IFLG=-10
                          RETURN
             ENDIF
101          CALL CLS
             WRITE(*,*)'CANNOT FIND NASA FORMAT INPUT FILE'
             WRITE(*,*)FILIN
             WRITE(*,*)'ENTER FILESEC { Q = QUIT } '
             READ(*,102)FILIN
             CALL LRDCHAR(FILIN)
             IF(FILIN.EQ.'Q'.OR.FILIN.EQ.' ')THEN
                          IFLG=2
                          RETURN
             ENDIF
             GOTO 110
100          CONTINUE
  650   ICOUNT=1
  655   READ(54,1010,REC=IADD,END=999,ERR=909)SPEC,(REF(I),I=1,2),
     +REF3,(IE(I),NE(I),I=1,4),PHASE,TRA(1),TRA(3),TRA(2)
             IF(TRA(2).EQ.0.)TRA(2)=TRA(3)
  658    CONTINUE
  660   READ(54,1020,END=999) (F(I),I=1,14)
        CALL CHECK(F,TRA)
        ICOUNT=ICOUNT+1
  70   FORMAT(A4)
 1010  FORMAT(A10,2A4,A6,4(A2,A3),A1,2F10.3,1X,F8.0)
 1020  FORMAT(5E15.8,/,5E15.8,/,4E15.8)
  900  CONTINUE
            RETURN
999         CONTINUE
            IFLG=1
            RETURN
909         CONTINUE
            IFLG=-20
            CLOSE(54,STATUS='KEEP')
            RETURN
       END
C
       SUBROUTINE CHECK (F,TRA)
            IMPLICIT REAL*8(A-H,O-Z)
       DIMENSION TRA(3),F(14),ST1(20),HT1(20),T1(20)
       CHARACTER*1 PHASE
       CHARACTER*2 IE(4)
       CHARACTER*3 NE(4)
       CHARACTER*4 REF(2),INAME
       CHARACTER*6 REF3,XNAME
            CHARACTER*14 UNITS
            COMMON/RTEMP/ST1,HT1,T1,NT
            COMMON/CONFG6/UNITS

C
C-----FIRST COMPUTING HF298 AND S298 FROM THE INPUT PARAMETERS
C----  WANT TO COMPUTE H AND S AT 298 WITH THE LOW TEMP PARAMETERS
C
*  REMEMBER HF IS H/RT
            DO 40  J= 1,NT
            IF(T1(J).LE.TRA(2))THEN
                        H=HF(7,T1(J),F)*T1(J)
                        S=ST(7,T1(J),F)
            ELSE IF(T1(J).GT.TRA(2))THEN
                        H=HF(0,T1(J),F)*T1(J)
                        S=ST(0,T1(J),F)
            ENDIF
C     CONVERT H TO H/(1000.*R) FOR OUTPUT
       HT1(J)=H/1000.
            ST1(J)=S
  40        CONTINUE
C
  50   RETURN
       END
```

APPENDIX V - D

FORTRAN SOURCE CODE FOR

THERMPRN

```
             PROGRAM THERMPRN
C
        IMPLICIT REAL*8 (A-H,O-Z)
        DIMENSION  TRA(3),F(14),CPL(50)
        CHARACTER*1 PHASE
        CHARACTER*2 IE(4), BLK1
        CHARACTER*3 BLK2,NE(4)
        CHARACTER*4 IFLAG,INAME,ID(10),REF(2)
        CHARACTER*6 XNAME,REF3
             CHARACTER*70 FILIN,FILOUT,NUL1,SPEC,NUL2
             character*14 nul
             LOGICAL EXST1,EXST2
             COMMON/IPOLY/IPOLY,IUNIT,IHF
        DATA  IFLAG,BLK1,BLK2/'END ','  ' ,'   '/
C   *****    ***********
C       PROGRAMS JANAFLST/THERINFO USES COEFFICIENTS IN THE EQUATION
C       (CP)/R = C1+C2*T+C3*T**2+C4*T**3+C5*T**4
C       BY THE METHOD OF LEAST SQUARES. THE ADDITIVE CONSTANT C6 IN
C       H/RT = C1+C2/2*T+C3/3*T**2+C4/4*T**4+C6/T IS ALSO DETERMINED.
C       (HERE H IS THE TOTAL ENTHALPY -- DEFINED AS HF298+H(T)-H(298))
C       C7 IS DETERMINED BY S/R=C1LNT+C2*T+C3/2*T+C3/2*T**2+C4/3*T**3
C       +C5/4*T**4+C7
C
C       THE INPUT IS TAKEN  DIRECTLY FROM THE THERMO DATA USED FOR
C       CHEMKIN
C
C
C       NPTS= NUMBER OF TEMPERATURE POINTS FOR A GIVEN SET
C       INAME= NAME OF SPECIES (FIRST 4 LETTERS)
C       XNAME= NAME OF SPECIES (NEXT 6 LETTERS)
C       GMW= GRAM MOLECULAR WEIGHT OF SPECIES
C       HF298= HEAT OF FORMATION AT 298 K (KCAL/MOLE) (JANAF INPUT)
C       T (I) = TEMP IN DEGREE K
C       CP(I) = HEAT CAPACITY 9JANAF UNITS)
C       S(I)=  ENTROPY (JANAF UNITS)
C       H(I)=  H(T)-H(298) (JANAF UNITS)
C
C
C       FILE ASSIGNMENTS ARE
C           FILE 1 - INPUT DATA FITTED IN NASAFIT FORMAT FOR
C                    CHECKING 9SAME FORMAT AS USED FOR INTERPRETER)
C           FILE 3 - OUTPUT LISTING H AND S 298 K AND CP'S OVER TEMP
C                    RANGES
C
C
C*****************************************************************************
C
C
110          CONTINUE
             DO 111 I=1,30
111          WRITE(*,*)' '
             WRITE(*,*)'ENTER INPUT FILENAME {.DAT}'
             READ(*,70)FILIN
```

```
          CALL FNAME(FILIN,'DAT')
          INQUIRE(FILE=FILIN,EXIST=EXST1)
          if(exst1)then
                    OPEN(1,FILE=FILIN,STATUS='OLD')
                    goto 120
          endif
          write(*,*)' cannot find input file:'
          write(*,'(1X,A70)')filin
          write(*,*)' hit return to continue or "Q" to abort'
          read(*,3)nul
          if(nul.eq.'q'.or.nul.eq.'Q')stop 2
          goto 110

120       CONTINUE
          CALL CLS
          WRITE(*,*)' enter species name { Q to end }'
          READ(*,'(A70)')SPEC
          CALL DBLNK(SPEC,LENGTH,0)
          if(spec.eq.' ')goto 120
          IF(SPEC.EQ.'Q'.OR.SPEC.EQ.'q')GOTO 900

          CALL LINES(0,2)
          WRITE(*,*)'ENTER "1" for both polynomials'
          WRITE(*,*)'      "2" switch from low T to high T at Tmid'
          write(*,*)' enter option: {2}'
          READ(*,3)NUL
          CALL READCHAR(NUL)
          IF(NUMIN.EQ.'Q')GOTO 120
          IPOLY=2
          IF(NUL.NE.'2'.AND.NUL.NE.' ')IPOLY=1

          CALL LINES (0,3)
          WRITE(*,*)'ENTER "0" for { Kcal/mol }'
          WRITE(*,*)'      "1" for { KJ/mol }'
          write(*,*)' enter option: {0}'
          READ(*,3)NUL
          CALL READCHAR(NUL)
          IF(NUL.EQ.'Q')GOTO 120
          IUNIT=0
          IF(NUL.NE.'0'.AND.NUL.NE.' ')IUNIT=1

          CALL LINES (0,3)
          WRITE(*,*)'ENTER "0" for Cp only'
          WRITE(*,*)'      "1" for Cp, Hf, & S'
          write(*,*)' enter option: {0}'
          READ(*,3)NUL
          CALL READCHAR(NUL)
          IF(NUL.EQ.'Q')GOTO 120
          IHF=0
          IF(NUL.NE.'0'.AND.NUL.NE.' ')THEN
                    WRITE(*,*)'ENTER "0" for Hf(T) & S(T)'
                    write(*,*)'      "1" for (Hf(T)-Hf(298 K))'
                    write(*,*)'             & (S(T)-S(298 K))'
                    write(*,*)' enter option: {0}'
```

```
                         READ(*,3)NUL
                         CALL READCHAR(NUL)
                         IF(NUL.EQ.'Q')GOTO 120
                         IHF=1
                         IF(NUL.NE.'0'.AND.NUL.NE.' ')THEN
                                      IHF=2
                         ENDIF
            ENDIF
*     end options
3           FORMAT(A14)
70          FORMAT(A70)
130         CONTINUE
                         REWIND(1)
131                      read(1,'(A70)',end=140)nul1
                         NUL1(10:70)=' '
                         CALL DEBLNK(NUL1,NUL2,LNUL,0)
                         IF(NUL1(:LNUL).EQ.SPEC(:LENGTH))THEN
                                      BACKSPACE(1)
                                      GOTO 655
                         ELSE
                                      goto 131
                         ENDIF
140                      backspace(1)
                         backspace(1)
                         CALL CLS
                         WRITE(*,*)SPEC
                         WRITE(*,*)'  SPECIES NOT FOUND'
                         PAUSE 'HIT RETURN TO CONTINUE '
                         GOTO 120

   10   READ (1,71) INAME
   12   READ (1,71) TNAME
C
C
C   ** READ DATA IN FROM FILES USED BY CHEMKIN, NOT JANAFIT.**
C        OR THERCOEF
C     GET ID FOR PRINTOUT
  100   FORMAT(1X,'ENTER ID FOR THERMO FILE')
  101   FORMAT(15A4)
  650   ICOUNT=1
  102   FORMAT (1H1,15A4)
  119   FORMAT (1X,'SPECIES',4X,'HF(298)',1X,'S (298) ',1X,' CP300 ',
     +' CP400 ',' CP500  ',' CP600 ',' CP800 ',' CP1000  ','CP1500 ',
     3 8X,' DATE  REF  ',3X,' ELEMENTS ')
187             FORMAT (1X,28X,'CP2000 CP2500 CP3000 CP4000 CP5000 CP6000 ')
  655   READ(1,1010,END=900) INAME, XNAME, (REF(I),I=1,2),REF3,
     1 (IE(I),NE(I),I=1,4),PHASE,TRA(1),TRA(3),TMID
         IF (INAME.EQ.IFLAG) GOTO 900
         DO 658 J = 1,4
         IF (NE(J).EQ.0)  IE(J) = BLK1
  658    CONTINUE
C        TRA(1)=LOW TWMP   TRA(2)=MID-TEMP  TRA(3)=HIGH TEMP
  660   READ(1,1020,END=900) (F(I),I=1,14)
C       F(8)-F(14) ARE LOW TEMP COEFFICIENTS
```

```
C       F(1)-F(7)  ARE HIGH TEMP COEFFICIENTS
        CALL CHECK(INAME,XNAME,IE,NE,REF,REF3,PHASE,F,TMID)
        ICOUNT=ICOUNT+1
C       IF(ICOUNT.GE.55) GO TO 650
        GO TO 120
   71   FORMAT(A4)
1010  FORMAT(A4,A6,2A4,A6,4(A2,A3),A1,2F10.3,1X,F8.0)
1020   FORMAT(5E15.8,/,5E15.8,/,4E15.8)
  900   CONTINUE
             CLOSE(1,STATUS='KEEP')
             STOP
        END
C
C       SUBROUTINE CHECK
        SUBROUTINE CHECK (INAME,XNAME,IE,NE,REF,REF3,PHASE,F,TMID)
             IMPLICIT REAL *8 (A-H,O-Z)
        DIMENSION F(14)
        CHARACTER*1 PHASE
        CHARACTER*2 IE(4)
        CHARACTER*3 NE(4)
        CHARACTER*4 REF(2),INAME
        CHARACTER*6 REF3,XNAME
             CHARACTER*70 NUL
             COMMON/IPOLY/IPOLY,IUNIT,IHF
C
             IF(IUNIT.EQ.1)THEN
                          CONST=4.184
             ELSE
                          CONST=1.0
             ENDIF
             WRITE(*,'(1X,A4,A6)')INAME,XNAME
             WRITE(*,*)'ENTER FILENAME FOR TABLE OUTPUT ("PRN")'
             WRITE(*,*)'   include file extension '
             READ(*,'(A70)')NUL
             OPEN (43,FILE=NUL,STATUS='UNKNOWN')
             WRITE(43,'(1X,A4,A6)')INAME,XNAME
             IF(IHF.GT.1)THEN
                          T=298.15
                          HZERO=HF(7,T,F)*1.987*CONST*T/1000.
                          SZERO=ST(7,T,F)*1.987*CONST
             ELSE
                          HZERO=0.0
                          SZERO=0.0
             ENDIF
*     1 DATA POINT AT 298.15
*   20 DATA POINTS 300 - 2000 K
*     9 MORE DATA POINTS 2000 - 6000 K EVERY 500 K
             IF(IPOLY.EQ.1.AND.IHF.EQ.1)THEN
      WRITE(43,*)'      T(K)      Cp(T)l    Cp(T)h     Hf(T)l     Hf(T)h
     +     S(T)l      S(T)h'
             ELSEIF(IPOLY.NE.1.AND.IHF.EQ.1)THEN
      WRITE(43,*)'        T(K)        Cp(T)        Hf(T)        S(T)'
             ELSEIF(IPOLY.NE.1.AND.IHF.NE.1)THEN
      WRITE(43,*)'        T(K)        Cp(T)'
```

```
             ELSE
      WRITE(43,*)'      T(K)        Cp(T)l       Cp(T)h'
             ENDIF
             DO 40  J= 1,29
             IF(J.EQ.1)THEN
                          T=100.
             ELSEIF(J.EQ.2)THEN
                          T=200.
             ELSEIF(J.EQ.3)THEN
                          T=298.15
             ELSEIF(J.EQ.4)THEN
                          T=300.
             ELSEIF(J.GT.4.AND.J.LE.21)THEN
                          T=T+100.
             ELSE
                          T=T+500.
             ENDIF
             CPTL=CPT(7,T,F)*1.987*CONST
             CPTH=CPT(0,T,F)*1.987*CONST
             HFOFTH=HF(0,T,F)*1.987*CONST*T/1000.
             SOFTH=ST(0,T,F)*1.987*CONST
             HFOFTL=HF(7,T,F)*1.987*CONST*T/1000.
             SOFTL=ST(7,T,F)*1.987*CONST
             IF(IPOLY.EQ.1)THEN
                          IF(IHF.EQ.0)THEN
                                       WRITE(43,'(1X,3(2X,F12.3))')T,CPTL,CPTH
                          ELSE
                                       WRITE(43,'(1X,7(1X,F10.1))')T,CPTL,CPTH,
     +           HFOFTL-HZERO,HFOFTH-HZERO,SOFTL-SZERO,SOFTH-SZERO
                          ENDIF
             ELSE
                          IF(T.LE.TMID)THEN
                                       IF(IHF.EQ.0)THEN
                                       WRITE(43,'(1X,2(2X,F12.3))')T,CPTL
                                       ELSE
                                       WRITE(43,'(1X,4(2X,F12.3))')T,CPTL,
     +                 HFOFTL-HZERO,SOFTL-SZERO
                                       ENDIF
                          ELSE
                                       IF(IHF.EQ.0)THEN
                                       WRITE(43,'(1X,2(2X,F12.3))')T,CPTH
                                       ELSE
                                       WRITE(43,'(1X,4(2X,F12.3))')T,CPTH,
     +                 HFOFTH-HZERO,SOFTH-SZERO
                                       ENDIF
                          ENDIF
             ENDIF
  40         CONTINUE
  165  FORMAT ( 1X,A4,A6,F7.2,2X,F7.2,3X,7(F7.2,1X),2A4,A6,
     1  3X,4(A2,A3,1X),A1)
167          FORMAT(30X,6(F7.2,1X))
   50  CLOSE(43)
             RETURN
       END
```

APPENDIX V - E

FORTRAN SOURCE CODE FOR

THE THERM SUBROUTINE LIBRARY

```
        SUBROUTINE DELCHK(K,ID)
        COMMON/DELETE/NDEL,IDEL(500)
        ID=0
        DO 100 I=1,NDEL
                IF(K.EQ.IDEL(I))THEN
                        ID=1
                        RETURN
                ENDIF
100     CONTINUE
        RETURN
        END
```

```
      SUBROUTINE LISTGR(AS,A,NGRPT,IFIRST,ILAST,IENTRY,IHELP,NE1,IE1)
C ******* LIST GROUPS CURRENTLY IN THE GROUP db *********************
$LARGE
************************************************************************
*                                                                      *
*                  SUBROUTINE LISTGR : Written by : Ed Ritter          *
*                                                                      *
*   This subprogram lists the group database w/ on-line references.    *
*   It can also list a species database loaded from a "LST" file.      *
*                                                                      *
*   VARIABLES:                                                         *
*   AS : vector of group id (names) which are identified when          *
*          estimating a species                                        *
*   A  : matrix containing Hf, S,&Cp(300,400,500,600,800,1000,1500 K)*
* NGRPT: number of entries in AS & A.                                  *
* IFIRST,ILAST : address of first and last entry from each group file*
* IENTRY: a flag which determines which program blocks are executed    *
* NE1 matrix of element numbers (used only wfor species data.          *
* IE1 matrix of elements for species use only                          *
*                                                                      *
*                                                                      *
*                                                                      *
*                                                                      *
************************************************************************
      CHARACTER*14 AS(1), NUL,NUMIN,A1,A2,A3,CS
        CHARACTER*70 NUMIN2,S2,KS(25),DUMMY,S1,GRDOC(400),GRDL2(100)
        CHARACTER*4 IE1(4,400)
             CHARACTER*14 UNITS,NULS
             INTEGER NFILES,I3(25),IND1(70),IND2(70),IND3(70),IGRL2(400)
        COMMON/DELETE/NDEL,IDEL(500)
             COMMON/GRFNM1/NFILES,I3
             COMMON/GRFNM2/KS
             COMMON/GROUPD/GRDOC,GRDL2
             COMMON/GROUPI/IGRL2
      REAL A(400,9),BO(9)
      INTEGER IFIRST(1),ILAST(1),IEND,KK,FLG,IENTRY,NE1(4,400),IND(70),
     +IX(2)
             COMMON/CONFG6/UNITS
```

```
C   IENTRY= 8    FOR LISTING SPECIES DB
      IF(IENTRY.EQ.8)THEN
            IEND=NGRPT+1
            I1=1
            I2=3
            CALL CLS
            GOTO 600
      ENDIF
      IF((IENTRY.GT.0).AND.(IENTRY.LE.7))THEN
            IEND=ILAST(IENTRY)
            I1=IFIRST(IENTRY)
            I2=I1+2
                IF(IENTRY.NE.4) CALL CLS
            GOTO 600
      ENDIF
999    CALL CLS
      IF(IENTRY.NE.0)RETURN
     J1=0
     IFLAG=0
     NGR=NGRPT

C
C*************************************************************
C**        LIST GROUPS OPTIONS MENU   ******
C*************************************************************

       WRITE(*,1000)
1000  FORMAT(31X,'LIST GROUPS MENU')
*       WRITE(*,*)' '
       WRITE(*,1100)
1100  FORMAT(' ',20X,39('_'))
          WRITE(*,*)' '
          DO 981 IIJ=1,NFILES
          if(i3(iij).eq.0)goto 981
          CALL SLEN(KS(I3(IIJ)),LEN1)
          CALL INDXST(KS(I3(IIJ)),'\',NOC1,IND1,LEN1,1,KFLG)
          CALL INDXST(KS(I3(IIJ)),'.',NOC2,IND2,LEN1,1,KFLG)
          IF(NOC1.EQ.0.AND.NOC2.EQ.0)THEN
                    WRITE(*,*)IIJ,KS(I3(IIJ))(:LEN1)
                    GOTO 981
          ENDIF
          IF(NOC1.NE.0)THEN
                    IBEGIN=IND1(NOC1)+1
          ELSEIF(NOC1.EQ.0)THEN
                    IBEGIN=1
          ENDIF
          IF(NOC2.EQ.0)THEN
                    IEND=LEN1
          ELSE IF(NOC2.NE.0)THEN
                    IEND=IND2(NOC2)-1
          ENDIF
          if(ibegin.ge.iend)goto 981
          IF(IIJ.GT.15)GOTO 981
          WRITE(*,1200)IIJ,ks(i3(iij))(IBEGIN:IEND)
```

```
981          CONTINUE
1200    FORMAT(' ',18X,I3,'  -  ',A14)
         WRITE(*,1700)
1700    FORMAT(' ',20X,'A  -  All Groups in DataBase')
         write(*,1800)
1800    FORMAT(' ',20X,'Q  -  QUIT { return to previous menu}')
1850           WRITE(*,*)' '
         WRITE(*,*)'                       enter option : { Q }'
         read(*,1900) NUMIN
1900     FORMAT(A14)
         CALL READCHAR(NUMIN)
         IF((NUMIN(:1).EQ.'Q').OR.(NUMIN(:1).EQ.' '))RETURN
         IF(NUMIN(:1).EQ.'A') THEN
               I1=1
               I2=2+I1
               IEND=NGRPT+1
               CALL CLS
               GOTO 600
         ENDIF

         CALL READNUM(NUMIN,RNUM,KK,FLG)
         IF(FLG.EQ.1)GOTO 999
         I1=IFIRST(KK)
         I2=2+I1
         IEND=ILAST(KK)
         CALL CLS
600     CONTINUE
             if(i1.le.0)i1=1
        I1S=I1
             IF(IENTRY.EQ.4)THEN
             WRITE(*,*)' THE FOLLOWING BOND INCREMENT GROUPS ARE DEFINED:'
             CALL LINES(0,1)
             ENDIF
        IF(IENTRY.EQ.8)THEN
        WRITE(*,691)
691     FORMAT(1X,3('SP#','  SPECIES ID     ',3X))
        WRITE(*,*)' '
        WRITE(*,606)
        WRITE(*,*)' '
        GOTO 602
        ENDIF
        WRITE(*,601)
         WRITE(*,606)
         WRITE(*,*)' '
606      FORMAT(1X,76('_'))
601     FORMAT(1X,3('GR#','     GROUP ID     ',5X))
602      ILINE=1
599     CONTINUE
        A1=AS(I1)
        A2=AS(I1+1)
        A3=AS(I1+2)
*     IF IEND.EQ.NGRPT+1  THEN LIST ALL GROUPS HAS BEEN SELECTED
             IF((IENTRY.NE.8.AND.IENTRY.NE.4).AND.IEND.NE.NGRPT+1)THEN
                         IF(I1.GT.ILAST(KK))A1=' '
```

```
                           IF(I1+1.GT.ILAST(KK))A2=' '
                           IF(I1+2.GT.ILAST(KK))A3=' '
             ELSE IF(IENTRY.EQ.4)THEN
                           IF(I1.GT.IEND)A1=' '
                           IF(I1+1.GT.IEND)A2=' '
                           IF(I1+2.GT.IEND)A3=' '
             ELSE IF(IENTRY.EQ.8)THEN
                     DO 2134 LW=1,NDEL
                     IF(IDEL(LW).EQ.I1)A1=' '
                     IF(IDEL(LW).EQ.I1+1)A2=' '
                     IF(IDEL(LW).EQ.I1+2)A3=' '
2134             CONTINUE
             ENDIF
         WRITE(*,69)I1,A1,(I1+1),A2,(I1+2),A3
69      FORMAT(1X,3(I3,2X,'-',2X,A14,4X))
        ILINE=ILINE+1
        I1=I1+3
        I2=I1+2
        IF(I2.GT.IEND)I2=IEND
        IF((ILINE.GT.19).OR.(I1.GT.IEND))THEN
             IF(IENTRY.EQ.4)RETURN
             WRITE(*,*)'HIT RETURN TO CONTINUE  { "?" FOR HELP }'
             READ(*,3)NUL
   3         FORMAT(A14)
   4    FORMAT(A70)
             CALL READCHAR(NUL)
                           nuls=nul
             IF (NUL(:1).EQ.'Q')GOTO 999
3543    IF(NUL(:1).EQ.'?')THEN
                CALL CLS
        WRITE(*,*)'        LIST DATABASE OPTIONS'
             WRITE(*,*)'        ____________________'
        CALL LINES(0,1)
        WRITE(*,*)'   Q -  QUIT return to previous menu'
        WRITE(*,*)'  RETURN - TO CONTINUE'
        write(*,*)'   R -  REPEAT listing (from beginning)'
        write(*,*)'   E -  EXTENDED listing (Hf S Cp )'
             IF(IENTRY.NE.8)THEN
             write(*,*)'  REF-  EXTENDED listing with references'
        else if(IENTRY.eq.8)then
        write(*,*)'   A -  extended listing as above with'
        write(*,*)'        ATOMIC composition included'
        WRITE(*,*)'   D -  Delete a species (by species number)'
        WRITE(*,*)'   U -  Undelete a species'
        endif
        call lines(0,1)
        if(nul(:2).ne.'??')write(*,*)'       enter option {continue}'
        if(nul(:2).eq.'??')then
        call lines(0,2)
        Write(*,*)'  you will be prompted for a range.'
        if(ientry.ne.8)then
                write(*,*)'  use group numbers separated by a '
        endif
        if(ientry.eq.8)then
```

```
              write(*,*)'  use species numbers separated by a '
      endif
      write(*,*)'  comma OR blank.'
      write(*,*)'  example:    '
      write(*,*)'   38 70   OR   38,70    (INTEGERS ONLY)'
      write(*,*)' '
      write(*,*)'  will all    generate extended listing for '
      if(ientry.ne.8)write(*,*)'   groups  38  through  70'
      if(ientry.eq.8)write(*,*)'   species  38  through  70'
      write(*,*)'      {hit return to continue}'
      write(*,*)'           or enter otion:'
      endif
      read(*,3)nul
           IF(NUL(:3).EQ.'ref')NUL='REF'
           NULS=NUL
                   CALL READCHAR(NUL)
           IF (NUL(:1).EQ.'Q')GOTO 999

      ENDIF
      if(nul.eq.'R')then
      if(ientry.ne.8)then
              I1=IFIRST(IENTRY)
              IF(IENTRY.EQ.0)THEN I1=1
              I2=I1+2
              CALL CLS
              GOTO 600
      ENDIF
      IF(IENTRY.EQ.8)THEN
              I1=1
              I2=3
              CALL CLS
              GOTO 600
      ENDIF
      endif
      IF(NUL(:1).EQ.'D'.AND.IENTRY.EQ.8)THEN
5234          WRITE(*,*)'DELETE which species { 0 to abort }'
              WRITE(*,*)'    or enter species number to delete}'
              read(*,*,err=99999)idelete
              if(idelete.EQ.0)goto 573
              ndel=ndel+1
              idel(ndel)=idelete
              write(*,*)'species # ',idelete
              write(*,*)'has been deleted'
              CALL LINES(0,2)
              WRITE(*,*)' hit return to continue {Q : quit}'
              READ(*,3)A1
              call readchar(A1)
              if(A1(:1).eq.'Q')then
              I1=I1S
              GOTO 600
              endif
              goto 5234
      endif
573   continue
```

```
        IF(NUL(:1).EQ.'U'.AND.IENTRY.EQ.8)THEN
                CALL CLS
                DO 765 IB=1,NDEL
765             WRITE(*,766)IDEL(IB),AS(IDEL(IB))
766     FORMAT(1X,I3,1X,'-',1X,A14)
        CALL LINES(0,1)
        WRITE(*,*)' undelete which species { 0 to abort }'
        write(*,*)'    or enter species number to undelete'
        read(*,*,err=99999)idelete
        if(idelete.eq.0)goto 767
        do 769 ib=1,ndel
                if(idel(ib).ne.idelete)goto 769
                do 770 ib1=ib,ndel-1
770             IDEL(ib1)=idel(ib1+1)
                ndel=ndel-1

769     continue
        WRITE(*,*)'  hit return to continue { Q : quit }'
        read(*,3)a1
        call readchar(a1)
        if(a1.eq.'Q')then
                I1=I1S
                GOTO 600
        endif
        GOTO 573
        endif
767     continue
             IF(NUL(:3).EQ.'ref')NUL(:3)='REF'
        if(nul(:1).eq.'E'.or.(nul(:1).eq.'A'.and.ientry.eq.8)
     +.or.(nul(:3).eq.'REF'.and.ientry.ne.8))then

        IF(IENTRY.NE.8)WRITE(*,*)' ENTER RANGE OF GROUPS TO VIEW'
        IF(IENTRY.EQ.8)WRITE(*,*)' ENTER RANGE OF SPECIES TO VIEW'
             READ(*,4)S2
             CALL LRDCHAR(S2)
             IF(S2.EQ.' '.OR.S2.EQ.'Q')THEN
                        NUL='R'
                        GOTO 3543
             ENDIF
             DO 1776 INXS=1,70
1776         IF(S2(INXS:INXS).EQ.',')S2(INXS:INXS)=' '
             CALL DEBLNK(S2,NUMIN2,LEN,1)
             CALL INDXST(S2,' ',NOCCUR,IND,LEN,1,FLG)
             KT=0
             DO 1777 INXS=1,LEN
                        IF((ICHAR(S2(INXS:INXS)).GT.57.OR.ICHAR(S2(INXS:INXS))
     +                  .LT.48).AND.ICHAR(S2(INXS:INXS)).NE.32)KT=1
1777         CONTINUE
             IF(KT.EQ.1)THEN
             CS=' '
                        IF(NOCCUR.EQ.0)THEN
                                     CS(:LEN)=S2(:LEN)
                                     CS(LEN+1:10)=' '
                                     IK=1
```

```
1812                            IF(CS(:10).EQ.AS(IK)(:10))GOTO 1813
                                IK=IK+1
                                IF(IK.LE.NGRPT)GOTO 1812
                                WRITE(*,*)'ENTRY NOT FOUND'
                                WRITE(*,*)CS(:10)
                                        WRITE(*,*)' { hit return to continue }'
                                        read(*,4)numin2
                                        nul='R'
                                        goto 3543
1813                            CONTINUE
                                IBEGIN=IK
                                ISTOP=IK
                        ELSE IF(NOCCUR.EQ.1)THEN
                                CS(:IND(1)-1)=S2(:IND(1)-1)
                                CS(IND(1):10)=' '
                                IK=1
1912                            IF(CS(:10).EQ.AS(IK)(:10))GOTO 1913
                                IK=IK+1
                                IF(IK.LE.NGRPT)GOTO 1912
                                WRITE(*,*)'ENTRY NOT FOUND'
                                WRITE(*,*)CS(:10)
                                        WRITE(*,*)' { hit return to continue }'
                                        read(*,4)numin2
                                        nul='R'
                                        goto 3543
1913                            CONTINUE
                                IBEGIN=IK
                                CS(:LEN-IND(1))=S2(IND(1)+1:LEN)
                                CS(LEN-IND(1)+1:10)=' '
                                IK=1
2012                            IF(CS(:10).EQ.AS(IK)(:10))GOTO 2013
                                IK=IK+1
                                IF(IK.LE.NGRPT)GOTO 2012
                                WRITE(*,*)'ENTRY NOT FOUND'
                                WRITE(*,*)CS(:10)
                                        WRITE(*,*)' { hit return to continue }'
                                        read(*,4)numin2
                                        nul='R'
                                        goto 3543
2013                            CONTINUE
                                ISTOP=IK
                        ELSE IF(NOCCUR.GT.1)THEN
                                WRITE(*,*)'ERROR IN INPUT:'
                                WRITE(*,*)S2
                                WRITE(*,*)' { hit return to continue }'
                                read(*,4)S2
                                NUL='R'
                                GOTO 3543
                        ENDIF
                ENDIF
                IF(KT.EQ.0)THEN
                        IF(NOCCUR.EQ.0)THEN
                                NUMIN=S2(:LEN)
                                NUL=NUMIN
```

```
                           ELSE IF(NOCCUR.EQ.1)THEN
                                        NUMIN=S2(:IND(1)-1)
                                        NUL=S2(IND(1)+1:LEN)
                           ELSE IF(NOCCUR.GT.1)THEN
                                        WRITE(*,*)'ERROR IN INPUT:'
                                        WRITE(*,*)S2
                                        WRITE(*,*)' { hit return to continue }'
                                        read(*,4)S2
                                        NUL='R'
                                        GOTO 3543
                           ENDIF
                           CALL READNUM(NUMIN,RNUM,IBEGIN,FLG)
                           CALL READNUM(NUL,RNUM,ISTOP,FLG)
              ENDIF
                IF(KFLG.EQ.1)THEN
                        NUL='?'
                        GOTO 3543
                ENDIF
        K22=IBEGIN
        CALL CLS
7999    INL=0
             if(units.eq.'KCAL')then
                           write(*,*)'   Hf { Kcal/mol }    S & Cp  { cal/mol K }'
             else if(units.eq.'KJ')then
                           WRITE(*,*)'UNITS:  Hf{KJ/mol}   S & Cp {J/mol K}'
             endif
             IF(IENTRY.NE.8)THEN
        WRITE(*,*) 'GROUP ID          Hf     S     Cp 300    400    500
     +    600       800      1000'
             ELSE IF(IENTRY.EQ.8)THEN
        WRITE(*,*) 'SPECIES           Hf     S     Cp 300    400    500
     +    600       800      1000'
             ENDIF
8000    INL=INL+1
        IF(IENTRY.EQ.8)THEN
                DO 83575 K5=1,NDEL
                if(IDEL(K5).EQ.K22)THEN
                        K22=K22+1
                        GOTO 8000
                endif
83575   continue
        endif
             if(units.eq.'KJ')then
                          do 7672 ik1=1,8
7672                      bo(ik1)=a(k22,ik1)*4.184
                          write(*,7100)as(k22),(bo(ik1),ik1=1,8)
7100                      format(1x,a14,1x,8(f7.2,1X))
             else if(units.eq.'KCAL')then
                     WRITE(*,7000)AS(K22),(A(K22,I22),I22=1,8)
7000             FORMAT(' ',A14,1X,8(1X,F7.2))
             endif
             if((nuls.eq.'REF'.and.grdoc(k22).ne.' ').and.ientry.ne.8)THEN
                                write(*,8743)GRDOC(K22)
8743                      FORMAT(1X,A70)
```

```
                                 INL=INL+1
                           IF(IGRL2(K22).NE.0)THEN
                                        WRITE(*,8743)GRDL2(IGRL2(K22))
                                        INL=INL+1
                           ENDIF
             ENDIF
        IF(NULS.EQ.'A'.AND.IENTRY.EQ.8)WRITE(*,7001)(IE1(I22,K22),
     +NE1(I22,K22),I22=1,4)
7001    FORMAT(1X,4(7X,A4,2X,I3))
        IF(NULS.EQ.'A'.AND.IENTRY.EQ.8)INL=INL+1
        K22=K22+1
        IF(INL.GE.19.OR.K22.GT.ISTOP)THEN
        WRITE(*,*)'  { HIT RETURN TO CONTINUE }'
                READ(*,3)NUMIN
                CALL READCHAR(NUMIN)
                IF(NUMIN(:1).EQ.'Q')THEN
                        NUL='?'
                        GOTO 3543
                ENDIF
                IF(K22.LE.ISTOP)GOTO 7999
        NUL='?'
        GOTO 3543
        ENDIF
        GOTO 8000
        endif
             IF(I1.GE.IEND)GOTO 999
             CALL CLS
             GOTO 600
        ENDIF
        GOTO 599
 200    CONTINUE
 100  CONTINUE
99      RETURN
99999   call cls
        write(*,*)'   ERROR:   INTEGER VALUE(S) EXPECTED'
        call lines(0,2)
        WRITE(*,*)'   check required input and try again'
        call lines(0,1)
        write(*,*)'    { hit return to continue }'
        read(*,3)nul
        call readchar(nul)
        if(nul(:1).eq.'Q')return
        nul='??'
        goto 3543
      END
      SUBROUTINE FILE(LS,RS,B,CP2K,REF,REF3,IE,NE,CS,IX,MM1,E,FLAG,BFLAG
     +,TEXT,BLFLAG,FORM,FORMP,SPECP,RS22,CPINF,SYMR,SYMS,SYMM,BD,TS,FIL,
     +N,KOUNT,LENF,LENFP,DELB,DOCTYPE,KDOCT,DELTA,IOP)
C$LARGE
C
C***********************************************************************
C
C ------  THIS SUBROUTINE PRODUCES BOTH *.LST & *.DOC FILES -----------
C
* UD 10/23/88
C***********************************************************************
c    +7                                                         column 72 ]
      INTEGER IX(1), E(1), N, NGRPT, Q,EI,FIL,HEADER,KDOCT
      INTEGER FLAG, BFLAG, SYMM,SYMS, SYMR, CTRL, Z, NGRP, NATOM
      INTEGER IG, Y, NG, P, X, O, D,NE(1),FLG,TFS,TFS2
             INTEGER*2 IYR,IMON,IDAY
      REAL SCOR, CPINF, CP2K,DELTA(1)
      REAL*4 B(2,9),BO(9)
             CHARACTER*20 S5
      CHARACTER*70 TEXT(1),KS,FS,FS2,NASAFI,NASAFO,FI,dum1,s3,S4
             CHARACTER*72 IDLINE
      CHARACTER*14 NUMIN,TSS,SPECP,DUMMY,DOCTYPE,COMMAND,OUTPUT,UNITS,
     +CS(2,60), LS, RS,CP, TS, DS, RS22,FOUNITS,FIUNITS,NUM2,TRANGE
      CHARACTER ES, ANSS,PHASE
      CHARACTER*70 FORM,FORMP
      CHARACTER*4 IE(1),REF(2)
      CHARACTER*6 REF3
             CHARACTER*1 CFLAG
             COMMON/NASANM/NASAFI,NASAFO
             COMMON/CTRL2/COMMAND,OUTPUT
             COMMON/CONFG6/UNITS
             COMMON/CONFG8/TRANGE
             COMMON/UNITSET/FOUNITS,FIUNITS
             COMMON/FILEFLG/NCALL
             COMMON/DATE/IMON,IDAY,IYR
             COMMON/NROTOR/NROT,NROTS
        LOGICAL EXST,OPNED,OPNED2
                          SAVE FS2,FS
        TFS=1
        TFS2=1
        HEADER=0
             IF(TRANGE(:4).EQ.'2000')THEN
                          CFLAG='*'
             ELSE
                          CFLAG=' '
             ENDIF
             IF(NCALL.EQ.0)GOTO 1494
             INQUIRE(34,OPENED=OPNED2)
             IF(OPNED2)THEN
                          INQUIRE(34,NAME=FI)
                          IF(FI.EQ.FS2)CLOSE(34,STATUS='KEEP')
             ENDIF
             INQUIRE(52,OPENED=OPNED)
             IF(OPNED)THEN
                           IF(OUTPUT.EQ.'CLOSE')REWIND(52)
                           GOTO 1944
             ENDIF
             IB=1
             OPEN(52,FILE=FS2,ACCESS='DIRECT',RECL=70,FORM='FORMATTED',
     +STATUS='OLD')
1944                                    READ(52,'(A70)',END=1945)DUMMY
                                        IB=IB+1
                                        GOTO 1944
1945                                    BACKSPACE(52)
                                        NREC=IB
1956         CONTINUE
        INQUIRE(7,OPENED=OPNED)
       IF (OPNED.AND.(FIL .EQ. 1.OR.FIL.EQ.2)) GOTO 8990
1494    CALL CLS
             NCALL=1
        WRITE(*,*) 'Enter output filename:'
        WRITE(*,*)'{ no file extension is needed }: '
        READ(*,4) KS
        CALL LRDCHAR(KS)
        IF(KS(:1).EQ.'Q'.OR.KS(:1).EQ.' ')THEN
                          FIL=-10
                          RETURN
             ENDIF
        JJK=1
        IF(FIL.EQ.2)GOTO 1995
        WRITE(*,*)' '
        WRITE(*,*)'   CREATE DOCUMENT FILE [ Y or N ]'
        WRITE(*,*)' '
        WRITE(*,*)'   enter choice: { Y }'
        READ(*,3)NUMIN
3       FORMAT(A14)
        CALL READCHAR(NUMIN)
        IF(NUMIN(:1).EQ.' ')THEN
                FIL=0
                GOTO 1995
        ENDIF
        IF(NUMIN(:1).EQ.'Q')RETURN
        IF(NUMIN(:1).EQ.'N')FIL=2
1995    IF((KS(JJK:JJK).EQ.' ').OR.(KS(JJK:JJK).EQ.'.'))THEN
                   ILNG=JJK
C                  THEN THIS IS POSITION OF ' . '
C                  WHICH IS FOLLOWED BY APPROPRIATE FILE EXTENSION ".*"
                   GOTO 1996
        ENDIF

        JJK=JJK+1
        GOTO 1995
1996    CONTINUE
        ILNG2=ILNG+4
        IF(TFS.EQ.1)THEN
                FS=KS
                          NASAFO=KS
```

```
            FS(ILNG:ILNG2)='.LST'
                    NASAFI=FS
                    NASAFO(ILNG:ILNG2)='.DAT'
      ENDIF
      IF(TFS2.EQ.1)THEN
            FS2=KS
            FS2(ILNG:ILNG2)='.DOC'
            IF(TFS.EQ.0)GOTO 1495
      ENDIF
      IF(TFS.EQ.0)GOTO 1495
      INQUIRE(FILE=FS,EXIST=EXST,OPENED=OPNED,ERR=1234)
      IF(OPNED)GOTO 1495
      IF(EXST)THEN
            CALL CLS
            WRITE(*,32)FS
32    FORMAT(' ',A14,2X,' ALREADY EXISTS')
            CALL LINES(0,3)
            WRITE(*,*)'    1 - DELETE (OVERWRITE)'
            WRITE(*,*)'    2 - APPEND (ADD DATA TO END)'
            WRITE(*,*)'    3 - ENTER DIFFERENT FILENAME'
      WRITE(*,*)'    Q - CANCEL (RETURN TO PREVIOUS MENU)'
            CALL LINES(0,2)
            WRITE(*,*)'    enter option: {Q}'
            READ(*,3)NUMIN
            CALL READCHAR(NUMIN)
            IF(NUMIN.EQ.'Q'.OR.NUMIN.EQ.' ')THEN
                  FIL=0
                  RETURN
            ENDIF
            IF(NUMIN.EQ.'1'.OR.NUMIN(:1).EQ.'D')THEN
                    CALL CLS
                    WRITE(*,*)' choose output units for LST file'
                    write(*,*)' 1- Kcal/mol (cal/mol K)   2- KJ/mol (J/mol K
   +)'
                    write(*,*)'     enter option { 1 }'
                    read(*,3)num2
                    founits='KCAL'
                    if(num2(:1).eq.'2')founits='KJ'
                  OPEN(7, FILE=FS,STATUS='UNKNOWN')
                    if(founits.eq.'KJ')then
                    write(7,*)' UNITS:KJ'
                    else if(fouints.eq.'KCAL')then
                    write(7,*)' UNITS:KCAL'
                    endif
                  TFS=0
            ENDIF
            IF(NUMIN.EQ.'2'.OR.NUMIN(:1).EQ.'A')THEN
                  OPEN(7,FILE=FS,STATUS='OLD')
                              read(7,70,end=1695)dum1
70                            format(a70)
                              call lrdchar(dum1)
                              call deblnk(dum1,s3,len1,0)
                              founits='KCAL'
                              if(dum1(:8).eq.'UNITS:KJ')founits='KJ'
```

```
1493                    READ(7,3,END=1695)DUMMY
                        TFS=0
                        HEADER=1
                        GOTO 1493
1695                    CONTINUE
                       BACKSPACE (7)
C     NO BACKSPACE ON VAX
                                        IF(FIL.EQ.2)GOTO 1496
                        GOTO 1495
                ENDIF
                IF(NUMIN.EQ.'3'.OR.NUMIN(:1).EQ.'N'.OR.NUMIN(:1)
     +  .EQ.'F')THEN
                TFS=1
                GOTO 1494
                ENDIF
                          IF(FIL.EQ.2)GOTO 1496
                GOTO 1495
        ENDIF
                           WRITE(*,*)' choose output units for LST file'
                           write(*,*)' 1- Kcal/mol (cal/mol K)   2- KJ/mol (J/mol K
     +)'
                           write(*,*)'     enter option { 1 }'
                           read(*,3)numin
                           founits='KCAL'
                           if(numin.eq.'2')founits='KJ'
                     OPEN(7,FILE=FS,STATUS='NEW')
                           if(founits.eq.'KJ')then
                           write(7,*)' UNITS:KJ'
                           else if(fouints.eq.'KCAL')then
                           write(7,*)' UNITS:KCAL'
                           endif

1495    CONTINUE
        INQUIRE(FILE=FS2,EXIST=EXST,OPENED=OPNED,ERR=1234)
        IF(OPNED)THEN
                           INQUIRE(34,OPENED=OPNED,ERR=1234)
                           IF(OPNED)THEN
                                        CLOSE(34,STATUS='KEEP')
                                        IOP=0
                           ENDIF
             ENDIF
        IF(EXST)THEN
                CALL CLS
                WRITE(*,32)FS2
                CALL LINES(0,3)
                WRITE(*,*)'    1 - DELETE (OVERWRITE)'
                WRITE(*,*)'    2 - APPEND (ADD DATA TO END)'
                WRITE(*,*)'    3 - ENTER DIFFERENT FILENAME'
        WRITE(*,*)'    Q - ABORT (RETURN TO PREVIOUS MENU)'
                CALL LINES(0,2)
                WRITE(*,*)'    enter option: {Q}'
                READ(*,3)NUMIN
                CALL READCHAR(NUMIN)
                IF(NUMIN.EQ.'Q'.OR.NUMIN.EQ.' ')THEN
```

```
                        FIL=0
                        GOTO 1492
                ENDIF
        IF(NUMIN.EQ.'1'.OR.NUMIN(:1).EQ.'D')THEN
         OPEN(52, FILE=FS2,ACCESS='DIRECT',RECL=70,FORM='FORMATTED',
     +STATUS='UNKNOWN')
                        TFS2=0
                ENDIF
                IF(NUMIN.EQ.'2'.OR.NUMIN(:1).EQ.'A')THEN
         OPEN(52,FILE=FS2,ACCESS='DIRECT',RECL=70,FORM='FORMATTED',
     +STATUS='OLD')
                          IB=1
1483            READ(52,'(A70)',REC=IB,END=1696)DUMMY
                        TFS2=0
                                       IB=IB+1
                        GOTO 1483
1696                    NREC=IB
                        BACKSPACE(52)
                        GOTO 1496
                ENDIF
                IF(NUMIN.EQ.'3'.OR.NUMIN(:1).EQ.'N'.OR.NUMIN(:1)
     +  .EQ.'F')THEN
                TFS2=1
                GOTO 1494
                ENDIF
        ENDIF
         OPEN(52,FILE=FS2,ACCESS='DIRECT',RECL=70,FORM='FORMATTED',
     +STATUS='NEW')
                          IB=1
                GOTO 1496
1496    CONTINUE
        IF (FIL .EQ. 0.OR.FIL.EQ.2) THEN
        IF(FIL.EQ.0)FIL=1
        IF(HEADER.EQ.1)GOTO 8990
        CALL LINES(0,2)
        WRITE(*,*)'    enter title  for *.LST file output ( blank or Q=
     +no title )'
        READ(*,4)IDLINE
4       FORMAT(A70)
        WRITE(7,5)IDLINE
5            FORMAT(1X,A70)
        WRITE(7,3890)
3890  FORMAT(1X,'SPECIES       Hf       S    Cp 300      400      500
     +600     800      1000      1500      DATE         ELEMENTS')
        ENDIF
8990    CONTINUE
        IF(FIL.EQ.2)GOTO 1067
250   FORMAT(1X,'SPECIES')
        WRITE(52,250)
      IF(BLFLAG.EQ.1)WRITE(52,38)RS22
38      FORMAT(1X,A14)
             RS22(10:14)='     '
             RS(10:14)='     '
             LS(10:14)='     '
      IF(BFLAG.GE.2)THEN
                                WRITE(52,38)RS
      ELSE IF (FLAG .EQ. 1 .OR. BFLAG .EQ. 1 ) THEN
        WRITE(52,38)RS
      ELSE IF (FLAG .EQ. 0 .AND. BFLAG .EQ. 0 ) THEN
       WRITE(52,38)LS
             ENDIF

             IF(DOCTYPE.EQ.'NOCALC')THEN
                          S4='Thermo Data Entered: '
      ELSE IF(BFLAG.GE.2)THEN
                          S4='Thermo estimation for diradical'
      ELSE IF (FLAG .EQ. 1 .OR. BFLAG .EQ. 1 ) THEN
                S4= 'Thermo estimation for radical'
      ELSE IF (FLAG .EQ. 0 .AND. BFLAG .EQ. 0 ) THEN
                S4='Thermo estimation for molecule'
      ENDIF
             WRITE(52,'(A70)')S4
        X=1
        M=1
        IF (BLFLAG .EQ. 1) THEN
          WRITE(52,22) RS22,FORM(:LENF)
22        FORMAT(1X,A14,3X,A50)
23        FORMAT(1X,'RADICAL BASED UPON PARENT  ',A14,3X,A30)
          GOTO 9003
        ENDIF
        IF (BFLAG.GE.1) THEN
               WRITE (52,22)RS,FORM(:LENF)
               WRITE(52,23)SPECP
                WRITE(52,9089)FORMP(:LENFP)
9089    FORMAT(1X,'PARENT FORMULA',1X,A30)
                          S5='PARENT SYMMETRY '
                WRITE(52,'(A20,1X,I6)')S5,SYMS
        ENDIF
        IF (BFLAG.EQ.0) WRITE(52,22)  LS,FORM(:LENF)
*  record units
             if(units.eq.'KCAL')S5=' UNITS:KCAL'
             if(units.eq.'KJ')S5=' UNITS:KJ'
             WRITE(52,'(A20)')S5
        WRITE(52,1000)N
1000    FORMAT(' ','GROUPS',1X,I3)
9003      CALL WRITEGR(1,N,CS,E,IG,52,LS)
        IF(DOCTYPE.EQ.'NOCALC')WRITE(52,9654)DOCTYPE
9654    FORMAT(1X,A14)
             if(units.eq.'KCAL')then
        WRITE(52,7257)
7257    FORMAT( '  Hf    S   Cp 300    400    500    600    800   1000
     + 1500')
        WRITE(52,10)(B(1,I9),I9=1,9)
10      FORMAT(9F7.2)
        WRITE(52,6) CPINF
6       FORMAT(' ',10X,'CPINF = ',F7.2)
             else if(units.eq.'KJ')then
        WRITE(52,7258)
7258    FORMAT( '   Hf     S    Cp 300     400     500     600     800
     + 1000    1500')
             do 1792 ik1=1,9
1792         bo(ik1)=b(x,ik1)*4.184
             cpio=cpinf*4.184
             write(52,11)(bo(ik1),ik1=1,9)
11      format(1x,2(F8.2),3(F7.2),4(f8.2))
             write(52,6)cpio
             endif
             IF(NROT.NE.0)WRITE(52,8430)NROT
8430         FORMAT(' ',1X,'NROTORS:',I2)
        ICTRL=0
        DO 2200 I=1,9
2200    IF(DELTA(I).NE.0.0)ICTRL=1
        IF(ICTRL.EQ.1)THEN
               WRITE(52,2224)
2224             FORMAT(1X,'DELTA EDIT')
                          if(units.eq.'KJ')then
                                      do 1793 ik1=1,9
1793                                  bo(ik1)=delta(ik1)*4.184
                                      write(52,2225)(bo(i),i=1,9)
                          else if(units.eq.'KCAL')then
                                 WRITE(52,2225)(DELTA(I),I=1,9)
                          endif
2225             FORMAT(1X,9(F7.2))
        ENDIF
        IF (BFLAG .EQ. 1.OR.BFLAG.EQ.2) THEN
          WRITE(52,1100)SYMR
1100      FORMAT(' ','SYMMETRY ',1X,I6)
          WRITE(52,7259)
7259    FORMAT( 'R ln(2) has been added to S to account ')
          WRITE(52,7260)
7260    FORMAT( '       for unpaired electron')
             if(units.eq.'KCAL')then
                    WRITE(52,1110)BD
             else if(units.eq.'KJ')then
                          bdu=bd*4.184
                          write(52,1110)bdu
             endif
1110      FORMAT(' ','BOND',1X,F7.2)
             if(units.eq.'KCAL')then
                    IF(DELB.NE.0.0)WRITE(52,1122)DELB
             else if(units.eq.'KJ')then
                          delbu=delb*4.184
                          if(delb.ne.0.0)write(52,1122)delbu
             endif
1122    FORMAT(' ','DELTA BOND ENERGY = ',F10.3)
        ENDIF
        IF (BFLAG .EQ. 1 ) GOTO 900
        WRITE(52,1100) SYMM
900   CONTINUE

             IF(DOCTYPE.NE.'NOCALC')THEN
```

```
                         WRITE(52,1132)IMON,IDAY,IYR
1132          FORMAT(1X,' CREATION DATE: ',I2,'/',I2,'/',I2)
            ENDIF

      IF(KOUNT.NE.0)THEN
              WRITE(52,1120)KOUNT
1120            FORMAT(' ','TEXT',I3)
                    DO 1043 IH=1,KOUNT
1043            WRITE(52,4)TEXT(IH)

      ENDIF

             WRITE(52,256)
            S4=' '
            WRITE(52,'(A70)')S4
            WRITE(52,'(A70)')S4
256     FORMAT(' ',1X,'ENDSPECIES')
1492  CONTINUE
 1067     X=1
      PHASE(:1)='G'

 1066 FORMAT(A1,A10,F7.2,2X,F7.2,3X,7(F6.2,2X),2A4,A6,3X,4(A2,I3,
    + 1X),A1,1X,I1)
 1068 FORMAT(A1,A10,F8.2,1X,F8.2,2X,7(F7.2,1X),2A4,A6,3X,4(A2,I3,
    + 1X),A1,1X,I1)
91066 FORMAT(A1,A10,F7.2,2X,F7.2,3X,7(F6.2,2X),I2,2('/',I2),A6,3X,4(A2
    + ,I3,1X),A1,1X,I1)
91068 FORMAT(A1,A10,F8.2,1X,F8.2,2X,7(F7.2,1X),I2,2('/',I2),A6,3X,4(A2
    + ,I3,1X),A1,1X,I1)
           if(founits.eq.'KJ')then
                         do 1794 ik1=1,9
1794                     bo(ik1)=b(x,ik1)*4.184
           endif
      IF ((BFLAG .EQ. 1.OR.BFLAG.EQ.2).and.founits.eq.'KCAL') THEN
           if(rs.eq.' ')GOTO 1234
           IF(DOCTYPE.EQ.'NOCALC')THEN
    WRITE(7,1066)CFLAG,RS(:10),(B(1,I9),I9=1,9),(REF(I),I=1,2),REF3,
   +   (IE(I)(:2),NE(I),I=1,4),PHASE,NROT
           ELSE
    WRITE(7,91066)CFLAG,RS(:10),(B(1,I9),I9=1,9),IMON,IDAY,IYR,REF3,
   +   (IE(I)(:2),NE(I),I=1,4),PHASE,NROT
           ENDIF
    elseIF ((BFLAG .EQ. 1.OR.BFLAG.EQ.2).and.founits.eq.'KJ') THEN
           if(rs.eq.' ')GOTO 1234
           IF(DOCTYPE.EQ.'NOCALC')THEN
    write(7,1068)CFLAG,rs(:10),(bo(ik1),ik1=1,9),(ref(i),i=1,2),ref3,
   +(ie(i)(:2),ne(i),i=1,4),phase,NROT
           ELSE
    write(7,91068)CFLAG,rs(:10),(bo(ik1),ik1=1,9),IMON,IDAY,IYR,ref3,
   +(ie(i)(:2),ne(i),i=1,4),phase,NROT
           ENDIF
      ENDIF
      IF (BFLAG .EQ. 0.and.founits.eq.'KCAL') THEN
           if(ls.eq.' ')GOTO 1234
```

```
              IF(DOCTYPE.EQ.'NOCALC')THEN
       WRITE(7,1066)CFLAG,LS(:10),(B(1,I9),I9=1,9),(REF(I),I=1,2),REF3,
     +(IE(I)(:2),NE(I),I=1,4),PHASE,NROT
              ELSE
       WRITE(7,91066)CFLAG,LS(:10),(B(1,I9),I9=1,9),IMON,IDAY,IYR,REF3,
     +(IE(I)(:2),NE(I),I=1,4),PHASE,NROT
              ENDIF
        ELSE IF (BFLAG .EQ. 0.and.founits.eq.'KJ') THEN
              if(ls.eq.' ')GOTO 1234
              IF(DOCTYPE.EQ.'NOCALC')THEN
       write(7,1068)CFLAG,ls(:10),(bo(ik1),ik1=1,9),(ref(i),i=1,2),ref3,
     +(ie(i)(:2),ne(i),i=1,4),phase,NROT
              ELSE
       write(7,91068)CFLAG,ls(:10),(bo(ik1),ik1=1,9),IMON,IDAY,IYR,ref3,
     +(ie(i)(:2),ne(i),i=1,4),phase,NROT
              ENDIF
        ENDIF
1234  CONTINUE
              IF(OUTPUT.EQ.'CLOSE')CLOSE(52,STATUS='KEEP')
      RETURN
      END

      SUBROUTINE READSP(S,SS,ICNT,IE1,NE1,REF1,REF2,PHSE,IENTRY,IHELP)
      INTEGER IG, Y, NG, P, X, O, D,NE(4),NE1(4,400),FLG,IFIRST(7),
     +ILAST(7),U
      REAL S(400,9)
        LOGICAL EXST,OPNED
      CHARACTER*14 SS(400)
              CHARACTER*70 THERMFIL,dummy,s1
      CHARACTER*14 KS, LS, RS,CP, TS, FS,DS
      CHARACTER PHSE(400),TITLE*30,HEADIN*126
      CHARACTER*14 NUL,NUL1(2,50),funits
      CHARACTER*4 IE1(4,400),REF1(2,400)
      CHARACTER*6 REF3,REF2(400)
70            FORMAT(A70)
14            FORMAT(A14)
              funits='KCAL'
                CALL CLS
        U=65
10000           WRITE(*,*)'       ENTER THERMO SPECIES DATABASE FILENAME
     + { *.LST }'
        CALL LINES (0,4)
        READ(*,70)THERMFIL
897     FORMAT(A14)
        CALL LRDCHAR(THERMFIL)
        IF(THERMFIL.EQ.' '.OR.THERMFIL.EQ.'Q')RETURN
        IIH=0
956      IIH=IIH+1
          IF (THERMFIL(IIH:IIH).EQ.'.'.OR.THERMFIL(IIH:IIH).EQ.' ')THEN
                LEN=IIH-1
                GOTO 987
          ENDIF
        GOTO 956
987     CONTINUE
```

```
        THERMFIL(LEN+1:LEN+4)='.LST'
        INQUIRE(FILE=THERMFIL,EXIST=EXST,OPENED=OPNED,ERR=10001)
        IF(EXST)GOTO 10002
        CALL CLS
        WRITE(*,7000)THERMFIL
7000    FORMAT(' ',5X,A70)
             WRITE(*,*)'          does not exist'
        CALL LINES(0,2)
        WRITE(*,7001)
7001    FORMAT(' ',20X,'please try again')
        goto 10000
10002   IF(OPNED)THEN
                REWIND(7)
                U=7
                GOTO 10003
        ENDIF
        OPEN(U,FILE=THERMFIL,STATUS='OLD')
10003   READ(U,70)dummy
             call lrdchar(dummy)
             call deblnk(dummy,s1,len1,0)
             if(dummy(:6).eq.'UNITS:')then
                           if(dummy(7:8).eq.'KJ')funits='KJ'
                           read(U,70)dummy
             endif
        READ(U,70)dummy
        ICNT=1
988     CONTINUE
             if(funits.eq.'KCAL')then
        READ(U,345,END=346)SS(ICNT)(:10),S(ICNT,1),S(ICNT,2),(S(ICNT,II
     +),II=3,9),(REF1(I,ICNT),I=1,2),REF2(ICNT),(IE1(I,ICNT)(:2),NE1(I,
     +ICNT),I=1,4),PHSE(ICNT)
345     FORMAT(1X,A10,F7.2,2X,F7.2,3X,7(F6.2,2X),2A4,A6,3X,4(A2,I3,
     +  1X),A1)
             else if(funits.eq.'KJ')then
        READ(U,349,END=346)SS(ICNT)(:10),S(ICNT,1),S(ICNT,2),(S(ICNT,II
     +),II=3,9),(REF1(I,ICNT),I=1,2),REF2(ICNT),(IE1(I,ICNT)(:2),NE1(I,
     +ICNT),I=1,4),PHSE(ICNT)
 349   FORMAT(1X,A10,F8.2,1X,F8.2,2X,7(F7.2,1X),2A4,A6,3X,4(A2,I3,
     +  1X),A1)
             do 347 ik1=1,8
347          s(icnt,ik1)=s(icnt,ik1)/4.184
             endif

        IF(NE1(1,ICNT).NE.0.OR.NE1(2,ICNT).NE.0.OR.NE1(3,ICNT).NE.0
     +.OR.NE1(4,ICNT).NE.0)ICNT=ICNT+1
        GOTO 988
346     CONTINUE
        ICNT=ICNT-1
        IF(OPNED)THEN
               BACKSPACE(7)
                GOTO 10004
        ENDIF
        CLOSE(65,STATUS='KEEP')
10004   IF(IHELP.EQ.0)GOTO 10001
```

```
      CALL LISTGR(SS,S,ICNT,IFIRST,ILAST,8,IHELP,NE1,IE1)
10001 RETURN
      END
      SUBROUTINE CLS
C     CLEAR TERMINAL SCREEN BY WRITING 23 BLANK LINES
      DO 1000 I=1,23
      WRITE (*,*)' '
 1000 CONTINUE
      RETURN
      END
      SUBROUTINE READGR(NGR,CS,TOT,IACT,M1)
C$LARGE
C************************************************************
C*        PROMT THE USER FOR INPUT, READ GROUP              *
C*    DATA, THEN DECOUPLE GROUP ID AND THE NUMBER           *
C*    OF GROUPS WITH THE SAME ID                            *
C************************************************************
      DIMENSION NUM(3)
      INTEGER TOT
      CHARACTER*14 CS(2,60)
      CHARACTER*70 CS1,DUMMY,OPTION,HLP(10)
      CHARACTER B*80,C*80,D*80
             CHARACTER*2 ERRORCK
      COMMON/CONFG3/HLP
             COMMON/CONFG7/ERRORCK
             IF(ERRORCK.EQ.' '.AND.IACT.EQ.1)THEN
      WRITE(*,12)NGR
   12 FORMAT(1X,'ENTER GROUP # ',I3)
             ENDIF
      READ(*,10)B
   10 FORMAT(A72)
             IF(B(:1).EQ.'?')THEN
                          OPTION='ENTERGROUP'
                          CALL HELP(HLP(3),OPTION)
                          IACT=4
                          RETURN
             ENDIF
C**********************
C    CONVERT TO PROPER FORMAT FOR THE SEARCH
C**********************
        CS1=' '
        CS1(:70)=B(:70)
        B=' '
        CALL DEBLNK(CS1,DUMMY,LEN,1)
        B(:14)=CS1(:14)
      IACT=0
      FLAG=0.0
      M=0
      M1=0
      M2=0
      M3=0
      M4=0
      K=1
      TOT=0
             IF(ERRORCK.EQ.'NO'.AND.B.EQ.' ')THEN
                          IACT=-10
                          RETURN
        ELSEIF(B.EQ.' ')THEN
                IACT=2
                RETURN
        ENDIF
    2 IF (FLAG.EQ.1.0)GOTO 50
      IF((B(K:K).EQ.',').OR.(B(K:K).EQ.' '))THEN
                M=K
                FLAG=1
                GO TO 60
        ENDIF
      IF((B(K:K)).EQ.'0')GOTO 60
   50 IF((B(K:K)).EQ.' ')M3=K-1
      IF((B(K:K)).EQ.' ')GOTO 3
   60   K=K+1
      IF(K.GE.81)GOTO 4
      GOTO 2
    3 IF(M3.LE.M)TOT=1
      M1=M-1
      M2=M+1
      M4=M3-M2+1
      DO 35 IN=1,M1
      IF(ICHAR(B(IN:IN)).GE.97) THEN
        L=ICHAR(B(IN:IN))-32
        B(IN:IN)=CHAR(L)
      ENDIF
   35 CONTINUE
      CS(1,NGR) = B(1:M1)
      IF(CS(1,NGR).EQ.'D')IACT=1
      IF(CS(1,NGR).EQ.'Q'.OR.CS(1,NGR).EQ.'QUIT')IACT=2
      IF(CS(1,NGR).EQ.'L')IACT=3
      IF(IACT.NE.0)RETURN
      IF(M3.LE.M)RETURN
      D=B(M2:M3)
      J=0
      NUM(1)=0
      NUM(2)=0
      NUM(3)=0
      DO 11 I=M2,M3
      J=J+1
   11 NUM(J)=ICHAR(B(I:I))-48
      IF (J.EQ.2)NUM(1)=10*NUM(1)
      IF (J.EQ.3)NUM(1)=100*NUM(1)
      IF (J.EQ.3)NUM(2)=10*NUM(2)
      TOT=NUM(1)+NUM(2)+NUM(3)
   13 FORMAT(3A1)
    4 RETURN
      END
      SUBROUTINE READCHAR(LS)
C THIS SUBROUTINE RETURNS THE SAME STRING IN CAPITOL LETTERS
C NUMERIC EXPRESSIONS REMAIN THE SAME
      CHARACTER*(*) LS
        CALL DBLNK(LS,LEN,1)
      I12=1
1400  IF(I12.GT.LEN)GOTO 1450
      IF(ICHAR(LS(I12:I12)).GE.97)THEN
        N28=ICHAR(LS(I12:I12))-32
        LS(I12:I12)=CHAR(N28)
      ENDIF
      I12=I12+1
      IF (I12.GT.LEN)GOTO 1450
      GOTO 1400
1450  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE LRDCHAR(LS1)
C THIS SUBROUTINE RETURNS THE SAME STRING IN CAPITOL LETTERS
C NUMERIC EXPRESSIONS REMAIN THE SAME
      CHARACTER*70 LS
        CHARACTER*70 LS1,DUMMY
        CALL DEBLNK(LS1,DUMMY,LEN,1)
      I12=1
1400  IF(I12.GT.LEN)GOTO 1450
      IF(ICHAR(LS1(I12:I12)).GE.97)THEN
        N28=ICHAR(LS1(I12:I12))-32
        LS1(I12:I12)=CHAR(N28)
      ENDIF
      I12=I12+1
      IF (I12.GE.(LEN+1))GOTO 1450
      GOTO 1400
1450  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE RDCH127(LS1)
      CHARACTER*127 LS
        CHARACTER*127 LS1,DUMMY
        CALL DBL127(LS1,LEN,1)
      I12=1
1400  IF(I12.GT.LEN)GOTO 1450
      IF(ICHAR(LS1(I12:I12)).GE.97)THEN
        N28=ICHAR(LS1(I12:I12))-32
        LS1(I12:I12)=CHAR(N28)
      ENDIF
      I12=I12+1
      IF (I12.GE.(LEN+1))GOTO 1450
      GOTO 1400
1450  CONTINUE
      RETURN
      END
      SUBROUTINE LINES(D,I)
      INTEGER I,D
      IF(D.EQ.0) THEN
         DO 2020 JJ=1,I
2020     WRITE (*,*)' '
         RETURN
      ENDIF
         DO 2021 JJ=1,I
2021     WRITE (D,*)' '
      RETURN
      END
      SUBROUTINE WRITEGR(ISTRT,N,CS,E,IG,IDEV,SPEC)
C$LARGE
C  THIS SUBROUTINE WRITES GROUPS FOR CURRENT SPECIES TO
C         FILE OR SCREEN
      INTEGER IDEV,N,IG,E(1),ISTRT
      CHARACTER*14 CS(2,50),SPEC
C  FOR VAX TERM : IDEV=6; FOR PC IDEV=0
       GOTO 1066
200   FORMAT(' ',2X,'Gr #',1X,'-',2X,'GROUP ID',2X,'-',1X,'Quantity')
300   FORMAT(' ',3X,I3,1X,'-',1X,A14,1X,'-',I3)
1066  IF(N.GT.5)THEN
       WRITE(IDEV,400)
       IF(FLOAT(N/2).EQ.FLOAT(N)/2.0)THEN
        NEND=N/2
       ENDIF
       IF(FLOAT(N/2).NE.FLOAT(N)/2.0)THEN
        NEND=(N/2)+1
       ENDIF
      DO 1100 I=ISTRT,NEND
       IF(I+NEND.GT.N)THEN
      WRITE(IDEV,300)I,CS(1,I),E(I)
       GOTO 1100
       ENDIF
      WRITE(IDEV,500)I,CS(1,I),E(I),I+NEND,CS(1,I+NEND),E(I+NEND)
1100  CONTINUE
          IF(CS(2,I)(:5).EQ.'ERROR')THEN
                 WRITE(IDEV,7000)
                 WRITE(IDEV,7001)CS(1,I)
          ENDIF
          IF(CS(2,I+NEND)(:5).EQ.'ERROR')THEN
                 WRITE(IDEV,7000)
                 WRITE(IDEV,7001)CS(1,I+NEND)
          ENDIF
       GOTO 1776
       ENDIF
      IF(N.GT.0)     WRITE(IDEV,200)
      DO 1101 I=ISTRT,N
      WRITE(IDEV,300)I,CS(1,I),E(I)
          IF(CS(2,I)(:5).EQ.'ERROR')THEN
                 WRITE(IDEV,7000)
                 WRITE(IDEV,7001)CS(1,I)
          ENDIF
1101  CONTINUE
1776  CONTINUE
      RETURN
400   FORMAT(' ',2(2X,'Gr #',1X,'-',2X,'GROUP ID',2X,'-',1X,'Quantity'))
500   FORMAT(' ',3X,I3,1X,'-',1X,A14,1X,'-',I3,1X,'|',3X,I3,1X,'-',
     +1X,A14,1X,'-',I3)
7000  FORMAT(1X,'******************  ERROR GROUP ENTRY WAS NOT FOUND ***
     +****************')
7001  FORMAT(1X,'-----------> ',A14,' not found in current group databas
     +e !')
      END
      SUBROUTINE FINDGR(AS,NGRPT,CS,IX,IG,ISTAT,MM1)
C$LARGE
      INTEGER IX(1),ISTAT,IG,Y,SYMM,SYMR,NGRPT,MM1
      CHARACTER*14 AS(1),CS(2,50),COMMAND,OUTPUT
             LOGICAL OPNED
             COMMON/CTRL2/COMMAND,OUTPUT
        I = 1
        ISTAT=0
3030    IF (CS(1,IG).EQ.AS(I)) GOTO 3090
        I = I + 1
        IF (I .GE. NGRPT) GOTO 3052
        GOTO 3030
3052          CONTINUE
             IF(COMMAND.EQ.'AUTO_RECALC')THEN
                          INQUIRE(52,OPENED=OPNED)
                          IF(OPNED)THEN
                                       WRITE(52,7000)
7000         FORMAT(1X,'********** ERROR ********')
                                       WRITE(52,7001)CS(1,IG)
7001         FORMAT(1X,' ENTRY ',A14,' WAS NOT FOUND')
                                       WRITE(52,7002)
7002         FORMAT(1X,' ')
                          ENDIF
             ENDIF
             WRITE(*,7000)
             WRITE(*,7001)CS(1,IG)
        ISTAT=1
        RETURN
3090    IX(IG) = I
        CS(2,IG)=CS(1,IG)
C      remember save the parent for later processing
        RETURN
601   CONTINUE
      RETURN
      END
      SUBROUTINE EDIT(CS,E,B,AS,A,N,RS22,LS,RS,TEXT,BD,TS,IX,FORM,FORMP,
     +CPINF,SYMM,SYMR,BFLAG,FLAG,BLFLAG,SCOR,SCORP,NSAVE,IG,NE,IE,
     +NGRPT,IFIRST,ILAST,NATOM,KOUNT,SYMS,TEXTS,KOUNTS,SPECP,IHELP,
     +LENF,LENFP,EP,NATOMP,IP,NP,ISOURCE,DELB,DOCTYPE,KDOCT,DELTA,NED,
     +BSAVE,DOCFIL,FILNM,DOC)
C$LARGE
      INTEGER IX(50),E(50),N,NGRPT, Q,EI,FIL,Z1,I9(1),NSAVE,NUM,TOT
      INTEGER FLAG, BFLAG, SYMM,SYMS, SYMR, CTRL, Z, NGRP, NATOM,ES
      INTEGER IG, Y, NG, P, X, O, D,NE(1),FLG,IFIRST(1),ILAST(1),ESAVE,
     +EP(1),NP(1),NATOMP,KDOCT,NE1(4,400)
      REAL A(400,9), SCOR, CPINF, CP2K,RNUM,DELTA(1)
      REAL*4 B(2,9),BSAVE(1),BO(9)
        CHARACTER*4 IE1(4,400)
        CHARACTER*70 FORM,FORMP,FORMN,DUMMY
             CHARACTER*70 DOC(1),OPTION,HLP(10)
      CHARACTER*70 TEXT(1),TEXTS(1),FILNM
      CHARACTER*14 AS(1),NUMIN,TSS,SPECP,FS2,DOCTYPE,DOCFIL
      CHARACTER*14 CS(2,60), WS(1) ,KS, LS, RS,CP, TS, FS,DS,RS22
      CHARACTER ANSS,PHASE
      CHARACTER*14 NUL,NUL1(2,50),UNITS
      CHARACTER*4 IE(1),IP(1),REF(1)
      CHARACTER*6 REF3
            COMMON/CONFG3/HLP
            COMMON/CONFG6/UNITS
            COMMON/NROTOR/NROT,NROTS
   3  FORMAT(A14)
C*********************************************************************
C***********              EDIT MENU      ***************************
C*********************************************************************
10000 CALL CLS
        IF(IHELP.EQ.0.AND.(BFLAG.EQ.0.OR.FLAG.EQ.1))THEN
        WRITE(*,*)'EDIT SPECIES MENU:'
        WRITE(*,*)'1-SYMM, 2-THERMO, 3-FORM, 4-TEXT, 5-RETURN'
        WRITE(*,*)'6-ADD GRP, 7-DELETE GRP, 8-CHANGE GRP'
        GOTO 701
        ENDIF
        IF(IHELP.EQ.0.AND.(BFLAG.NE.0.))THEN
        WRITE(*,*)'EDIT SPECIES MENU:'
        WRITE(*,*)'1-SYMM, 2-THERMO, 3-FORM, 4-TEXT, 5-RETURN'
        WRITE(*,*)'6-CHANGE BD, 7-RESTORE PARENT'
        GOTO 701
        ENDIF
      WRITE(*,100)
100   FORMAT(' ',32X,'EDIT SPECIES MENU')
      WRITE(*,150)
150   FORMAT(' ',25X,35('_'))
            CALL LINES(0,1)
      WRITE(*,200)
200   FORMAT(' ',25X,'1  -  CHANGE SYMMETRY NUMBER')
      WRITE(*,450)
450   FORMAT(' ',25X,'2  -  CHANGE/MODIFY THERMO')
C      WRITE(*,451)
451   FORMAT(' ',25X,'      { changes are recorded as a group }')
      WRITE(*,500)
```

```
500   FORMAT(' ',25X,'3  -  CHANGE SPECIES ID / FORMULA')
      WRITE(*,550)
550   FORMAT(' ',25X,'4  -  ADD TEXT TO SPECIES SCREEN')
      WRITE(*,600)
600   FORMAT(' ',25X,'5  -  RETURN TO SPECIES SCREEN')
      IF(BFLAG.EQ.0.OR.FLAG.EQ.1)THEN
          WRITE(*,250)
250       FORMAT(' ',25X,'6  -  ADD A GROUP TO THE CURRENT SPECIES')
          WRITE(*,300)
300       FORMAT(' ',25X,'7  -  DELETE A GROUP')
          WRITE(*,350)
350       FORMAT(' ',25X,'8  -  CHANGE A GROUP')
              WRITE(*,351)
351           FORMAT(' ',25X,'9  -  CHANGE NROTORS')
          GOTO 2000
      ENDIF
      IF(BFLAG.NE.0)THEN
          WRITE(*,400)
400       FORMAT(' ',25X,'6  -  CHANGE BOND DISSOCIATION ENERGY')
          WRITE(*,650)
650       FORMAT(' ',25X,'7  -  RETURN TO PREVIOUS PARENT MOLECULE')
              WRITE(*,651)
651           FORMAT(' ',25X,'8  -  CHANGE NROTORS')
      ENDIF
2000  CONTINUE
            WRITE(*,660)
660         FORMAT(' ',25X,'?  -  HELP')
701     WRITE(*,700)
700   FORMAT(' ',30X,'enter option : { Q }')
C
C*****************   END OF EDIT MENU   *****************************
      READ(*,3)NUMIN
      CALL READCHAR(NUMIN)
      IF((NUMIN(:1).EQ.'5').OR.(NUMIN(:1).EQ.' ').OR.
     +(NUMIN(:1).EQ.'Q')) GOTO 999

*********************** HELP *******************************************

            IF(NUMIN(:1).EQ.'?')THEN
                        WRITE(*,4343)
4343        FORMAT('+',30X,'HELP ON WHICH OPTION ?')
            READ(*,70)OPTION
70          FORMAT(A70)
                        IF(OPTION.EQ.' '.OR.OPTION.EQ.'Q')GOTO 10000
                        NUMIN=OPTION(:14)
                        CALL READNUM(NUMIN,RNUM,NUMBR,KFLG)
                        IF(NUMBR.LE.5.OR.BFLAG.EQ.0)THEN
                                    CALL HELP(HLP(7),OPTION)
                                    GOTO 10000
                        ELSE IF(NUMBR.GE.6.AND.BFLAG.NE.0)THEN
                                    OPTION(2:2)='R'
                                    CALL HELP(HLP(7),OPTION)
                                    GOTO 10000
                        ENDIF
```

```
                           GOTO 10000
              ENDIF
************************ END OF HELP **********************************
764      CALL CLS
      IF((NUMIN(:1).EQ.'5').OR.(NUMIN(:1).EQ.' ').OR.
     +(NUMIN(:1).EQ.'Q')) GOTO 999
      CALL CLS
C***********************************************************************
C                    CHANGE SYMMETRY NUMBER
C***********************************************************************
      IF(NUMIN(:1).EQ.'1'.OR.NUMIN(:1).EQ.'S')THEN
           IF(BFLAG.EQ.0)THEN
              IF(FLAG.EQ.1)SYMR=SYMM
              WRITE(*,10)SYMM
10            FORMAT('0',15X,'SYMMETRY # = ',I3)
              WRITE(*,11)
11            FORMAT('0',15X,'enter new SYMMETRY #')
              READ(*,3)NUMIN
              CALL READCHAR(NUMIN)
              IF(NUMIN(:1).EQ.'Q'.OR.NUMIN(:1).EQ.' ')GOTO 10000
              CALL READNUM(NUMIN,RNUM,SYMM,FLG)
              IF(FLG.EQ.1)THEN
                   FLG=0
                   GOTO 10000
              ENDIF
                           IF(FLAG.EQ.1)THEN
                                        SYMR=SYMM
                           ENDIF
                   SYMS=SYMM
                   B(1,2)=B(1,2)+SCOR
              SCOR=1.98717*ALOG(SYMM*1.0)
              SCORP=SCOR
              B(1,2)=B(1,2)-SCOR
           ENDIF
           IF(BFLAG.NE.0)THEN
              WRITE(*,10)SYMR
              WRITE(*,11)
              READ(*,3)NUMIN
              CALL READCHAR(NUMIN)
              IF(NUMIN(:1).EQ.'Q'.OR.NUMIN(:1).EQ.' ')GOTO 10000
              CALL READNUM(NUMIN,RNUM,SYMR,FLG)
              IF(FLG.EQ.1)THEN
                   FLG=0
                   GOTO 10000
              ENDIF
              B(1,2)=B(1,2)+SCOR
                   SCOR=1.98717*ALOG(SYMR*1.0)
1999          B(1,2)=B(1,2)-SCOR
           ENDIF
           GOTO 999
      ENDIF
C***********************************************************************
C                    CHANGE FORMULA/SPECIES ID
C***********************************************************************
```

```
    IF((NUMIN(:1).EQ.'3').OR.(NUMIN(:1).EQ.'F'))THEN
       WRITE(*,20)
20     FORMAT(10X,'CURRENT SPECIES FORMULA IS: ')
           WRITE(*,*)FORM
       WRITE(*,21)
21     FORMAT('0',10X,'ENTER NEW FORMULA FOR SPECIES')
       WRITE(*,54)
54     FORMAT('0',10X,' 70 CHARACTERS MAX ')
77624 FORMAT(A70)
       READ(*,77624)FORMN
       CALL LRDCHAR(FORMN)
       IF((FORMN(:1).EQ.'Q').OR.(FORMN(:1).EQ.' '))GOTO 10000
         IFORM=0
     CALL DEBLNK(FORMN,DUMMY,LENF,0)
       CALL FORMINT(FORMN,LENF,IE,NE,NATOM,IFORM)
         if(iform.lt.0)then
                 numin(:1)='3'
                 goto 764
         endif
         form=formn
     DO 568 KJ=1,4
           IP(KJ)=IE(KJ)
568        NP(KJ)=NE(KJ)
     NATOMP=NATOM
       IF(BFLAG.EQ.0.OR.FLAG.EQ.1)THEN
           FORMP=FORM
           LENFP=LENF
       ENDIF
       WRITE(*,50000)
50000 FORMAT('0',10X,'ENTER NEW NAME FOR SPECIES')
       WRITE(*,50001)
50001 FORMAT('0',10X,'{ " Q " = QUIT : no change }')
       READ(*,3)NUMIN
       CALL READCHAR(NUMIN)
       IF((NUMIN(:1).EQ.'Q').OR.(NUMIN(:1).EQ.' '))GOTO 10000
       IF(FLAG.EQ.1)THEN
           RS22=NUMIN
                 RS22(10:14)='     '
           SPECP=NUMIN
                 SPECP(10:14)='     '
       ENDIF
       IF(BFLAG.NE.0)THEN
                 RS=NUMIN
                 RS(10:14)='     '
           ENDIF
       IF(BFLAG.EQ.0.AND.FLAG.EQ.0)THEN
           LS=NUMIN
                 LS(10:14)='     '
           SPECP=NUMIN
                 SPECP(10:14)='     '
       ENDIF
       GOTO 999
    ENDIF
C***********************************************************************
```

```
C                     ENTER TEXT
C**********************************************************************
7657       IF((NUMIN(:1).EQ.'4').OR.(NUMIN(:1).EQ.'T'))THEN
             IF(KOUNT.EQ.0)THEN
           WRITE(*,25)
25         FORMAT(5X,'ENTER TEXT FOR SPECIES SCREEN')
           WRITE(*,26)
26         FORMAT('0',5X,'ENTER " Q " OR BLANK WHEN FINISHED')
           KOUNT=0
                WRITE(*,*)' 70 characters max'
30         READ(*,4)TEXT(KOUNT)
4          FORMAT(A70)
           IF(((TEXT(KOUNT)(:1)).EQ.'Q').OR.((TEXT(KOUNT)).EQ.' ')
     +     .OR.((TEXT(KOUNT)(:1)).EQ.'q'))GOTO 999
           IF(BFLAG.EQ.0) TEXTS(KOUNT)=TEXT(KOUNT)
           KOUNT=KOUNT+1
           IF(KOUNT.GE.50)THEN
                 IF(BFLAG.EQ.0)KOUNTS=KOUNT
                 GOTO 999
           ENDIF
           GOTO 30
             ELSE IF (KOUNT.NE.0)THEN
             WRITE(*,*)' ENTER/MODIFY TEXT'
             DO 7655 IJK=1,KOUNT
7655         WRITE(*,7654)IJK,TEXT(IJK)
7654         FORMAT(1X,I3,': ',A70)
             CALL LINES(0,2)
             WRITE(*,*)' enter line# to modify; 0 = clear all; A = append; Q
     += quit (no change)'
             read(*,3)nul
             call readchar(nul)
             IF(NUL.EQ.' '.OR.NUL.EQ.'Q')GOTO 10000
                          IF(NUL.EQ.'0')THEN
                                       KOUNT=0
                                       NUMIN='4'
                                       GOTO 7657
                          ENDIF
                          IF(NUL.EQ.'A')THEN
           WRITE(*,25)
           WRITE(*,26)
             KOUNT=KOUNT+1
                WRITE(*,*)' 70 characters max'
32         READ(*,4)TEXT(KOUNT)
           IF(((TEXT(KOUNT)(:1)).EQ.'Q').OR.((TEXT(KOUNT)).EQ.' ')
     +     .OR.((TEXT(KOUNT)(:1)).EQ.'q'))GOTO 999
           IF(BFLAG.EQ.0) TEXTS(KOUNT)=TEXT(KOUNT)
           KOUNT=KOUNT+1
           IF(KOUNT.GE.50)THEN
                 IF(BFLAG.EQ.0)KOUNTS=KOUNT
                 GOTO 999
           ENDIF
           GOTO 32
                          ENDIF
             CALL READNUM(NUL,RNUM,LINE,FLG)
             IF(FLG.EQ.1)GOTO 10000
             IF(LINE.GT.KOUNT.OR.LINE.LE.0)GOTO 10000
             WRITE(*,*)' RE-ENTER TEXT FOR LINE# ',LINE
             READ(*,4)DUMMY
             IF(DUMMY.EQ.' '.OR.DUMMY.EQ.'Q'.OR.DUMMY.EQ.'q')GOTO 10000
             IF(DUMMY.EQ.'D'.OR.DUMMY.EQ.'d'.OR.DUMMY.EQ.'del'.OR.DUMMY.EQ.
     +'DEL'.OR.DUMMY.EQ.'DELETE'.OR.DUMMY.EQ.'delete')THEN
                          TEXT(LINE)=' '
                          GOTO 7657
             ENDIF
             TEXT(LINE)=DUMMY
             GOTO 7657
             ENDIF
      ENDIF
*---------------------
*     CHANGE NUMBER OF ROTORS
*---------------------
             IF(NUMIN.EQ.'9'.OR.(NUMIN.EQ.8.AND.BFLAG.NE.0))THEN
                          CALL CLS
                          WRITE(*,*)' number of free rotors at 5000 K'
                          write(*,'(10x,i2)')nrot
                          call lines(0,2)
                          write(*,*)' enter new number of free rotors'
                          read(*,'(a)')numin
                          call readchar(numin)
                          if(numin.eq.'Q'.OR.NUMIN.EQ.' ')goto 10000
                          if(numin.eq.'?')then
                                      OPTION='NROT'
                                      CALL HELP(HLP(8),OPTION)
                                      GOTO 10000
                          call readnum(numin,rnum,nrot,kflg)
                          if(kflg.ne.0)then
                                      kflg=0
                                      goto 10000
                          endif
                          if(bflag.ne.0)nrots=nrot
                          goto 999
             endif

*-----------------------

C*********************************************************************
C                      RECOVER PARENT/IF RADICAL
C*********************************************************************
      IF(((NUMIN(:1).EQ.'7').OR.(NUMIN(:1).EQ.'P')).AND.(BFLAG.NE.0))
     +THEN
        IF(ISOURCE.EQ.0)THEN
           N=NSAVE
           DO 896 Y=1,N
                CS(1,Y)=CS(2,Y)
                E(Y)=EP(Y)
896     CONTINUE
           DO 866 Y=1,9
866        B(1,Y)=B(2,Y)
```

```
           IF(FLAG.EQ.0)BFLAG=0
           IF(FLAG.EQ.1.AND.BFLAG.NE.0)BFLAG=0
                NROT=NROTS
           SYMM=SYMS
           SCOR=SCORP
           B(1,2)=B(1,2)-SCOR
           KOUNT=KOUNTS
           FORM=FORMP
        LENF=LENFP
           NATOM=NATOMP
                DO 548 KJ=1,4
                IE(KJ)=IP(KJ)
548             NE(KJ)=NP(KJ)
        NATOMP=NATOM
           DO 867 IZ1=1,KOUNT
867        TEXT(IZ1)=TEXTS(IZ1)
           GOTO 999
        ENDIF
                IF(ISOURCE.EQ.1)THEN
        IENTER=1
        SYMS=0
        SYMM=0
        SYMR=0
        FORM=' '
        FORMP=' '
        N=0
        NSAVE=0
        LS=' '
        RS=' '
        SPEC=' '
        NATOM=0
        NATOMP=0
        FLAG=0
        FLG=0
        BFLAG=0
        BLFLAG=0
        SCOR=0.0
        SCORP=0.0
        BD=0.0
        DELB=0.0
        DO 869 I=1,60
        CS(1,I)=' '
869     CS(2,I)=' '
        DO 873 I=1,50
        E(I)=0
873     IX(I)=0
        DO 871 I=1,4
        IE(I)=' '
871     NE(I)=0
      CALL LOADSP(AS,A,IFIRST,ILAST,NGRPT,CS,B,N,NSAVE,SYMM,SYMS,
     +SYMR,SCOR,SCORP,FORM,FORMP,SPECP,IE,NE,NATOM,FLAG,BFLAG,FLG,LS,RS,
     +RS22,FILST,DOCFIL,IX,E,LENF,EP,FILNM,DOC,NDOC,TEXT,TEXTS,KOUNT,
     +IOP,IDOC,NSPEC,BD,LENFP,IENTER,JC,DELB,DOCTYPE,KDOCT,DELTA)
                IF(FLG.EQ.1)GOTO 10000
        DO 54631 JCOUNT=1,N
        IF((IX(JCOUNT).GE.IFIRST(4)).AND.(IX(JCOUNT).LE.ILAST(4)))THEN
                BFLAG=1
                GOTO 54631
        ENDIF
        IF((IX(JCOUNT).GE.IFIRST(3)).AND.(IX(JCOUNT).LE.ILAST(3)))THEN
                FLAG=1
                GOTO 54631
        ENDIF
        IF(FLAG.EQ.1)GOTO 54631
      FLAG=0
54631  CONTINUE
       ISOURCE=1
          CPINF = ((3.0*NATOM) - (2.0+(FLOAT(NROT)/2.)))*1.987
        CP2K = B(1,9) + .37*(CPINF - B(1, 9))
                GOTO 999
        ENDIF
      ENDIF
C***********************************************************************
C                      EDIT THERMO
C***********************************************************************
      IF((NUMIN(:1).EQ.'2').OR.(NUMIN(:1).EQ.'T'))THEN
2500      CALL CLS
      WRITE(*,*)'                    EDIT THERMO SCREEN'
      WRITE(*,*)'-----------------------------------------------------'


      IF(BFLAG.GE.2)WRITE(*,*)'Thermo estimation for diradical'
             IF(DOCTYPE.EQ.'NOCALC')THEN
             WRITE(*,*) 'Thermo Data Entered: '
        ELSEIF (FLAG .EQ. 1 .OR. BFLAG .EQ. 1 ) THEN
        WRITE(*,*) 'Thermo estimation for radical'
        ELSEIF (FLAG .EQ. 0 .AND. BFLAG .EQ. 0 ) THEN
        WRITE(*,*) 'Thermo estimation for molecule'
      ENDIF
        IF (BLFLAG .EQ. 1) THEN
          WRITE(*,22) RS22,FORM
22        FORMAT(1X,A14,3X,A50)
23        FORMAT('0',1X,'RADICAL BASED UPON PARENT  ',A14,3X,A50)
          GOTO 9003
        ENDIF
        IF (BFLAG.GE.1) THEN
               WRITE (*,22)RS,FORM(:LENF)
               WRITE(*,23)SPECP,FORMP(:LENFP)
        ENDIF
        IF (BFLAG.EQ.0) WRITE(*,22)  LS,FORM(:LENF)
          CALL LINES(0,1)
        CP2K = B(1,9) + .37*(CPINF - B(1, 9))
C9003      CALL WRITEGR(1,N,CS,E,IG,0,LS)
             if(units.eq.'KCAL')write(*,*)'     Hf (Kcal/mol)    S & Cp  (cal
     +/mol K)'
             if(units.eq.'KJ')write(*,*)'     Hf (KJ/mol)   S & Cp (J/mol K)'
9003          CALL LINES(0,1)
             if(units.eq.'KJ')then
```

```
                          do 765 ik1=1,9
765                       bo(ik1)=b(1,ik1)*4.184
                          cpio=cpinf*4.184
                          WRITE(*,93)
93      format(6X,'Hf',6X,'S',4X,'Cp  300',5X,'400',6X,'500',5X,'600'
     +,5X,'800',5X,'1000',5x,'1500')
                          write(*,92)(bo(ik1),ik1=1,9)
92                        format(1x,9(f8.2,1x))
                          cpio=cpinf*4.184
                          write(*,94)cpio
             else if(units.eq.'KCAL')then
             write(*,9368)
9368   format(5X,'Hf',6X,'S',4X,'Cp 300',5X,'400',5X,'500',4X,'600'
     +,4x,'800',4X,'1000',4X,'1500')
        WRITE(*,9) (B(1,id9),id9=1,9)
9                   FORMAT(9F8.2)
                     WRITE(*,94)  CPINF
94                        format(1x, 'CPinf = ',2x,f7.2)
             endif
      CALL LINES(0,2)
      WRITE(*,*)'1{Hf} 2{S} 3{CP300} 4{CP400} 5{CP500} 6{CP600} 7{CP800}
     + 8{CP1000} 9{CP1500}'
      CALL LINES(0,1)
      WRITE(*,*)'            enter option { 1-9 : Q=QUIT }'
      READ(*,3)NUMIN
      CALL READCHAR(NUMIN)
      IF((NUMIN(:1).EQ.'Q').OR.(NUMIN(:1).EQ.' '))GOTO 10000
      CALL READNUM(NUMIN,RNUM,NUM,FLG)
      IF (FLG.EQ.1)THEN
           FLG=0
           GOTO 10000
      ENDIF
      WRITE(*,*)'          A - ADD CORRECTION  OR  R - REPLACE'
      READ(*,3)NUL
      CALL READCHAR(NUL)
      IF((NUL(:1).EQ.'Q').OR.(NUL(:1).EQ.' '))GOTO 10000
      IF(NUL(:1).EQ.'A')THEN
           WRITE(*,*)'        ENTER CORRECTION'
           READ(*,3)NUL
           CALL READCHAR(NUL)
           IF((NUL(:1).EQ.'Q').OR.(NUL(:1).EQ.' '))GOTO 10000
           CALL READNUM(NUL,RNUM,IDUMMY,FLG)
              IF(FLG.EQ.1)THEN
                  FLG=0
                  GOTO 10000
              ENDIF
             if(units.eq.'KJ')rnum=rnum/4.184
               IF(RNUM.NE.0.)THEN
       IF(NED.EQ.0)THEN
       DO 10101 I=1,7
10101          BSAVE(I)=B(1,I)
               NED=1
       ENDIF
          B(1,NUM)=B(1,NUM)+RNUM
```

```
           DELTA(NUM)=DELTA(NUM)+RNUM

                ENDIF
           GOTO 2500
      ENDIF
      IF(NUL(:1).EQ.'R')THEN
           WRITE(*,*)'       ENTER NEW VALUE'
           READ(*,3)NUL
           CALL READCHAR(NUL)
           IF((NUL(:1).EQ.'Q').OR.(NUL(:1).EQ.' '))GOTO 10000
           CALL READNUM(NUL,RNUM,IDUMMY,FLG)
              IF(FLG.EQ.1)THEN
                  FLG=0
                  GOTO 10000
              ENDIF
              if(units.eq.'KJ')rnum=rnum/4.184
           IF(abs(RNUM-B(1,NUM)).gt.0.01)THEN

        IF(NED.EQ.0)THEN
        DO 10121 I=1,7
10121           BSAVE(I)=B(1,I)
                NED=1
        ENDIF
         DELTA(NUM)=RNUM-BSAVE(NUM)
           B(1,NUM)=RNUM
           ENDIF
           GOTO 2500
      ENDIF
      ENDIF
C***********************************************************************
C             CHANGE BOND DISSOCIATION/IF RADICAL
C***********************************************************************

      IF(((NUMIN(:1).EQ.'6').OR.(NUMIN(:1).EQ.'B')).AND.(BFLAG.NE.0))
     +THEN
             if(units.eq.'KJ')then
                          bj=bd*4.184
                          write(*,45)bj
             else if(units.eq.'KCAL')then
                     WRITE(*,40)BD
40         FORMAT(' ',10X,'CURRENT BOND DISSOCIATION ENERGY IS',2X,F7.2,
     +' Kcal/mol')
45         FORMAT(' ',10X,'CURRENT BOND DISSOCIATION ENERGY IS',2X,F7.2,
     +' KJ/mol')
             endif
           WRITE(*,41)TS
41         FORMAT('0',10X,'BOND TYPE  IS',2X,A14)
           WRITE(*,42)
42         FORMAT('0',10X,'ENTER NEW BOND DISSOCIATION ENERGY')
           READ(*,3)NUMIN
           CALL READCHAR(NUMIN)
           IF((NUMIN(:1).EQ.'Q').OR.(NUMIN(:1).EQ.' '))GOTO 10000
           CALL READNUM(NUMIN,BDN,IDUMMY,FLG)
           IF(FLG.EQ.1)THEN
```

```
              FLG=0
              GOTO 10000
           ENDIF
              if(units.eq.'KJ')bdn=bdn/4.184
           WRITE(*,43)
43           FORMAT('0',10X,'CHANGE BOND NAME ( Y or N )')
           READ(*,3)NUMIN
           CALL READCHAR(NUMIN)
           IF((NUMIN(:1).EQ.'Q').OR.(NUMIN(:1).EQ.' '))GOTO 10000
           IF(NUMIN(:1).EQ.'Y')THEN
              WRITE(*,44)
44            FORMAT('0','ENTER NEW BOND NAME : " Q " TO QUIT')
              READ(*,3)NUMIN
              CALL READCHAR(NUMIN)
                 if(numin.eq.' ')goto 99
           IF((NUMIN(:1).EQ.'Q'))GOTO 10000
              TS=NUMIN
              GOTO 99
           ENDIF
           TS(12:14)='MOD'
99       BD=BD-DELB
         DELB=BDN-BD
         B(1,1)=B(1,1)+DELB
                BD=BDN
           GOTO 999
      ENDIF
C***********************************************************************
C                  ADD/DELETE/CHANGE GROUP
C                       PARENT ONLY
C***********************************************************************
        IF(BFLAG.NE.0)GOTO 999
        IF(NUMIN(:1).EQ.'A')NUMIN(:1)='6'
        IF(NUMIN(:1).EQ.'D')NUMIN(:1)='7'
        IF(NUMIN(:1).EQ.'C')NUMIN(:1)='8'
        IF((NUMIN(:1).EQ.'6').OR.(NUMIN(:1).EQ.'7').OR.(NUMIN(:1).EQ.
     +'8')) THEN
2600      CALL CLS
        DO 8991 JCOUNT=1,N
        IF((IX(JCOUNT).GE.IFIRST(3)).AND.(IX(JCOUNT).LE.ILAST(3)))THEN
                FLAG=1
                GOTO 8991
        ENDIF
      FLAG=0
8991  CONTINUE
      IF((NUMIN(:1).EQ.'6'))WRITE(*,50)
      IF((NUMIN(:1).EQ.'7'))WRITE(*,51)
      IF((NUMIN(:1).EQ.'8'))WRITE(*,52)
50    FORMAT(' ', 25X,'ADD GROUP SCREEN')
51    FORMAT(' ',25X,'DELETE GROUP SCREEN')
52    FORMAT(' ',25X,'CHANGE GROUP SCREEN')
      WRITE(*,*)'-------------------------------------------------------
     +----------------'
```

```
             IF(DOCTYPE.EQ.'NOCALC')THEN
                          WRITE(*,*)'Thermo Data Entered: '
      ELSE IF(BFLAG.GE.2)THEN
                          WRITE(*,*)'Thermo estimation for diradical'
      ELSE IF (FLAG .EQ. 1 .OR. BFLAG .EQ. 1 ) THEN
        WRITE(*,*) 'Thermo estimation for radical'
      ELSE IF (FLAG .EQ. 0 .AND. BFLAG .EQ. 0 ) THEN
        WRITE(*,*) 'Thermo estimation for molecule'
      ENDIF
        IF (BLFLAG .EQ. 1) THEN
          WRITE(*,22) RS22,FORM(:LENF)
          GOTO 2300
        ENDIF
        IF (BFLAG.GE.1) THEN
               WRITE (*,22)RS,FORM (:LENF)
               WRITE(*,23)SPECP,FORMP(:LENFP)
        ENDIF
        IF (BFLAG.EQ.0) WRITE(*,22)  LS,FORM(:LENF)
2300        CALL LINES(0,1)
        CP2K = B(1,9) + .37*(CPINF - B(1, 9))
             if(units.eq.'KCAL')write(*,*)'     Hf (Kcal/mol)    S & Cp  (cal
     +/mol K)'
             if(units.eq.'KJ')write(*,*)'     Hf (KJ/mol)   S & Cp (J/mol K)'
         CALL LINES(0,1)
             if(units.eq.'KJ')then
                          do 7652 ik1=1,9
7652                      bo(ik1)=b(1,ik1)*4.184
                          bo(8)=cp2k*4.184
                          cpio=cpinf*4.184
                          WRITE(*,93)
                          write(*,92)(bo(ik1),ik1=1,9)
                          cpio=cpinf*4.184
                          write(*,94)cpio
             else if(units.eq.'KCAL')then
             WRITE(*,9368)
        WRITE(*,9) (B(1,ID9),ID9=1,9)
                    WRITE(*,94)  CPINF
             endif
          CALL LINES(0,1)
           WRITE(*,*)'                CURRENT GROUPS'
          CALL WRITEGR(1,N,CS,E,N,6,LS)
C--------------------- ADD A GROUP BLOCK ------------------------------
          IF((NUMIN(:1).EQ.'6').OR.(NUMIN(:1).EQ.'A'))THEN
              N=N+1
              IACT=0
7766              CALL READGR(N,CS,TOT,IACT,MM1)
              IF(IACT.EQ.0)GOTO 7765
              IF(IACT.EQ.2)THEN
                  IACT=0
                  N=N-1
                  GOTO 10000
              ENDIF
              IF(IACT.EQ.3)THEN
                  IENTRY=0
           CALL LISTGR(AS,A,NGRPT,IFIRST,ILAST,IENTRY,IHELP,NE1,IE1)
                  CALL CLS
                  IACT=0
                  N=N-1
                  GOTO 2600
              ENDIF
              IF(IACT.EQ.1)THEN
                  CALL CLS
                  IACT=0
                  N=N-1
                  WRITE(*,*)'           USE OPTION 2 TO DELETE A GROUP'
                  CALL LINES(0,4)
                  WRITE(*,*)'             { hit return to continue }'
                  READ(*,3)NUL
                  NUMIN(:1)='6'
                  GOTO 2600
              ENDIF
7765          CONTINUE
                        ISTAT=0
            CALL FINDGR(AS,NGRPT,CS,IX,N,ISTAT,MM1)
              IF(ISTAT.EQ.1)THEN
                  ISTAT=0
                  GOTO 7766
             ENDIF
        CS(2,N)=CS(1,N)
      IF((IX(N).LE.ILAST(3)).AND.(IX(N).GE.IFIRST(3)))FLAG=1
      IF((IX(N).LE.ILAST(4)).AND.(IX(N).GE.IFIRST(4)))THEN
        CALL CLS
        WRITE(*,*)'         ERROR : YOU CANNOT ENTER'
        WRITE(*,*)'         BOND DISSOCIATION GROUPS'
        WRITE(*,*)'             ON THIS SCREEN'
        CALL LINES(0,3)
        WRITE(*,*)'         { hit return to continue }'
        call lines(0,3)
        N=N-1
        READ(*,3)NUMIN
        NUMIN(:1)='6'
        GOTO 2600
      ENDIF

              E(N)=TOT
                EP(N)=TOT
              DO 7767 Y=1,7
              B(1,Y)=B(1,Y)+(E(N)*A(IX(N),Y))
              B(2,Y)=B(2,Y)+(E(N)*A(IX(N),Y))
7767          CONTINUE
              NSAVE=N
                IG=IG+1
              NUMIN(:1)='6'
              GOTO 2600
           ENDIF
C-------------------- END OF ADD A GROUP BLOCK -----------------------
C
C-------------------  DELETE A GROUP BLOCK --------------------------
```

```
          IF((NUMIN(:1).EQ.'7').OR.(NUMIN(:1).EQ.'D'))THEN
             IACT=0
7667         WRITE(*,747)
747   FORMAT('0',10X,'ENTER # OF GROUP TO BE DELETED')
          READ(*,3)NUMIN
          CALL READCHAR(NUMIN)
          IF((NUMIN(:1).EQ.'Q').OR.(NUMIN(:1).EQ.' '))GOTO 10000
          IF(((ICHAR(NUMIN(:1))).GT.57).OR.((ICHAR(NUMIN(:1))).LT.48))
     +THEN
             CALL CLS
             WRITE(*,*)'    ERRROR : ENTER ONLY THE GROUP NUMBER'
             CALL LINES(0,2)
             WRITE(*,*)'       { hit return to continue }'
             READ(*,3)NUL
             GOTO 10000
          ENDIF
          CALL READNUM(NUMIN,RNUM,IADD,FLG)
          IF (FLG.EQ.1)THEN
             FLG=0
             GOTO 10000
          ENDIF
C NOW WE HAVE THE GROUP ID #
             DO 7776 Y=1,7
             B(1,Y)=B(1,Y)-(E(IADD)*A(IX(IADD),Y))
             B(2,Y)=B(2,Y)-(E(IADD)*A(IX(IADD),Y))
7776         CONTINUE

C  WE SHIFT THE GROUPS IN ARRAY TO PUT VACANCY AT END OF ARRAY
C  THEN SHORTEN THE ARRAY BY 1
               DO 11111 III=1,N
                       IF(III.GT.IADD)THEN
                               IX(III-1)=IX(III)
                               CS(1,III-1)=CS(1,III)
                               CS(2,III-1)=CS(1,III)
                               E(III-1)=E(III)
                               EP(III-1)=EP(III)
                       ENDIF
11111          CONTINUE
          N=N-1
          IG=IG-1
             NSAVE=N
               NUMIN(:1)='7'
             GOTO 2600
          ENDIF
C-------------------- END OF DELETE A GROUP BLOCK ---------------------
C
C------------------------ CHANGE A GROUP ------------------------------
          IF((NUMIN(:1).EQ.'8').OR.(NUMIN(:1).EQ.'C'))THEN
2021      CONTINUE
               IF(ISTAT.EQ.1)THEN
                       WRITE(*,*)'                      ENTRY NOT FOUND'
                       ISTAT=0
               ENDIF
          WRITE(*,2020)
```

```
2020     FORMAT('0',10X,'ENTER # OF GROUP TO CHANGE')
         READ(*,3)NUMIN
         IF((NUMIN(:1).EQ.'Q').OR.(NUMIN(:1).EQ.' '))GOTO 10000
         IF(((ICHAR(NUMIN(:1))).GT.57).OR.((ICHAR(NUMIN(:1))).LT.48))
    +THEN
           CALL CLS
           WRITE(*,*)'    ERRROR : ENTER ONLY THE GROUP NUMBER'
           CALL LINES(0,2)
           WRITE(*,*)'      { hit return to continue }'
           READ(*,3)NUL
           GOTO 2021
         ENDIF
         CALL READNUM(NUMIN,RNUM,IADD,FLG)
         IF (FLG.EQ.1)THEN
           FLG=0
           GOTO 10000
         ENDIF
C   NOW WE HAVE THE GROUP ID #
         WRITE(*,2022)
2022 FORMAT('0',10X,'{ 1 - NEW GROUP ID }')
         WRITE(*,2023)
2023 FORMAT(' ',10X,'{ 2 - DIFFERENT QUANTITY }')
         WRITE(*,2024)
2024 FORMAT(' ',10X,'{ 3 - CHANGE BOTH ID & QUANTITY }')
         WRITE(*,2025)
2025 FORMAT('0',10X,' enter option : { Q }')
         read(*,3)NUMIN
         CALL READCHAR(NUMIN)
             IF(NUMIN(:1).EQ.'2')GOTO 88889
         IF((NUMIN(:1).EQ.'Q').OR.(NUMIN(:1).EQ.' '))GOTO 10000
         IF ((NUMIN(:1).EQ.'3').OR.(NUMIN(:1).EQ.'1'))THEN
           IF(NUMIN(:1).EQ.'1')ESAVE=E(IADD)
           DO 8776 Y=1,7
           B(1,Y)=B(1,Y)-(E(IADD)*A(IX(IADD),Y))
           B(2,Y)=B(2,Y)-(E(IADD)*A(IX(IADD),Y))
8776       CONTINUE
           IACT=0
8766          CALL READGR(IADD,CS,TOT,IACT,MM1)
           IF(IACT.EQ.0)GOTO 8765
           IF(IACT.EQ.1)THEN
              IACT=0
              GOTO 10000
           ENDIF
           IF(IACT.EQ.3)THEN
              IENTRY=0
        CALL LISTGR(AS,A,NGRPT,IFIRST,ILAST,IENTRY,IHELP,NE1,IE1)
              CALL CLS
              IACT=0
              NUMIN(:1)='8'
              GOTO 2600
           ENDIF
           IF(IACT.EQ.2)THEN
              CALL CLS
              IACT=0
```

```
                WRITE(*,*)'           USE OPTION 2 TO DELETE A GROUP'
                CALL LINES(0,4)
                WRITE(*,*)'             { hit return to continue }'
                READ(*,3)NUL
                NUMIN(:1)='8'
                GOTO 2600
             ENDIF
8765         CONTINUE
               ISTAT=0
           CALL FINDGR(AS,NGRPT,CS,IX,IADD,ISTAT,MM1)
             IF(ISTAT.EQ.1)THEN
                NUMIN(:1)='8'
                GOTO 2600
            ENDIF
       CS(2,IADD)=CS(1,IADD)
     IF((IX(IADD).LE.ILAST(3)).AND.(IX(IADD).GE.IFIRST(3)))FLAG=1
     IF((IX(IADD).LE.ILAST(4)).AND.(IX(IADD).GE.IFIRST(4)))THEN
       CALL CLS
       WRITE(*,*)'          ERROR : YOU CANNOT ENTER'
       WRITE(*,*)'          BOND DISSOCIATION GROUPS'
       WRITE(*,*)'              ON THIS SCREEN'
       CALL LINES(0,3)
       WRITE(*,*)'          { hit return to continue }'
       call lines(0,3)
       READ(*,3)NUMIN
       NUMIN(:1)='8'
       GOTO 2600
     ENDIF
            IF(NUMIN(:1).EQ.'3')E(IADD)=TOT
            IF(NUMIN(:1).EQ.'3')EP(IADD)=TOT
            IF(NUMIN(:1).EQ.'1')E(IADD)=ESAVE
             DO 8767 Y=1,7
             B(1,Y)=B(1,Y)+(E(IADD)*A(IX(IADD),Y))
             B(2,Y)=B(2,Y)+(E(IADD)*A(IX(IADD),Y))
8767         CONTINUE
               NUMIN(:1)='8'
             GOTO 2600
             ENDIF
88889     CONTINUE
          IF(NUMIN(:1).EQ.'2')THEN
             WRITE(*,8790)IADD
8790      FORMAT('0',10X ,' ENTER NEW QUANTITY FOR GR#',2X,I3)
          READ(*,3)NUMIN
          IF((NUMIN(:1).EQ.'Q').OR.(NUMIN(:1).EQ.' '))GOTO 10000
          CALL READNUM(NUMIN,RNUM,ES,FLG)
          IF(FLG.EQ.1)THEN
             FLG=0
               NUMIN(:1)='8'
             GOTO 2600
          ENDIF
             DO 9767 Y=1,7
             B(1,Y)=B(1,Y)+((ES-E(IADD))*A(IX(IADD),Y))
             B(2,Y)=B(2,Y)+((ES-E(IADD))*A(IX(IADD),Y))
9767         CONTINUE
```

```
                E(IADD)=ES
                           EP(IADD)=ES
                IF (ES.EQ.0)THEN
C  WE SHIFT THE GROUPS IN ARRAY TO PUT VACANCY AT END OF ARRAY
C  THEN SHORTEN THE ARRAY BY 1
                DO 22222 III=1,N
                        IF(III.LE.1)GOTO 22222
                        IF(III.GT.IADD)THEN
                                E(III-1)=E(III)
                                EP(III-1)=EP(III)
                                CS(1,III-1)=CS(1,III)
                                CS(2,III-1)=CS(1,III)
                                IX(III-1)=IX(III)
                        ENDIF
22222           CONTINUE
           N=N-1
           IG=IG-1
           NSAVE=N
                ENDIF
                NUMIN(:1)='8'
              GOTO 2600
              ENDIF
           ENDIF
C-------------------- END OF CHANGE A GROUP BLOCK ---------------------
C
C***********************************************************************
C    END OF MAIN BLOCK FOR GROUP OPERATIONS
C***********************************************************************
      ENDIF
9999  WRITE(*,*)'ERROR IN INPUT OR OPTION IS NOT YET AVAILABLE'
      WRITE(*,*)' '
      WRITE(*,*)'  (hit return to continue)'
      READ(*,3)NUL
999   CONTINUE
      RETURN
      END
      SUBROUTINE READNUM(NUMIN,RNUM,NUM,FLG)
C  THIS SUBROUTINE RETURNS NUMERIC VALUE OF CHARACTER IF IT EXISTS
C   ELSE USER IS PROMPTED TO RE-ENTER NUMERIC VALUE
      CHARACTER*(*) NUMIN
      REAL RNUM,RENUM(132)
      INTEGER NUM,FLG,M,INTG,MDOT,FLAG,MULT,MULTFL
             INTEGER INUM(132),NUM1
      CALL DBLNK(NUMIN,LEN,0)
5      FLAG=0
        FLG=0
      M=0
             NUM1=0
      NUM=0
      RNUM=0.0
        MDOT=0
        INTG=0
        MULT=1
        MULTFL=0
      I=1
C  IF NUMIN IS A NULL STRING NUM DEFAULTS TO NUM=1
      IF(NUMIN.EQ.' ')THEN
             NUM1=1
         NUM=1
         RNUM=1.0
         GOTO 400
      ENDIF
 100  M=ICHAR(NUMIN(I:I))
      IF(M.EQ.32)THEN
         I=I-1
         GOTO 200
      ENDIF
        IF(NUMIN(I:I).EQ.'-')THEN
                MULT=INT(-1.0)
                MULTFL=I
                I=I+1
                GOTO 100
        ENDIF
      IF(NUMIN(I:I).EQ.'.')THEN
         MDOT=I
         INTG=I-1
         I=I+1
         FLAG=1
         GOTO 100
      ENDIF
        IMM1=I-1
34      IF (((M.GT.57).OR.(M.LT.45).OR.(M.EQ.47)))THEN
         CALL CLS
         FLAG=0
         WRITE(*,*)' '
         WRITE(*,*)' ERROR IN NUMERIC INPUT'
         WRITE(*,*)' NUMERIC EXPRESSION EXPECTED'
         WRITE(*,'(1X,A)')NUMIN,NUMIN(I:I)
         WRITE(*,*)' PLEASE RE-ENTER {Q = "QUIT"}'
         READ(*,3)NUMIN
```

```
   3     FORMAT(A)
         IF((NUMIN(:1).EQ.'Q').OR.(NUMIN(:1).EQ.'q'))THEN
            FLG=1
            GOTO 400
         ENDIF
        GOTO 5
      ENDIF
      IF((M.GE.48).AND.(M.LE.57))THEN
         INUM(I)=M-48
         I=I+1
         GOTO 100
      ENDIF
  200 CONTINUE
      DO 300 J=1,I
        IF(MDOT.EQ.0)INTG=I
        IF(J.EQ.MULTFL)GOTO 300
        IF(J.EQ.MDOT)GOTO 300
        IF((MDOT.NE.0).AND.(J.GT.MDOT))GOTO 233
      INUM(J)=(10**(INTG-J))*INUM(J)
      NUM1=NUM1+INUM(J)
      RNUM=RNUM+(1.0*INUM(J))
        GOTO 300
 233    RENUM(J)=((INUM(J)*1.0)/(10.0**(J-MDOT)))
      RNUM=RNUM+RENUM(J)
  300 CONTINUE
  400 CONTINUE
        RNUM=RNUM*FLOAT(MULT)
        NUM1=NUM1*MULT
             IF(NUM1.GT.32000)THEN
                          NUM=0
             ELSE
                          NUM=NUM1
             ENDIF
        RETURN
      END
```

```
         SUBROUTINE DEBLNK(S1,S2,LST,CCSW)
         CHARACTER*(*) S1,S2
         INTEGER LS1,CCSW,FLG
         FLG=0
         LST=LEN(S1)
         NNB=0
         KOUNT=0
3        FORMAT(A70)
         S2=S1
         DO 100 I=1,LST
         IF(S1(I:I).NE.' ')NNB=NNB+1
         IF(S1(I:LST).EQ.' ')THEN
               LST=I-1
               GOTO 450
         ENDIF
100      CONTINUE
450           CONTINUE
400      I=1
         KOUNT=KOUNT+1
300      CONTINUE
         IF(I.GT.LST)GOTO 200
         IF(S1(I:I).EQ.' '.AND.(CCSW.EQ.0.OR.I.EQ.1))THEN
               S1(I:LST-1)=S1(I+1:LST)
               S1(LST:LST)=' '
               LST=LST-1
         ENDIF
         IP1=I+1
         IF(S1(I:I).EQ.' '.AND.S1(IP1:IP1).EQ.' '.AND.CCSW.EQ.1)THEN
               S1(I:LST-1)=S1(I+1:LST)
               S1(LST:LST)=' '
               LST=LST-1
         ENDIF
         I=I+1
         GOTO 300
200      CONTINUE
         IF(KOUNT.GT.LST)GOTO 500
         IF(LST.GT.NNB)GOTO 400
500      CONTINUE
         RETURN
         END
         SUBROUTINE DBLNK(S1,LST,CCSW)
         CHARACTER*(*) S1
         INTEGER LS1,CCSW,FLG
         FLG=0
         LST=LEN(S1)
         NNB=0
         KOUNT=0
3        FORMAT(A70)
         DO 100 I=1,LST
         IF(S1(I:I).NE.' ')NNB=NNB+1
         IF(S1(I:LST).EQ.' ')THEN
               LST=I-1
               GOTO 450
         ENDIF
100      CONTINUE
450           CONTINUE
400      I=1
         KOUNT=KOUNT+1
300      CONTINUE
         IF(I.GT.LST)GOTO 200
         IF(S1(I:I).EQ.' '.AND.(CCSW.EQ.0.OR.I.EQ.1))THEN
               S1(I:LST-1)=S1(I+1:LST)
               S1(LST:LST)=' '
               LST=LST-1
         ENDIF
         IP1=I+1
         IF(S1(I:I).EQ.' '.AND.S1(IP1:IP1).EQ.' '.AND.CCSW.EQ.1)THEN
               S1(I:LST-1)=S1(I+1:LST)
               S1(LST:LST)=' '
               LST=LST-1
         ENDIF
         I=I+1
         GOTO 300
200      CONTINUE
         IF(KOUNT.GT.LST)GOTO 500
         IF(LST.GT.NNB)GOTO 400
500      CONTINUE
         RETURN
         END
```

```
              SUBROUTINE FORMINT(FORM,LEN1,IE,NE,NATOMT,IENTRY)
$LARGE
              CHARACTER*2 ELEM(50)
              CHARACTER*70 FORM,DUMMY
              CHARACTER*4 IE(4),IES(4),IE1(4,400),B,C,TMP
              INTEGER NE(4),M,M1,N(4),L,ITEMP,LEN1,ID(70),NOC1(70),
      +  NOC2(70),IN2C(35,4),IN1C(70,4),I2CH(35),I1CH(50),N2CH,N1CH,
      +  IFM(4),IG(35),NG,INUM(70),NUMBER(70),NNUM,NES(4),NLP,IDLP(70),
      +  NRP,IDRP(70),NSLASH,IDSLSH(70),IFOUND(4),NE1(4,400),NNZERO,
      +  NELT(20),IELEM(10),NEL(20),IEL(4),ISSQ(20)
              COMMON/CONFG1/ELEM
              COMMON/CONFG2/NELEM,N1CH,N2CH,I1CH,I2CH
              DO 234 I=1,NELEM
234           NELT(I)=0
              DO 123 I=1,4
              NES(I)=0
              NE(I)=0
              IES(I)=' '
123           IE(I)=' '
              DO 345 I=1,LEN1
              IF(FORM(I:I).EQ.'['.OR.FORM(I:I).EQ.'{')FORM(I:I)='('
              IF(FORM(I:I).EQ.']'.OR.FORM(I:I).EQ.'}')FORM(I:I)=')'
345           CONTINUE
              NSUBS=0
              NATOMT=0
              NATOM1=0
              IF(LEN1.EQ.1)THEN
                           IE(1)(:1)=FORM(:1)
                           NE(1)=1
                           NATOM=1
                           GOTO 71
*                                        XFER TO OUTPUT/RETURN BLOCK
              ENDIF
************************************************************************
*                          SEARCH FOR NUMERIC CHARACTERS                          *
************************************************************************
              CALL INDXNUM(FORM,LEN1,NNUM,NUMBER,INUM,KFLG)
************************************************************************
*
************************************************************************
*    SEARCH FOR PARENTHESES
*****************************
              CALL INDXST(FORM,'(',NLP,IDLP,LEN1,1,KFLG)
              CALL INDXST(FORM,')',NRP,IDRP,LEN1,1,KFLG)
              CALL INDXST(FORM,'/',NSLASH,IDSLSH,LEN1,1,KFLG)
************************************************************************
              IF(NLP.EQ.0.AND.NRP.EQ.0.AND.NSLASH.EQ.0)THEN
                           CALL FINTERP(FORM,LEN1,IE,NE,NATOMT,IENTRY)
                           GOTO 71
*                                        XFER TO OUTPUT/RETURN BLOCK
              ELSE IF(NSLASH.NE.0.AND.(NLP.NE.0.OR.NRP.NE.0))THEN
*                          CAN'T MIX STRING OPTIONS
                           IENTRY=-3
                           GOTO 71
```

```
*                                        XFER TO OUTPUT/RETURN BLOCK
             ELSE IF(NLP.NE.NRP)THEN
*               UNBALANCED PARENTHESES
                           IENTRY=-2
                           GOTO 71
*                                        XFER TO OUTPUT/RETURN BLOCK
             ELSE IF(NSLASH.NE.0)THEN
*               DECOMPOSE SUBSTRINGS AND PASS PIECEWISE
                           IF(NSLASH.EQ.1.AND.IDSLSH(1).EQ.1)THEN
                                        IENTRY=-4
                                        GOTO 71
*                                        XFER TO OUTPUT/RETURN BLOCK
                           ENDIF
                           IF(IDSLSH(1).NE.1)THEN
                                        DUMMY=' '
                                        DUMMY=FORM(:IDSLSH(1)-1)
                                        CALL SLEN(DUMMY,LENS)
                           CALL FINTERP(DUMMY,LENS,IES,NES,NATOM1,IENTRY)
                                        NATOMT=NATOMT+NATOM1
                                        NSUBS=1
                                        DO 1001 IK=1,4
                                        IE1(IK,1)=IES(IK)
1001                                    NE1(IK,1)=NES(IK)
                           ENDIF
                           DO 2001 JK=1,NSLASH-1
                           DUMMY=' '
                           IF(IDSLSH(JK)+1.GT.IDSLSH(JK+1)-1)GOTO 2001
                           NSUBS=NSUBS+1
                           DUMMY=FORM(IDSLSH(JK)+1:IDSLSH(JK+1)-1)
                           CALL SLEN(DUMMY,LENS)
                           CALL FINTERP(DUMMY,LENS,IES,NES,NATOM1,IENTRY)
                           NATOMT=NATOM1+NATOMT
                           DO 1002 IK1=1,4
                           IE1(IK1,1+JK)=IES(IK1)
1002                       NE1(IK1,1+JK)=NES(IK1)
2001                       CONTINUE
                           IF(LEN1.NE.IDSLSH(NSLASH))THEN
                           DUMMY=' '
                           NSUBS=NSUBS+1
                           DUMMY=FORM(IDSLSH(NSLASH)+1:LEN1)
                           CALL SLEN(DUMMY,LENS)
                           CALL FINTERP(DUMMY,LENS,IES,NES,NATOM1,IENTRY)
                           NATOMT=NATOM1+NATOMT
                           DO 1003 IK1=1,4
                           IE1(IK1,NSUBS)=IES(IK1)
1003                       NE1(IK1,NSUBS)=NES(IK1)
                           ENDIF
                           GOTO 81
*               SUM ELEMENTS OVER ALL SUBSTRINGS
             ELSE IF(NLP.EQ.NRP)THEN
*               DECOMPOSE PARENTHETICAL SUBSTRINGS AND PASS
                           DO 5101 J=1,NRP
                           DO 5201 I=J,NNUM
                           ISSQ(J)=1
```

```
                        IF(INUM(I).EQ.IDRP(J)+1)THEN
                                    ISSQ(J)=NUMBER(I)
                                    GOTO 5101
                        ENDIF
5201                    CONTINUE
5101                    CONTINUE
                        DO 6301 I=1,NRP-1
                        IF(IDRP(I).GT.IDLP(I+1))THEN
                                    WRITE(*,*)' TOO MANY LEVELS OF PARENTHESES'
                                    PAUSE ' hit  return to continue'
                                    IENTRY=-6
                                    GOTO 71
                        ENDIF
6301                    CONTINUE
                        IF(IDRP(NRP).LT.IDLP(NLP))THEN
                                    WRITE(*,*)' TOO MANY LEVELS OF PARENTHESES'
                                    PAUSE ' hit  return to continue'
                                    IENTRY=-6
                                    GOTO 71
                        ENDIF
                        IF(IDLP(1).NE.1)THEN
                                    DUMMY=' '
                                    DUMMY=FORM(:IDLP(1)-1)
                                    NSUBS=NSUBS+1
                                    CALL SLEN(DUMMY,LENS)
                        CALL FINTERP(DUMMY,LENS,IES,NES,NATOM1,IENTRY)
                                    NATOMT=NATOMT+NATOM1
                                    DO 5001 IK=1,4
                                    IE1(IK,NSUBS)=IES(IK)
5001                                NE1(IK,NSUBS)=NES(IK)
787         format(1x,4(4x,a4,i3))
                        ENDIF
                        DO 6201 J=1,NRP
                        NCHARN=1
                        DUMMY=' '
                        IF(ISSQ(J).GT.9)NCHARN=2
                        IF(ISSQ(J).GT.99)NCHARN=3
                        IF(ISSQ(J).GT.999)NCHARN=4
                        IF(ISSQ(J).GT.9999)NCHARN=5
            IF(J.LT.NRP)THEN
                        IF(IDLP(J+1)-1.EQ.IDRP(J))GOTO 6201
                        IF(IDRP(J)+NCHARN+1.LT.IDLP(J+1))THEN
                        DUMMY=FORM(IDRP(J)+NCHARN+1:IDLP(J+1)-1)
                        GOTO 91
                        ENDIF
            ENDIF

            GOTO 6201
91          CONTINUE
                        CALL SLEN(DUMMY,LENS)
                        CALL FINTERP(DUMMY,LENS,IES,NES,NATOM1,IENTRY)
                        NATOMT=NATOM1+NATOMT
                        NSUBS=NSUBS+1
            DO 5301 IK=1,4
            IE1(IK,NSUBS)=IES(IK)
5301        NE1(IK,NSUBS)=NES(IK)
6201        CONTINUE
                          IF(IDRP(NRP)+NCHARN+1.LE.LEN1)THEN
                                       DUMMY=FORM(IDRP(NRP)+NCHARN+1:LEN1)
                                       CALL SLEN(DUMMY,LENS)
                          CALL FINTERP(DUMMY,LENS,IES,NES,NATOM1,IENTRY)
                                       NATOMT=NATOM1+NATOMT
                                       NSUBS=NSUBS+1
                                       DO 5321 IK=1,4
                                       IE1(IK,NSUBS)=IES(IK)
5321                                   NE1(IK,NSUBS)=NES(IK)
                          ENDIF
                          DO 6001 J=1,NRP
                          DUMMY=' '
                          DUMMY=FORM(IDLP(J)+1:IDRP(J)-1)
                          CALL SLEN(DUMMY,LENS)
                          CALL FINTERP(DUMMY,LENS,IES,NES,NATOM1,IENTRY)
                          NATOMT=(NATOM1*ISSQ(J))+NATOMT
                          NSUBS=NSUBS+1
                          DO 5401 IK=1,4
                          IE1(IK,NSUBS)=IES(IK)
5401                      NE1(IK,NSUBS)=(NES(IK)*ISSQ(J))
6001                      CONTINUE
                          GOTO 81
*              SUM ELEMENTS OVER ALL SUBSTRINGS
            ENDIF
*******************************************************************
*       SUBSTRING ELEMENT SUMMATION BLOCK
*******************************************************************
81                        CONTINUE
                          DO 850 IK2=1,NSUBS
                          CALL NATOMS(IK2,IE1,NE1,IEL,NEL,NATOM)
                          DO 950 I1=1,4
950                       NELT(IEL(I1))=NELT(IEL(I1))+NEL(IEL(I1))
850                       CONTINUE
                          NNZERO=0
                          DO 750 I1=1,NELEM
                          IF(NELT(I1).NE.0)THEN
                                       NNZERO=NNZERO+1
                                       IELEM(NNZERO)=I1
                          ENDIF
750                       CONTINUE
                          IF(NNZERO.GT.4)THEN
                                       IENTRY=-1
                                       GOTO 71
*                                      XFER TO OUTPUT/RETURN BLOCK
                          ENDIF
                          DO 650 I1=1,NNZERO
                          NE(I1)=NELT(IELEM(I1))
                          IE(I1)=ELEM(IELEM(I1))
650                       CONTINUE
*******************************************************************
*           OUTPUT/RETURN BLOCK
```

```
************************************************************************
71            CONTINUE
              IF(IENTRY.EQ.0)THEN
                            WRITE(*,*)'FORMULA ENTERED :'
                            WRITE(*,*)FORM
                            call lines(0,2)
                            WRITE(*,*)'  ELEMENTAL COMPOSITION:'
                            call lines(0,1)
                            DO 23 I=1,4
23                          WRITE(*,*)IE(I),NE(I)
              ENDIF
              RETURN
              END
```

```
            SUBROUTINE INDXNUM(S1,LEN1,N,NUM,IND,KFLG)
$LARGE
            CHARACTER*(*) S1,S2*14
            INTEGER N,NUM(132),IND(132),KFLG,LEN1,ITMP(132),NOC
70          FORMAT(A)
            KFLG=0
            N=0
            LSTR=LEN(S1)
            DO 10 I=1,LSTR
10          IND(I)=0
            NOC=0
            DO 200 I=1,LEN1
            IF(ICHAR(S1(I:I)).LE.57.AND.ICHAR(S1(I:I)).GE.48)THEN
                        NOC=NOC+1
                        ITMP(NOC)=I
            ENDIF
200         CONTINUE
            IF(NOC.EQ.0)THEN
                        KFLG=1
                        RETURN
            ENDIF
            IF(NOC.EQ.1)THEN
                        N=1
                        S2=S1(ITMP(1):ITMP(1))
                        CALL READNUM(S2,RN,NUM(1),KFLG)
                        IND(1)=ITMP(1)
                        RETURN
            ENDIF
            I=0
3000        I=I+1
            IF(ITMP(I).NE.ITMP(I+1)-1)THEN
                        N=N+1
                        S2=S1(ITMP(I):ITMP(I))
                        CALL READNUM(S2,RN,NUM(N),KFLG)
                        IND(N)=ITMP(I)
            ELSE IF(ITMP(I).EQ.ITMP(I+1)-1)THEN
                        N=N+1
                        S2=S1(ITMP(I):ITMP(I+1))
                        CALL READNUM(S2,RN,NUM(N),KFLG)
                        IND(N)=ITMP(I)
                        I=I+1
            ENDIF
            IF(I.LT.NOC-1)GOTO 3000
            IF(ITMP(NOC-1)+1.NE.ITMP(NOC))THEN
                        N=N+1
                        S2=S1(ITMP(NOC):ITMP(NOC))
                        CALL READNUM(S2,RN,NUM(N),KFLG)
                        IND(N)=ITMP(NOC)
            ENDIF
            RETURN
            END
```

```
             SUBROUTINE FINTERP(FORM,LEN1,IE,NE,NATOM,IENTRY)
$LARGE
             CHARACTER*2 ELEM(50)
             CHARACTER*70 FORM,DUMMY
             CHARACTER*4 IE(4),B,C,IE1(70),TMP
             INTEGER NE(4),M,M1,N(4),L,NE1(70),ITEMP,LEN1,ID(70),NOC1(70),
      +  NOC2(70),IN2C(35,4),IN1C(70,4),I2CH(35),I1CH(50),N2CH,N1CH,
      +  IFM(4),IG(35),NG,INUM(70),NUMBER(70),NNUM
             COMMON/CONFG1/ELEM
             COMMON/CONFG2/NELEM,N1CH,N2CH,I1CH,I2CH
*    INITALIZE ARRAYS
             DO 1234 I=1,4
             IE(I)=' '
1234         NE(I)=0
*            IF(LEN1.EQ.1)GOTO 1000
**********************************************************************
*                            SEARCH FOR NUMERIC CHARACTERS                         *
**********************************************************************
             CALL INDXNUM(FORM,LEN1,NNUM,NUMBER,INUM,KFLG)
**********************************************************************
*                            BEGIN SEARCH FOR 2 CHARACTER ELEMENTS                 *
**********************************************************************
             I1=0
             DO 1000 I=1,N2CH
             DUMMY=ELEM(I2CH(I))(:2)
             CALL INDXST(FORM,DUMMY,NOC,ID,LEN1,2,KFLG)
             IF(NOC.NE.0)THEN
                          I1=I1+1
                          NOC2(I1)=NOC
                          IFM(I1)=I2CH(I)
                          DO 100 J=1,NOC
                          DO 150 K1=1,NNUM
                          IF(ID(J)+2.EQ.INUM(K1))THEN
                                       NOC2(I1)=NOC2(I1)+NUMBER(K1)-1
                          ENDIF
150                       CONTINUE
100                       IN2C(J,I1)=ID(J)
             ENDIF
             NEL2=I1
             NELTOT=I1
             IF(NELTOT.GT.4)THEN
                          CALL CLS
                          WRITE(*,*)'  ERROR : FORMULA APPEARS TO CONTAIN'
                          WRITE(*,*)'  MORE THAN 4 ELEMENTS and CANNOT BE'
                          WRITE(*,*)' PUT INTO NASA FORMAT FOR USE WITH CHEMKIN.'
                          READ(*,3)B
3                         FORMAT(A4)
                          IENTRY=-1
                          RETURN
             ENDIF
1000         CONTINUE
```

```
*
**********************************************************************
*                          BEGIN SEARCH FOR 1 CHARACTER ELEMENTS                *
**********************************************************************
             I1=0
             N1C=0
             DO 2000 I=1,N1CH
             DUMMY=ELEM(I1CH(I))(:1)
             N1CS=N1C
             CALL INDXST(FORM,DUMMY,NOC,ID,LEN1,1,KFLG)
             IF(NOC.NE.0)THEN
                          ISAME=0
                          I1=I1+1
             DO 30 J=1,NEL2
             IF(ELEM(I1CH(I))(:1).EQ.ELEM(IFM(J))(:1))THEN
                          ISAME=1
                          NO1CA=0
                          NG=0
             DO 41 I3=1,NOC
             DO 40 I2=1,NOC2(J)
                          IF(ID(I3).EQ.IN2C(I2,J))THEN
                                       NG=NG+1
                                       IG(NG)=I3
                          ENDIF
40           CONTINUE
41           CONTINUE
             DO 45 I2=1,NOC
             IST=0
                          DO 46 I3=1,NG
46                        IF(I2.EQ.IG(I3))IST=1
                          IF(IST.EQ.0)THEN
                                       NO1CA=NO1CA+1
                          DO 350 K1=1,NNUM
                          IF(ID(I2)+1.EQ.INUM(K1))THEN
                                       NO1CA=NO1CA+NUMBER(K1)-1
                          ENDIF
350                       CONTINUE
                                       IN1C(I2,I1)=ID(I2)
                          ENDIF
45           CONTINUE
             ENDIF

30           CONTINUE
             IF(ISAME.EQ.0)THEN
                          NOC1(I1)=NOC
                          IFM(NEL2+I1)=I1CH(I)
                          DO 50 I2=1,NOC
                          DO 250 K1=1,NNUM
                          IF(ID(I2)+1.EQ.INUM(K1))THEN
                                       NOC1(I1)=NOC1(I1)+NUMBER(K1)-1
                          ENDIF
250                       CONTINUE
50                        IN1C(I2,I1)=ID(I2)
                          NIC=I1
```

```
            ELSE IF(ISAME.NE.0)THEN
                    IF(NO1CA.EQ.0)THEN
                            I1=I1-1
                    ELSE IF(NO1CA.NE.0)THEN
                            IFM(NEL2+I1)=I1CH(I)
                            NOC1(I1)=NO1CA
                    ENDIF
                    NIC=I1
            ENDIF
            IF(NEL2+I1.GT.4)THEN
                    CALL CLS
                    WRITE(*,*)'  ERROR : FORMULA APPEARS TO CONTAIN'
                    WRITE(*,*)'  MORE THAN 4 ELEMENTS and CANNOT BE'
                    WRITE(*,*)' PUT INTO NASA FORMAT FOR USE WITH CHEMKIN.'
                    READ(*,3)B
                    IENTRY=-1
                    RETURN
            ENDIF
            ENDIF
2000        CONTINUE
            NEL1=I1
            NATOM=0
            DO 324 J2=1,NEL1
324                 NATOM=NATOM+NOC1(J2)
            DO 325 J2=1,NEL2
325                 NATOM=NATOM+NOC2(J2)
            DO 60 I=1,NEL2
            IE(I)(:2)=ELEM(IFM(I))
60          NE(I)=NOC2(I)
            DO 70 I=1,NEL1
            IE(I+NEL2)(:2)=ELEM(IFM(I+NEL2))
70          NE(I+NEL2)=NOC1(I)

71          CONTINUE
            RETURN
            END
            SUBROUTINE SLEN (S1,LSTR)
$LARGE
            CHARACTER*(*) S1
            INTEGER LSTR, I
            LSTR=LEN(S1)
            IF(S1.EQ.' ')THEN
                        LSTR=0
                        GOTO 2000
            ENDIF
            I=0
1000        I=I+1
            IF(S1(I+1:LSTR).EQ.' ')THEN
                        LSTR=I
            ELSEIF(I.LE.LSTR)THEN
                        GOTO 1000
            ENDIF
2000        CONTINUE
            RETURN
            END
```

```
         SUBROUTINE INDXST(S1,S2,N,INDX,LS1,LS2,FLG)
$LARGE
         CHARACTER S1*(*),S2*(*)
         INTEGER N,INDX(*),LS1,LS2,FLG
********************************************************************************
*                                                                              *
*  SUBROUTINE INDXST  :   INDeX STring                                         *
*                                                                              *
*                                                                              *
*    The purpose of this subroutine is to search string S1  (arbitrary         *
*   length) for all occurrences of subsrting S2.   S2 must be shorter          *
*   than S1.  Returned are the number of occurrences of S2 in S1;  the         *
*   vector of starting addresses of each occurrence; string lengths for        *
*   both S1 and S2 if not supplied; and an error flag.                         *
*                                                                              *
*   Variable definitions:                                                      *
*     S1 - input string to be searched (character length defined by            *
*          calling routine).                                                   *
*     S2 - target substring; length determined by calling routine.             *
*     N  - number of occurences of substring S2 in S1.                         *
*  INDX(*)- vector giving the starting address for each occurrence of          *
*          S2 in S1.                                                           *
*     LS1- length (non-blank) of S1; if not supplied it is determined          *
*          and returned to the calling routine.                                *
*     LS2- length ( = 1 for a single blank ) of target substring S2;           *
*          if not supplied it is determined and returned to the calling        *
*          routine.                                                            *
*     FLG- error flag.  FLG = 0 if no error encountered                        *
*                       FLG < 0 if LS2 > LS1 or other error occurrs            *
*                                                                              *
*                                                                              *
********************************************************************************
              LMAX=LEN(S1)
        DO 10 I=1,LMAX
10      INDX(I)=0
              N=0
              FLG=0
*       DETERMINE STRING LENGTH IF NOT PASSED TO THIS ROUTINE
        IF(LS1.EQ.0.AND.LS2.EQ.0)THEN
                                    DO 100 I=1,LMAX
                       IP1=I+1
                       IF(I.GT.1)THEN
                                           IM1=I-1
                             ELSE
                                           IM1=1
                             ENDIF
                       IF(LS1.EQ.0.AND.S1(I:I).EQ.' '.AND.S1(IP1:IP1).EQ.' ')
     +           LS1=IM1
                       IF(LS2.EQ.0.AND.S2(I:I).EQ.' '.AND.S2(IP1:IP1).EQ.' ')
     +           LS2=IM1
100                 CONTINUE
              ENDIF
150     CONTINUE
              IF(LS1.LT.LS2)THEN
```

```
                           FLG=-10
                           RETURN
              ENDIF
         IFOUND=0
         DO 200 I=1,LS1
         I2=I+LS2-1
         IF((I2-I).GT.LS2)GOTO 200
         IF(S1(I:I2).EQ.S2(:LS2))THEN
                 IFOUND=IFOUND+1
                 INDX(IFOUND)=I
         ENDIF
200      CONTINUE
         N=IFOUND
         RETURN
         END
```

```
      SUBROUTINE MATBAL(NLHS,LX,NRHS,RX,IE1,NE1,COEF,IERR)
      INTEGER LX(*),RX(*),NE1(4,400),IELL(4),NELL(50),IELR(4),
     +NELR(20),NELLT(20),NELRT(20),NELEM,I1CH(50),I2CH(35),LNATOM,
     +RNATOM
      REAL*8 COEF(*),DNELL(50),DNELLT(50),DNELR(50),DNELRT(50)
      CHARACTER*4 IE1(4,400)
      CHARACTER*2 ELEM(50)
      COMMON/CONFG1/ELEM
      COMMON/CONFG2/NELEM,N1CH,N2CH,I1CH,I2CH
      DO 10 I=1,NELEM
      DNELLT(I)=0.0
      DNELL(I)=0.0
      NELLT(I)=0
      DNELR(I)=0.0
      DNELRT(I)=0.0
10    NELRT(I)=0
      LNATOM=0
      RNATOM=0
      DO 800 I=1,NLHS
      CALL NATOMS(LX(I),IE1,NE1,IELL,NELL,NATOM)
      DO 850 I1=1,NELEM
      DNELL(I1)=COEF(I)*FLOAT(NELL(I1))
      DNELLT(I1)=DNELLT(I1)+DNELL(I1)
850   CONTINUE
      LNATOM=NATOM
800   CONTINUE
      DO 900 I=1,NRHS
      CALL NATOMS(RX(I),IE1,NE1,IELL,NELL,NATOM)
      DO 950 I1=1,NELEM
      DNELR(I1)=COEF(I+NLHS)*FLOAT(NELL(I1))
      DNELRT(I1)=DNELRT(I1)+DNELR(I1)
950   CONTINUE
      RNATOM=NATOM
900   CONTINUE
      DO 1000 I=1,NELEM
      IF(INT(100.*DNELRT(I)).NE.INT(100.*DNELLT(I)))THEN
              WRITE(*,*)' REACTION DOES NOT BALANCE!'
              WRITE(*,*)'# of ',ELEM(I),' on LHS= ',DNELLT(I)
              WRITE(*,*)'# of ',ELEM(I),' on RHS= ',DNELRT(I)
              IERR=1
              RETURN
      ENDIF
1000  CONTINUE
      IERR=0
      RETURN
      END
```

```
             SUBROUTINE NATOMS(J,IE1,NE1,IEL,NEL,NATOM)
             INTEGER NE1(4,400),NELEM,N1CH,N2CH,I1CH(50),I2CH(35),IEL(*),
     +NEL(*),NATOM
             CHARACTER*2 ELEM(50)
             CHARACTER*4 IE1(4,400)
             CHARACTER*14 NUL
             COMMON/CONFG1/ELEM
             COMMON/CONFG2/NELEM,N1CH,N2CH,IICH,I2CH
             DO 10 I=1,4
10           IEL(I)=0
             DO 20 I=1,NELEM
20           NEL(I)=0
             NATOM=0
             I1=0
             DO 1000 I=1,4
             DO 1050 K=1,NELEM
             IF(ELEM(K)(:2).EQ.IE1(I,J)(:2))THEN
                          I1=I1+1
                          IEL(I1)=K
                          NEL(K)=NEL(K)+NE1(I,J)
             ENDIF
1050         CONTINUE
1000         CONTINUE
             RETURN
             END
```

```
        SUBROUTINE ATOMSORT(ISORT,NE1,IE1,ICNT,KEY)
        INTEGER ISORT(1),NE1(4,400),ICNT,LEVEL,KEY(10),IL1(50),
     +IL2(50),INC,ICTRL,IK,J,I,K,ITMP,NELEM,N1CH,N2CH,I1CH(50),
     +I2CH(35),IEL(4),NEL(50)
             CHARACTER*2 ELEM(50)
        CHARACTER*4 IE1(4,400)
        CHARACTER*14 NUL
             COMMON/CONFG1/ELEM
             COMMON/CONFG2/NELEM,N1CH,N2CH,I1CH,I2CH
             COMMON/CONFG5/NKEYS
C       INITIALIZE INDIRECT ADDERSS VECTOR: ISORT
        DO 10 I=1,ICNT
10      ISORT(I)=I
C       INITIALIZE DEFAULT SORT KEY VECTOR. KEY IS DEFINED AS THE FIRST
C            6 ELEMENTS LISTED IN FILE : "THERM.CFG"
                DO 20 I=1,NKEYS
20              KEY(I)=I
        DO 10000 LEVEL=1,NKEYS
        ICTRL=0
        J=0
        DO 1000 I=1,ICNT-1
                IF(ICTRL.EQ.1)RETURN
                ICTRL=1
                        INC=1
                J=0
                DO 2000 IK=1,ICNT-I
                J=J+INC
        CALL NATOMS(ISORT(J),IE1,NE1,IEL,NEL,NATOM)
             DO 1500 I1=1,NELEM
1500         IL1(I1)=NEL(I1)
        CALL NATOMS(ISORT(J+INC),IE1,NE1,IEL,NEL,NATOM)
             DO 1600 I1=1,NELEM
1600         IL2(I1)=NEL(I1)
        IF(LEVEL.EQ.1)THEN
                K=LEVEL
                IF(IL2(KEY(K)).GT.IL1(KEY(K)))GOTO 2000
                GOTO 150
        ENDIF
        DO 100 K=1,LEVEL
                IF(IL2(KEY(K)).GT.IL1(KEY(K)))GOTO 2000
100     CONTINUE
150     CONTINUE
                ICTRL=0
                ITMP=ISORT(J)
                ISORT(J)=ISORT(J+INC)
                ISORT(J+INC)=ITMP
2000    CONTINUE
1000    CONTINUE
10000   CONTINUE
3       FORMAT(A14)
        CALL CLS
        RETURN
        END
```

```
      SUBROUTINE LOADSP(AS,A,IFIRST,ILAST,NGRPT,CS,B,NG,NSAVE,SYMM,SYMS,
     +SYMR,SCOR,SCORP,FORM,FORMP,SPECP,IE,NE,NATOM,FLAG,BFLAG,FLG,LS,RS,
     +RS22,FILST,DOCFIL,IX,E,LENF,EP,FILNM,DOC,NDOC,TEXT,TEXTS,KOUNT,
     +IOP,IDOC,NSPEC,BD,LENFP,IENTER,JC,DELB,DOCTYPE,KDOCT,DELTA)
      LOGICAL EXST,OPNED
      REAL*4 B(2,9),BO(9)
      REAL A(400,9),SCOR,SCORP,DELTA(1)
      INTEGER FLAG,BFLAG,FLG,SYMM,SYMS,SYMR,NG,NSAVE,NGRPT,ILAST(1),
     +IFIRST(1),NE(1),NATOM,LNG(60),NOCCUR,IND(70),U,E(1),IX(1),Y,
     +IND1,IND2,EP(50),IDOC(1),NSPEC,ICOUNT,KDOCT,JIND(70),NE1(4,400)
             CHARACTER*70 TEXT(1),TEXTS(1)
        CHARACTER*70 DOC1(100)
             CHARACTER*14 DOC(1)
      CHARACTER*70 SI,NUL,NUL1,DUMMY,FORM,FORMP,TXT,FILNM,FILNM2
      CHARACTER*14 AS(1),NUMIN,SPEC,SPECP,CS(2,60),DOCFIL,RS,RS22,
     +LS,DOCTYPE,DUM1,DUM2,DUM3,DUM4,A1,UNITS,FUNITS
      CHARACTER*4 IE(1),IE1(4,400)
             COMMON/DELETE/NDEL,IDEL(500)
      COMMON/CONTRL/FIL
             COMMON/CONFG6/UNITS
             COMMON/ELEM1/NE1
             COMMON/ELEM2/IE1
             COMMON/NROTOR/NROT,NROTS
70           FORMAT(A70)
             ILIST=0
             DO 10 II=1,KOUNT
             TEXT(II)=' '
             TEXTS(II)=' '
10    CONTINUE
             KOUNT=0
        DO 92 I=1,9
92      DELTA(I)=0.0
        IF(KDOCT.EQ.0.OR.KDOC.EQ.1)DOCTYPE=' '
        IF(KDOCT.EQ.2)DOCTYPE='NOCALC'
        LS=' '
        FORM=' '
        FORMP=' '
      SPEC=' '
      NUL=' '
      LEN=0
      SI=' '
      NG=0
      DO 20 I=1,9
      B(1,I)=0
20    B(2,I)=0
        IF(IENTER.EQ.1.AND.IOP.NE.0)THEN
                SPEC=SPECP
                IENTER=0
                J1=0
                GOTO 1777
        ENDIF
        IF((IENTER.EQ.2.OR.IENTER.EQ.10).AND.IOP.NE.0)THEN
                J1=JC
                          U=34
                GOTO 1878
        ENDIF
        SPECP=' '
        IF(IOP.EQ.0)THEN
        CALL CLS
      WRITE(*,700)
      WRITE(*,701)
700   FORMAT(' ',15X,'enter document filename')
701   FORMAT('0',15X,'for species input')
      CALL LINES(0,2)
      IF(FILNM.NE.' ')WRITE(*,708)FILNM
708   FORMAT(' ',1X,'{ ',A70,' }')
      READ(*,70)FILNM2
1     FORMAT(A14)
      CALL READCHAR(FILNM2)
      IF(FILNM2.EQ.'Q'.OR.(FILNM2.EQ.' '.AND.FILNM.EQ.' '))THEN
           FLG=1
           RETURN
      ENDIF
        IF(FILNM2.NE.' ')FILNM=FILNM2
             CALL FNAME(FILNM,'DOC')
1000  INQUIRE(FILE=FILNM,EXIST=EXST,OPENED=OPNED,ERR=90000)
      IF(OPNED)THEN
                          INQUIRE(34,OPENED=OPNED)
                          IF(OPNED)THEN
                                       U=34
                                       GOTO 1003
                          ENDIF
                          INQUIRE(52,OPENED=OPNED)
                          IF(OPNED)THEN
                                       U=52
                                       GOTO 1003
                          ENDIF
      ENDIF
      IF(EXST)THEN
           OPEN(34,FILE=FILNM,ACCESS='DIRECT',RECL=70,FORM='FORMATTED'
     +,STATUS='OLD')
           U=34
           GOTO 1003
      ENDIF
      CALL CLS
      WRITE(*,702)FILNM
702   FORMAT(' ',10X,A14,2X,'does not exist')
      WRITE(*,703)
703   FORMAT(' ',10X,'{ hit return to continue }')
      READ(*,3)NUL
3     FORMAT(A70)
4       FORMAT(' ',A70)
      IF(NUL.EQ.' ')GOTO 10
      FLG=1
      RETURN
             ENDIF
1003  CONTINUE
        IF(IENTER.EQ.3)THEN
```

```
                            NDEL=0
                            GOTO 1066
             ENDIF
             IF(IOP.NE.1)THEN
                           NDEL=0
                           ILIST=1
                           GOTO 1066
             ENDIF
             CALL CLS
      WRITE(*,704)
704   FORMAT('0',10X,'enter species id ( "?" for options)')
      CALL LINES(0,3)
      READ(*,1)SPEC
      CALL READCHAR(SPEC)
      IF(SPEC.EQ.'Q'.OR.SPEC.EQ.' ')THEN
           FLG=1
           RETURN
      ENDIF
        NUL1=' '
        NUL1(:14)=SPEC(:14)
      CALL DBLNK(NUL1,LENSP,0)
        SPEC(:14)=NUL1(:14)
        IF(SPEC.EQ.'?')THEN
        CALL CLS
        WRITE(*,*)'                    READ SPECIES DOCUMENT OPTIONS'
        WRITE(*,*)'                    _____________________________'
        CALL LINES(0,1)
        WRITE(*,*)'                      1  -  CHANGE DATAFILE'
        WRITE(*,*)'                      2  -  CONTINUE'
        IF(IOP.NE.0)THEN
                     WRITE(*,*)'                      3  -  LIST SPECIES'
                     CALL LINES(0,1)
        ENDIF
        WRITE(*,*)'                           enter option: { 2 }'
        READ(*,1)NUMIN
        CALL READCHAR(NUMIN)
        IF (NUMIN(:1).EQ.' '.OR.NUMIN(:1).EQ.'2')GOTO 1003
        IF(NUMIN(:1).EQ.'1')THEN
                IOP=0
                CLOSE(34,STATUS='KEEP')
                GOTO 10
        ENDIF
        IF(IOP.NE.0)THEN
        IF(NUMIN(:1).EQ.'3')THEN
                          ilist=1
                          goto 41234
23              FORMAT(1X,3(5X,A10))
1067            CONTINUE
        ENDIF
        GOTO 1003
        ENDIF
        GOTO 1003
        ENDIF
1066    CONTINUE
```

```
        IF(IOP.EQ.0)THEN
                NSPEC=0
                CALL CLS
        write(*,*)'                      INITIALIZING'
             WRITE(*,*)'              THERMO DOCUMENTATION FILE'
        CALL LINES(0,3)
                I=0
                J2=0
                NJ=0
        WRITE(*,*)'                       PLEASE WAIT'
        CALL LINES(0,3)
             I=0
             NDEL=0
             READ(U,'(A70)',REC=1)NUL
10000        I=I+1
        IF(I.NE.1)READ(U,'(A70)',REC=I,END=99000)NUL
                          CALL DBLNK(NUL,LEN,0)
                          IF(NUL(:10).EQ.'ENDSPECIES')THEN
                                       JSAVE=NSPEC
                                       GOTO 10000
                          ENDIF
                IF(NUL(:7).EQ.'SPECIES'.OR.NUL(:8).EQ.' SPECIES'
     +.OR.NUL(:9).EQ.'0 SPECIES'.OR.NUL(:9).EQ.'  SPECIES'.OR.NUL(:7)
     +.EQ.'DELETED'.OR.NUL(:8).EQ.' DELETED'
     +.OR.NUL(:9).EQ.'0 DELETED'.OR.NUL(:9).EQ.'  DELETED')THEN
                       NSPEC=NSPEC+1
                       J2=J2+1
                                       I=I+1
                       IDOC(J2)=I+1
                          IF(NUL(:7).EQ.'DELETED'.OR.NUL(:8).EQ.' DELETED'
     +.OR.NUL(:9).EQ.'0 DELETED'.OR.NUL(:9).EQ.'  DELETED')THEN
                                       NDEL=NDEL+1
                                       IDEL(NDEL)=J2
                          ENDIF
                                       READ(U,'(A70)',REC=I,END=99000)NUL
                                       DOC(J2)=NUL(:14)
7               format(1x,5(2x,a10))
                          I=I+2
                          READ(U,'(A70)',REC=I,END=99000)NUL
                          CALL LRDCHAR(NUL)
                          CALL DBLNK(NUL,LEN,1)
                          CALL INDXST(NUL,' ',NOCCUR,IND,LEN,1,FLG)
                          IF(NOCCUR.EQ.0)THEN
                                       WRITE(*,*)'ERROR : CAN''T FIND SPECIES FORMULA'
                                       WRITE(*,*)DOC(J2)
                                       RETURN
                          ENDIF
                          FORM=NUL(IND(1)+1:LEN)
                          LENF=LEN-IND(1)
                          CALL FORMINT(FORM,LENF,IE,NE,NATOM,1)
             WRITE(*,876)J2,DOC(J2)(:14),(IE(JCNT),NE(JCNT),JCNT=1,4)
876          FORMAT(1X,I3,2X,A14,2X,4(2X,A4,I3))
                          DO 1776 IKT=1,4
                          NE1(IKT,J2)=NE(IKT)
                          IE1(IKT,J2)=IE(IKT)
1776                      CONTINUE
                ENDIF
             GOTO 10000
99000           CONTINUE
                          IF(NSPEC.NE.JSAVE)NSPEC=JSAVE
                          NDOC=I
             IOP=1
             ENDIF
41234   continue
             IF(COMMAND.NE.'AUTO_RECALC'.and.iop.eq.1)THEN
             if(ilist.eq.1)then
                          nul(:1)='?'
                          goto 699
             endif
******************** LIST SPECIES SCREEN  *******************
*********************  DELETE/UNDELETE/VIEW ******************
             WRITE(*,*)'{ hit return to continue : ? for options }'
             read(*,3)NUL
699          IF(NUL(:1).EQ.'?')THEN
600          CALL CLS
             ILINE=0
             DO 5235 IJKN=1,NSPEC+4,4
             ILINE=ILINE+1
             IF(ILINE.GT.22)THEN
                          WRITE(*,*)'{ hit return or enter option " D / U / V " }'
                          read(*,'(a70)')nul
                          call readchar(nul)
                          if(nul.eq.'D'.or.nul.eq.'U'.or.nul.eq.'V')then
                                       goto 5333
                          else
                                       iline=0
                          endif
             ENDIF
             dum1=' '
             dum2=' '
             dum3=' '
             dum4=' '
             IJK2=IJKN+1
             IJK3=IJKN+2
             IJK4=IJKN+3
             call delchk(ijkn,i1)
             call delchk(ijk2,i2)
             call delchk(ijk3,i3)
             call delchk(ijk4,i4)
             IF(IJKN.LE.NSPEC.AND.I1.EQ.0)DUM1=DOC(IJKN)(:14)
             IF(IJK2.LE.NSPEC.AND.I2.EQ.0)DUM2=DOC(IJK2)(:14)
             IF(IJK3.LE.NSPEC.AND.I3.EQ.0)DUM3=DOC(IJK3)(:14)
             IF(IJK4.LE.NSPEC.AND.I4.EQ.0)DUM4=DOC(IJK4)(:14)
             IF(DUM1.EQ.' '.AND.DUM2.EQ.' '.AND.DUM3.EQ.' '.AND.DUM4.EQ.' ')
     +  GOTO 5235
             WRITE(*,5236)IJKN,DUM1,IJK2,DUM2,IJK3,DUM3,IJK4,DUM4
5236         FORMAT(1X,4(I3,1X,A14))
5235         CONTINUE
```

```
************************* GET SPECIES OPTIONS *******************************
      WRITE(*,*)' enter "D" to delete; "U" to undelete; "V" to view; or
     +hit return to continue'
             read(*,3)nul
             call lrdchar(nul)
             if(nul.eq.' ')goto 767
5333    continue
*-----------------------------------------------------------------
*          delete species
*-----------------------------------------------------------------
        IF(NUL(:1).EQ.'D')THEN
5234            WRITE(*,*)'DELETE which species { 0 to abort }'
                WRITE(*,*)'    or enter species number to delete}'
                read(*,*,err=600)idelete
                          CALL DELCHK(IDELETE,ID)
                if(idelete.EQ.0.OR.ID.EQ.1)goto 600
                ndel=ndel+1
                idel(ndel)=idelete
                write(*,*)'species # ',idelete
                write(*,*)'has been deleted'
                          J55=IDOC(IDELETE)-2
                          NUL=' DELETED'
                          WRITE(U,'(A70)',REC=J55,ERR=600)NUL
                          goto 600
        endif
573     continue
*-----------------------------------------------------------------
*                undelete species
*-----------------------------------------------------------------
        IF(NUL(:1).EQ.'U')THEN
                CALL CLS
                DO 765 IB=1,NDEL
765             WRITE(*,766)IDEL(IB),DOC(IDEL(IB))(:14)
766               FORMAT(1X,I3,1X,'-',1X,A14)
                     CALL LINES(0,1)
                     WRITE(*,*)' undelete which species { 0 to abort }'
                     write(*,*)'     or enter species number to undelete'
                     read(*,*,err=600)idelete
                     if(idelete.eq.0)goto 600
                     do 769 ib=1,ndel
                if(idel(ib).ne.idelete)goto 769
                          J55=IDOC(IDELETE)-2
                          NUL=' SPECIES'
                          WRITE(U,'(A70)',REC=J55,ERR=600)NUL
                do 770 ib1=ib,ndel-1
770             IDEL(ib1)=idel(ib1+1)
                ndel=ndel-1
769               continue
                     GOTO 600
        endif
*-----------------------------------------------------------------
*                   view species
*-----------------------------------------------------------------
             IF(NUL(:1).EQ.'V')THEN
```

```
                    WRITE(*,*)' view which species { enter species number }'
                    read(*,*,err=600)iview
                    if(iview.lt.nspec)then
                              j55=idoc(iview)-2
                              j56=idoc(iview+1)-4
                    elseif(iview.eq.nspec)then
                              j55=idoc(iview)-2
                              j56=ndoc-4
                    else
                              goto 600
                    endif
                    call cls
                    iline=0
                    do 9123 irecord=j55,j56
                    iline=iline+1
                    if(iline.ge.22)then
                              pause '{ hit return to continue }'
                              iline=0
                    else
                              read(U,'(a70)',rec=irecord,err=600,end=600)nul
                              write(*,'(1x,a70)')nul
                    endif
9123                continue
                    pause '{ hit return to continue }'
                    goto 600
          ENDIF
********************************************************************************
767   continue
          ENDIF
********************  end of if (nul(:1).eq.'?') block   **********************
          ENDIF
      IF(IENTER.EQ.3)RETURN
          if(ilist.eq.1)then
                    ilist=0
                    goto 1003
          endif


      J1=0
1777  J1=J1+1
1778  CONTINUE
          IF(J1.GT.NSPEC)GOTO 95000
          DUMMY=' '
          DUMMY(:14)=DOC(J1)(:14)
          CALL DBLNK(DUMMY,NUM,0)
          DOC(J1)(:14)=DUMMY(:14)
          IF(SPEC(:14).NE.DOC(J1)(:14).AND.J1.LT.NSPEC)GOTO 1777
1878                CONTINUE
*                   REWIND(U)
*                   DO 99099 J5=1,IDOC(J1)-2
                    READ(U,'(A70)',REC=IDOC(J1)-2,END=90000)NUL
99099               CONTINUE
CWRITE(*,*)NUL
                    IF(J1.LT.NSPEC)THEN
```

```
                                        DO 99088 J5=1,(IDOC(J1+1)-IDOC(J1))
                                        J55=IDOC(J1)-2+J5
                                        READ(U,'(A70)',REC=J55,END=90000)NUL
                                        DOC1(J5)=NUL(:70)
99088                                   CONTINUE
CWRITE(*,*)DOC1(J5)
                           ENDIF
                           IF(J1.EQ.NSPEC)THEN
                                        DO 99089 J5=1,(NDOC-IDOC(J1)-1)
                                        J55=IDOC(J1)-2+J5
                                        READ(U,'(A70)',REC=J55,END=90000)NUL
99089                                   DOC1(J5)=NUL(:70)
                           ENDIF
C             READ DOCUMENT ARRAY JUST LOADED!
              FUNITS='KCAL'
        IDAT=0
        DELB=0.0
        ISTAT=0
        TYPE='M'
        IDONE=0
              NROT=0
        FLG=0
              J=0
1254          J=J+1
        CALL DBLNK(DOC1(J),NUM,1)
              IF(J1.LT.NSPEC.AND.(J.GT.IDOC(J1+1)-IDOC(J1)))GOTO 95000
              IF(J1.EQ.NSPEC.AND.(J.GT.NDOC-IDOC(J1)))GOTO 95000

        J=J+2
1255      CALL DBLNK(DOC1(J),LEN,1)

      CALL INDXST(DOC1(J),' ',NOCCUR,IND,LEN,1,FLG)
        INDX=IND(1)-1
      LS(:INDX)=DOC1(J)(:INDX)
        FORM=' '
      FORM(:LEN-(IND(1)+1)+1)=DOC1(J)(IND(1)+1:LEN)
        FORMP=FORM
        LENF=LEN-(IND(1)+1)+1
      CALL FORMINT(FORM,LENF,IE,NE,NATOM,1)
20000   CONTINUE
        J=J+1
      CALL DBLNK(DOC1(J),LEN,0)
             IF(DOC1(J)(:6).EQ.'UNITS:')THEN
                           IF(DOC1(J)(7:8).EQ.'KJ')FUNITS='KJ'
                           GOTO 20000
             ENDIF
        IF(DOC1(J)(:6).EQ.'NOCALC'.AND.(KDOC.EQ.0.OR.KDOC.EQ.2))
     +THEN
                DOCTYPE='NOCALC'
                GOTO 20000
        ENDIF
        IF(DOC1(J)(:7).EQ.'RADICAL'.OR.DOC1(J)(:8).EQ.'ORADICAL'
     +.OR.(TYPE.EQ.'R'.AND.IDONE.NE.1))THEN
        BFLAG=1
        IF(TYPE.EQ.'M')THEN
      IF(DOC1(J)(:10).EQ.'ENDSPECIES')GOTO 95000
                TYPE='R'
        IF(DOC1(J)(18:23).EQ.'PARENT')SPECP=DOC1(J)(24:LEN)
        IF(DOC1(J)(17:22).EQ.'PARENT')SPECP=DOC1(J)(23:LEN)
                RS=LS
                LS=SPECP
        ENDIF
                J=J+1
                CALL DBLNK(DOC1(J),LEN,0)
                IF(DOC1(J)(:LEN).EQ.'ENDSPECIES')GOTO 95000
                IF(DOC1(J)(:13).NE.'PARENTFORMULA')GOTO 20000
                IF(LEN.GT.13)THEN
                        FORMP=DOC1(J)(14:LEN)
                        LENFP=LEN-14+1
                ENDIF
                IF(LEN.EQ.13)THEN
                        J=J+1
                        CALL DBLNK(DOC1(J),LEN,0)
                        FORMP=DOC1(J)(:LEN)
                        LENFP=LEN
                ENDIF
                J=J+1
                CALL DBLNK(DOC1(J),LEN,0)
                IF(DOC1(J)(:14).NE.'PARENTSYMMETRY')GOTO 20000
                NUMIN=' '
                NUMIN=DOC1(J)(15:LEN)
                CALL READNUM(NUMIN,RNUM,SYMS,KFLG)
                IDONE=1

        ENDIF
C             SEARCHING FOR KEYWORD "GROUPS"
      IF(DOC1(J)(:6).NE.'GROUPS'.AND.J.LT.(IDOC(J1+1)-IDOC(J1)))
     +GOTO 20000
      NG=0
      IF(LEN.EQ.7)THEN
           NG=ICHAR(DOC1(J)(7:7))-48
           GOTO 1111
      ENDIF
      IF(LEN.EQ.8)THEN
           NG=10*(ICHAR(DOC1(J)(7:7))-48)+ICHAR(DOC1(J)(8:8))-48
           GOTO 1111
      ENDIF
1111  CONTINUE
        IF(NG.EQ.0)THEN
                DOCTYPE='NOCALC'
                GOTO 3001
        ENDIF

        NSAVE=NG
        J=J+1
        I1=0
223     CONTINUE
        J=J+1
        IF(J.LT.(NDOC-INX).AND.DOC1(J).EQ.' ')GOTO 223
      CALL DBLNK(DOC1(J),LEN,0)
        IF(J.LT.NDOC.AND.DOC1(J)(:2).EQ.'Gr')GOTO 223
      CALL INDXST(DOC1(J),'-',NOCCUR,IND,LEN,1,FLG)
      LEN1=LEN
        IF(NOCCUR.EQ.4)THEN
      CALL INDXST(DOC1(J),'|',NOCR,JIND,LEN1,1,FLG)
        IGS1=1
        IGE1=IND(1)-1
        IGS2=JIND(1)+1
        IGE2=IND(3)-1
        NUMIN=' '
        NUMIN=DOC1(J)(IGS1:IGE1)
        CALL READNUM(NUMIN,RNUM,KG1,KFLG)
        NUMIN=' '
        NUMIN=DOC1(J)(IGS2:IGE2)
        CALL READNUM(NUMIN,RNUM,KG2,KFLG)
        JINDS=JIND(1)-1
        IN1P1=IND(1)+1
        IN2M1=IND(2)-1
        IN1P2=IND(3)+1
        IN2M2=IND(4)-1
                IND1=IND(2)+1
                IND3=IND(4)+1
      IF(JINDS.EQ.IND1)E(KG1)=ICHAR(DOC1(J)(IND1:IND1))-48
      IF(LEN1.EQ.IND3)E(KG2)=ICHAR(DOC1(J)(IND3:IND3))-48
      IF(JINDS.EQ.IND(2)+2)THEN
                IND2=IND(2)+2
           E(KG1)=10*(ICHAR(DOC1(J)(IND1:IND1))-48)+ICHAR(DOC1(J)
     +(IND2:IND2))-48
      ENDIF
      IF(LEN1.EQ.IND(4)+2)THEN
                IND2=IND(2)+2
                IND4=IND(4)+2
        E(KG2)=10*(ICHAR(DOC1(J)(IND3:IND3))-48)+ICHAR(DOC1(J)(IND4:
     +IND4))-48
      ENDIF
        EP(KG1)=E(KG1)
        EP(KG2)=E(KG2)
      LNG(KG1)=IND(2)-IND(1)-1
      LNG(KG2)=IND(4)-IND(3)-1
      CS(1,KG1)(:LNG(KG1))=DOC1(J)(IN1P1:IN2M1)
        CS(1,KG2)(:LNG(KG2))=DOC1(J)(IN1P2:IN2M2)
             CALL FINDGR(AS,NGRPT,CS,IX,KG2,ISTAT,LNG(I))
             IF(ISTAT.EQ.1)THEN
                          CS(2,KG1)=' '
                          CS(2,KG1)(:5)='ERROR'
                          IX(KG1)=0
             ELSE
                     CS(2,KG1)=CS(1,KG1)
             ENDIF
             CALL FINDGR(AS,NGRPT,CS,IX,KG1,ISTAT,LNG(I))
             IF(ISTAT.EQ.1)THEN
                          CS(2,KG2)=' '
```

```
                           CS(2,KG2)(:5)='ERROR'
                           IX(KG2)=0
              ELSE
                      CS(2,KG2)=CS(1,KG2)
              ENDIF
         I1=I1+2
         ENDIF
         IF(NOCCUR.EQ.2)THEN
         IGS1=1
         IGE1=IND(1)-1
         NUMIN=' '
         NUMIN=DOC1(J)(IGS1:IGE1)
         CALL READNUM(NUMIN,RNUM,KG1,KFLG)
         IN1P1=IND(1)+1
         IN2M1=IND(2)-1
                 IND1=IND(2)+1
       IF(LEN.EQ.IND1)E(KG1)=ICHAR(DOC1(J)(IND1:IND1))-48
       IF(LEN.EQ.IND(2)+2)THEN
                 IND2=IND(2)+2
            E(KG1)=10*(ICHAR(DOC1(J)(IND1:IND1))-48)+ICHAR(DOC1(J)
      +(IND2:IND2))-48
       ENDIF
         EP(KG1)=E(KG1)
       LNG(KG1)=IND(2)-IND(1)-1
       CS(1,KG1)(:LNG(KG1))=DOC1(J)(IN1P1:IN2M1)
       CALL FINDGR(AS,NGRPT,CS,IX,KG1,ISTAT,LNG(I))
              IF(ISTAT.EQ.1)THEN
                           CS(2,KG1)=' '
                           CS(2,KG1)(:5)='ERROR'
                           IX(KG1)=0
              ELSE
                      CS(2,KG1)=CS(1,KG1)
              ENDIF
        I1=I1+1
        ENDIF
222     IF(I1.LT.NG)GOTO 223
3001    CONTINUE
        J=J+1
      CALL LRDCHAR(DOC1(J))
      CALL DBLNK(DOC1(J),LEN,0)
             IF(DOC1(J)(:8).EQ.'NROTORS:')THEN
                           NUL=' '
                           NUL(:LEN-8)=DOC1(J)(9:LEN)
                           CALL READNUM(NUL,RNUM,NROT,KFLG)
                           IF(KFLG.NE.0)THEN
                                        NROT=0
                           ENDIF
                           NROTS=NROT
                           GOTO 3001
             ENDIF
      IF(DOC1(J)(:10).EQ.'ENDSPECIES')THEN
         IF(IDONE.NE.2)SYMM=1
           GOTO 6000
      ENDIF
```

```
        IF(DOC1(J)(:6).EQ.'NOCALC'.AND.(KDOC.EQ.0.OR.KDOC.EQ.2))
     +THEN
               DOCTYPE='NOCALC'
               GOTO 3001
        ENDIF
        IF(DOCTYPE.EQ.'NOCALC'.AND.IDAT.EQ.0)THEN
               IF(DOC1(J)(:2).EQ.'HF'.OR.DOC1(J)(:2).EQ.'Hf')THEN
               J=J+1
               CALL DBLNK(DOC1(J),LEN,1)
               IF(LEN.EQ.0)THEN
               WRITE(*,*)'ERROR CONVERTING THERMO'
               WRITE(*,*)'DOCTYPE RESET TO "CALC"'
               WRITE(*,*)'IF # GROUPS > 0 , ELSE ABANDON'
               WRITE(*,*)          'HIT RETURN TO CONTINUE'
               READ(*,3)NUL
               IF(NG.EQ.0)THEN
                       ISTAT=2
                       FLG=2
                       RETURN
               ENDIF
               DOCTYPE=' '
               GOTO 3001
               ENDIF
               CALL INDXST(DOC1(J),' ',NOCCUR,IND,LEN,1,FLG)
               NUL=DOC1(J)(:IND(1)-1)
               CALL READNUM(NUL,B(1,1),NUM,KFLG)
               DO 298 K8=1,NOCCUR-1
               NUL=DOC1(J)(IND(K8)+1:IND(K8+1)-1)
               CALL READNUM(NUL,B(1,K8+1),NUM,KFLG)
298            CONTINUE
               NUL=DOC1(J)(IND(NOCCUR)+1:LEN)
               CALL READNUM(NUL,B(1,NOCCUR+1),NUM,KFLG)
               IDAT=1
                                      if(funits.eq.'KJ')then
                                                   do 579 ik1=1,noccur+1
579                                                b(1,ik1)=b(1,ik1)/4.184
                                      endif
             IF(IENTER.EQ.10)THEN
                         DO 1957 IIU=1,NG
                         IF(CS(1,I)(:2).EQ.'OI')THEN
                                      B(1,2)=B(1,2)-A(IX(IIU),2)
                         ENDIF
1957                     CONTINUE
             ENDIF
               GOTO 3001
               ENDIF
        ENDIF
               IF(DOC1(J)(:9).EQ.'DELTAEDIT')THEN
               J=J+1
               CALL DBLNK(DOC1(J),LEN,1)
               IF(LEN.EQ.0)THEN
               WRITE(*,*)'ERROR CONVERTING THERMO'
               WRITE(*,*)'DELTA EDIT GROUP IGNORED'
               WRITE(*,*)          'HIT RETURN TO CONTINUE'
```

```
                READ(*,3)NUL
                GOTO 3001
                ENDIF
                CALL INDXST(DOC1(J),' ',NOCCUR,IND,LEN,1,FLG)
                NUL=DOC1(J)(:IND(1)-1)
                CALL READNUM(NUL,DELTA(1),NUM,KFLG)
                DO 598 K8=1,NOCCUR-1
                NUL=DOC1(J)(IND(K8)+1:IND(K8+1)-1)
                CALL READNUM(NUL,DELTA(K8+1),NUM,KFLG)
598             CONTINUE
                NUL=DOC1(J)(IND(NOCCUR)+1:LEN)
                CALL READNUM(NUL,DELTA(NOCCUR+1),NUM,KFLG)
                           if(funits.eq.'KJ')then
                                        do 578 ik1=1,noccur+1
578                                     delta(ik1)=delta(ik1)/4.184
                           endif
                GOTO 3001
                ENDIF
        IF(DOC1(J)(:4).EQ.'BOND'.AND.TYPE.EQ.'R')THEN
                NUMIN=' '
                NUMIN=DOC1(J)(5:LEN)
                CALL READNUM(NUMIN,BD,NUM,KFLG)
                           if(funits.eq.'KJ')bd=bd/4.184
                GOTO 3001
        ENDIF
        IF(DOC1(J)(:16).EQ.'DELTABONDENERGY=')THEN
                NUMIN=DOC1(J)(17:LEN)
                CALL READNUM(NUMIN,DELB,NUM,KFLG)
            IF(DOCTYPE.NE.'NOCALC')B(1,1)=B(1,1)+DELB
             if(funits.eq.'KJ')delb=delb/4.184
                GOTO 3001
        ENDIF
             IF(DOC1(J)(:4).EQ.'TEXT')THEN
                           NUMIN=DOC1(J)(5:LEN)
                           CALL READNUM(NUMIN,RNUM,KOUNT,KFLG)
                           DO 634 IJ=1,KOUNT
                                        J=J+1
                                        IF(DOC1(J)(:10).EQ.'ENDSPECIES')GOTO 635
                                        TEXT(IJ)=DOC1(J)
634                        CONTINUE
                           GOTO 3001
             ENDIF
635      IF(DOC1(J)(:10).EQ.'ENDSPECIES')THEN
         IF(IDONE.NE.2)SYMM=1
           GOTO 6000
      ENDIF
      IF(DOC1(J)(:8).NE.'SYMMETRY')GOTO 3001
        IDONE=2
      IF(LEN.EQ.9)SYMM=ICHAR(DOC1(J)(9:9))-48
      IF(LEN.EQ.10)SYMM=10*(ICHAR(DOC1(J)(9:9))-48)+ICHAR(DOC1(J)(10:
     +10))-48
      IF(LEN.EQ.11)SYMM=100*(ICHAR(DOC1(J)(9:9))-48)+10*(ICHAR(DOC1(J)
     +(10:10))-48)+ICHAR(DOC1(J)(LEN:LEN))-48
        GOTO 3001
```

```
6000    CONTINUE
        IF(TYPE.NE.'R')  SYMS=SYMM
        IF(TYPE.EQ.'R')  SYMR=SYMM
      SCOR=1.987*LOG(SYMM*1.0)
             IF(IENTER.EQ.10.AND.DOCTYPE.EQ.'NOCALC')THEN
                          B(1,2)=B(1,2)+SCOR
             ENDIF
        IF(DOCTYPE.EQ.'NOCALC'.AND.IDAT.EQ.1)THEN
                ISTAT=1
                RETURN
        ENDIF
************************************************************************
******           CALCULATE SPECIES PROPERTIES                    ******
************************************************************************
             NTEMPS=7
      DO 3100 I2=1,NG
             IF(IX(I2).NE.0)THEN
             IF(A(IX(I2),9).EQ.0.0.AND.A(IX(I2),8).NE.0.0)NTEMPS=6
      DO 850 Y=1,9
             IF(IENTER.EQ.10.AND.AS(IX(I2))(:2).EQ.'OI')GOTO 3100
      B(1,Y)=E(I2)*A(IX(I2),Y)+B(1,Y)
        IF(IX(I2).LT.IFIRST(4).OR.IX(I2).GT.ILAST(4))THEN
      B(2,Y)=E(I2)*A(IX(I2),Y)+B(2,Y)
        ENDIF
850     CONTINUE
        IF(IX(I2).LT.IFIRST(4).OR.IX(I2).GT.ILAST(4))GOTO 3100
        B(1,1)=B(1,1)-52.1
             ENDIF
3100  CONTINUE
             IF(NTEMPS.EQ.6)THEN
                          B(1,9)=0.0
                          B(2,9)=0.0
             ENDIF
             DO 568 Y=1,9
             IF(DELTA(Y).NE.0)THEN
                          B(1,Y)=B(1,Y)+DELTA(Y)
                          B(2,Y)=B(2,Y)+DELTA(Y)
             ENDIF
568          CONTINUE
             IF(IENTER.NE.10)THEN
      B(1,2)=B(1,2)-SCOR
        SCORP=1.987*LOG(SYMS*1.0)
        IF(TYPE.NE.'R')SCORP=SCOR
             ENDIF
        ISTAT=1
          IF(ISTAT.EQ.2.AND.FLG.EQ.2)GOTO 90000
          IF(FLG.EQ.1)RETURN
          IF(IDOC(J1).GT.NDOC)GOTO 95000
      FLG=0
      RETURN
***********************************************************************
90000 CALL CLS
      WRITE(*,*)'            ERROR : SPECIES NOT FOUND'
      CALL LINES (0,2)
      WRITE(*,*)'                { hit return to continue }'
      READ(*,3)NUL
1492      FLG=1
        CLOSE(U,STATUS='KEEP')
        RETURN
95000   FLG=1
        WRITE(*,*)'   J1 > NSPEC'
        WRITE(*,*)' J1 = ',J1
        WRITE(*,*)' NSPEC= ',NSPEC
      WRITE(*,*)'             ERROR : SPECIES NOT FOUND'
      CALL LINES (0,2)
      WRITE(*,*)'                { hit return to continue }'
      READ(*,3)NUL
      RETURN
      END
      SUBROUTINE CONFIG(ELEM,NFILES,KS,ISTAT,OUTPUT)
      CHARACTER*70 NUL,DUMMY,KS(1),HLP(10),ERRORMSG
      CHARACTER*14 NUMIN,OUTPUT,UNITS,TRANGE
      CHARACTER*2 ELEM(1),DRIVE1,DRIVE2,DRIVE3,ERRORCK
      INTEGER I,LEN1,NELEM,NFILES,ISTAT,I1CH(50),I2CH(35)
      COMMON/CONFG2/NELEM,N1CH,N2CH,I1CH,I2CH
      COMMON/CONFG3/HLP
      COMMON/CONFG4/DRIVE1,DRIVE2,DRIVE3
      COMMON/CONFG5/NKEYS
      COMMON/CONFG6/UNITS
      COMMON/CONFG7/ERRORCK
      COMMON/CONFG8/TRANGE
      TRANGE='2000'
      UNITS='KCAL'
      OUTPUT='QUICK'
      DRIVE1=' '
      DRIVE2=' '
      DRIVE3=' '
      NKEYS=3
      ERRORCK=' '
2     FORMAT(A2)
14    FORMAT(A14)
70    FORMAT(A70)
      ISTAT=0
      OPEN(96,FILE='THERM.CFG',STATUS='OLD',ERR=9999)
      ERRORMSG=' KEYWORD " #FILES " MISSING OR NOT FIRST LINE'
      READ(96,70,END=9999)NUL
      CALL LRDCHAR(NUL)
      CALL DEBLNK(NUL,DUMMY,LEN1,0)
      IF(NUL(:6).NE.'#FILES')GOTO 9999
      NUMIN=NUL(7:LEN1)
      CALL READNUM(NUMIN,RNUM,NFILES,KFLG)
      ERRORMSG=' FEWER FILES FOUND THAN SPECIFIED IN " #FILES " '
      DO 10 I=1,NFILES
      READ(96,70,END=9999)KS(I)
      CALL DBLNK(KS(I),LEN1,0)
      IF(KS(I).EQ.'#ELEMENTS')GOTO 9999
10    CONTINUE
      ERRORMSG=' KEYWORD " #ELEMENTS " IS MISSING'
      READ(96,70,END=9999)NUL
      CALL LRDCHAR(NUL)
      CALL DEBLNK(NUL,DUMMY,LEN1,0)
      IF(NUL(:9).NE.'#ELEMENTS')GOTO 9999
      NUMIN=NUL(10:LEN1)
      CALL READNUM(NUMIN,RNUM,NELEM,KFLG)
      N1CH=0
      N2CH=0
      ERRORMSG=' FEWER ELEMENTS FOUND THAN SPECIIED IN " #ELEMENTS"'
      DO 20 I=1,NELEM
      READ(96,70,END=9999)NUL
      CALL DEBLNK(NUL,DUMMY,LEN1,0)
      ELEM(I)=NUL(:LEN1)
      IF(LEN1.EQ.1)THEN
              N1CH=N1CH+1
```

```
                           I1CH(N1CH)=I
                           GOTO 20
             ENDIF
             IF(LEN1.EQ.2)THEN
                           N2CH=N2CH+1
                           I2CH(N2CH)=I
             ENDIF
             IF(LEN1.GT.2)THEN
                           GOTO 9999
             ENDIF
20           CONTINUE
             ERRORMSG=' KEYWORD " #HELP " MISSING'
             READ(96,70,END=9999)NUL
             CALL LRDCHAR(NUL)
             CALL DEBLNK(NUL,DUMMY,LEN1,0)
             IF(NUL(:5).NE.'#HELP')GOTO 9999
             NUMIN=NUL(6:LEN1)
             CALL READNUM(NUMIN,RNUM,NHELP,KFLG)
             ERRORMSG=' FEWER HELP FILES FOUND THAN SPECIFIED IN " #HELP "'
             DO 60 I=1,NHELP
             READ(96,70,END=9999)HLP(I)
             CALL DBLNK(HLP(I),LEN1,0)
             IF(HLP(I)(:6).EQ.'OPTION')GOTO 9999
60      CONTINUE

*
*    OPTIONS ANY ORDER IS ACCEPTABLE
*
*     DEFAULTS ARE DEFINED: ONLY NECESSARY IF WANT TO SPECIFY
*      OPTION DIFFERENT FROM DEFAULT
*
10000        CONTINUE
             READ(96,70,END=9777)NUL
             CALL DEBLNK(NUL,DUMMY,LEN1,0)
             IF(NUL(:12).EQ.'THERMFIT.EXE')THEN
                           IF(NUL(13:14).NE.' ')DRIVE1=NUL(13:14)
             ELSE IF(NUL(:12).EQ.'THERMLST.EXE')THEN
                           IF(NUL(13:14).NE.' ')DRIVE2=NUL(13:14)
             ELSE IF(NUL(:12).EQ.'THERMRXN.EXE')THEN
                           IF(NUL(13:14).NE.' ')DRIVE3=NUL(13:14)
             ELSEIF(NUL(:5).EQ.'#KEYS')THEN
                           NUMIN=NUL(6:LEN1)
                           CALL READNUM(NUMIN,RNUM,NKEYS,KFLG)
             ELSEIF(NUL(:11).EQ.'ERRORCK=OFF')THEN
                           ERRORCK='NO'
                   ELSEIF(NUL(:9).EQ.'TRANGE=EX'.OR.NUL(:8).EQ.'TRANGE=5')THEN
                           TRANGE='5000'
             ELSEIF(NUL(:6).EQ.'UNITS:')THEN
                           IF(NUL(7:8).EQ.'KJ')UNITS='KJ'
             ELSEIF(NUL(:7).EQ.'OUTPUT:')THEN
                           IF(NUL(8:LEN1).EQ.'CLOSE')OUTPUT='CLOSE'
             ENDIF
             GOTO 10000
9777         CLOSE(96,STATUS='KEEP')
```

```
           ISTAT=0
           RETURN
9999       ISTAT=1
           CALL LINES(0,1)
           WRITE(*,'(1X,A70)')ERRORMSG
           CALL LINES(0,1)
           WRITE(*,*)'   ERROR ENCOUNTERED IN THERM.CFG'
           WRITE(*,*)'   REQUIRED CONFIG DATA IS MISSING'
           CLOSE(96,STATUS='KEEP')
           RETURN
           END
```

```
        SUBROUTINE HELP(FNAME,OPTION)
        CHARACTER*70 FNAME
        CHARACTER*70 NUL,NUMIN,S2,OPTION
        LOGICAL EXST
1       CONTINUE
        INQUIRE(FILE=FNAME,EXIST=EXST,ERR=9900)
70      FORMAT(A70)
71      FORMAT(2X,A70)
        IF(EXST)THEN
                CALL DEBLNK(OPTION,S2,LEN2,0)
                OPEN(69,FILE=FNAME,STATUS='OLD')
                IF(OPTION(:LEN2).EQ.'A')GOTO 200
                I=0
100             I=I+1
                READ(69,70,END=9900)NUL
                IF(NUL(:1).NE.'O'.and.nul(:2).ne.' O')GOTO 100
                CALL DEBLNK(NUL,S2,LEN1,0)
                IF(NUL(7:LEN1).NE.OPTION(:LEN2))GOTO 100
222             CALL CLS
                J=0
200             J=J+1
                IF(J.GE.22)THEN
                        WRITE(*,*)' { hit return to continue }'
                        READ(*,70)S2
                        GOTO 222
                ENDIF
                READ(69,70,END=9900)NUL
                IF(NUL(:3).NE.'END'.AND.NUL(:3).NE.'end'.AND.NUL(:4)
     +.NE.' END'.AND.NUL(:4).NE.' end')THEN
                        WRITE(*,70)NUL
                        GOTO 200
                ENDIF
                WRITE(*,*)' { hit return to continue }'
                READ(*,70)S2
                IF(OPTION(:LEN2).NE.'A')THEN
                        CLOSE(69,STATUS='KEEP')
                        RETURN
                ENDIF
                GOTO 222
        ENDIF
        CALL CLS
        WRITE(*,*)'  CAN''T FIND HELP FILE'
        WRITE(*,71)FNAME
        WRITE(*,*)'   HELP FILES MUST RESIDE IN THE CURRENT DIRECTORY !'
        CALL LINES(0,2)
        WRITE(*,*)'   HIT RETURN TO ENTER A NEW DRIVE/DIRECTORY SPEC'
        WRITE(*,*)'   OR ENTER "Q" TO ABORT HELP'
        READ(*,70)NUL
        IF(NUL.EQ.' ')THEN
                PAUSE 'ENTER DOS COMMAND TO CHANGE DIRECTORY/DISK'
                GOTO 1
        ENDIF
        CALL LRDCHAR(NUL)
        IF(NUL.EQ.'Q')RETURN
```

```
9900          CALL CLS
              WRITE(*,*)' NO FURTHER HELP AVAILABLE'
              WRITE(*,*)' { hit return to continue }'
              READ(*,70)S2
              CLOSE(69,STATUS='KEEP')
              RETURN
              END
```

```
            SUBROUTINE FNAME (S1,S2)
*
*
*           S1  =  CHARACTER*(*) INPUT FILENAME
*       S3  =  CHARACTER*(*)  FILE EXTENSION
*
*
            CHARACTER*(*) S1
            CHARACTER*(*) S2
            INTEGER IND(70),FLG,LEN
            CALL READCHAR(S1)
            CALL DBLNK(S1,LEN,0)
            CALL INDXST(S1,'.',NOC,IND,LEN,1,FLG)
            IF(NOC.EQ.0)THEN
                      S1(LEN+1:LEN+1)='.'
                      S1(LEN+2:LEN+4)=S2
                      S1(LEN+5:70)=' '
            ELSE IF(NOC.NE.0)THEN
                      S1(IND(1)+1:IND(1)+3)=S2
                      S1(IND(1)+4:70)=' '
            ENDIF
            RETURN
            END
```

```
              SUBROUTINE FINDSP(S1,DOC,NSPEC,IDOC,IRECN,ISTAT)
$LARGE
              CHARACTER*14 S1,DOC(1)
              CHARACTER*70 NUL, DUMMY
              INTEGER NSPEC,IRECN,IDOC(1),ISTAT
70            FORMAT(A70)
14            FORMAT(A14)
              ISTAT=0
              DO 100 I=1,NSPEC
              DUMMY=' '
              DUMMY(:14)=DOC(I)(:14)
              CALL DEBLNK(DUMMY,NUL,NUM,0)
              DOC(I)(:14)=DUMMY(:14)
              IF(S1(:14).EQ.DOC(I)(:14))THEN
                            IRECN=I
                            RETURN
              ENDIF
100           CONTINUE
              ISTAT=1
              WRITE(*,*)'SPECIES ',S1(:14),' NOT FOUND'
              READ(*,14)S2
              RETURN
              END
```

```
            SUBROUTINE INQNUM(S1,IANS,NUMLEN,LEN,IBL)
**********************************************************************
*
*                       written by Ed Ritter : 11/20/88                                 *
*
**********************************************************************
*
*           THIS SUBROUTINE SCANS STRING S1 FOR A LEADING NUMBER      *
*      The search is carried out from left to right.            *
*      The first non-numeric character encountered ends         *
*       the numeric substring.                                                          *
*
*           return codes :IANS                                                          *
*
*           IANS = 0  : the string is non-numeric                           *
*                = 1  : string S1 is an integer                             *
*                = 2  : S1 is a decimal                                      *
*                = 3  : S1 has a leading integer of length NUMLEN       *
*                = 4  : S1 has a leading decimal of length NUMLEN.      *
*
*      NUMLEN    : the length of the leading numeric portion of S1 *
*      LEN       : the length of the S1 without blanks              *
*      IBL       : allows the calling routine to control blank      *
*                  removal in subroutine DEBLNK                     *
*                  IBL=0 : leave no blanks in S1                    *
*                            IBL=1 : leave one remaining blank in each blank *
*                                   field                                       *
*
**********************************************************************
$large
            CHARACTER*(*) S1
            INTEGER IANS,NUMLEN,LEN,ICTRL,IDEC,IBL
*
*  ICTRL <> 0, when a non-numeric character has been encountered in S1.
*  IDEC <> 0, if a decimal point is encountered in a numeric field;
*            in this case IDEC = address of decimal point.
10          continue
            NUMLEN=0
            ICTRL=0
            IDEC=0
            IANS=0
            CALL DBLNK (S1,LEN,IBL)
            DO 100 I=1,LEN
            IF((ICHAR(S1(I:I)).GE.48.AND.ICHAR(S1(I:I)).LE.57).OR.
    +(S1(I:I).EQ.'.'.AND.IDEC.EQ.0))THEN
                        IF(ICTRL.EQ.0)THEN
                                   NUMLEN=NUMLEN+1
                                   IF(S1(I:I).EQ.'.')IDEC=I
                        ENDIF
            ELSE
                        ICTRL=1
            ENDIF
100         CONTINUE
            IF(NUMLEN.EQ.0)THEN
```

```
                 IANS=0
        ELSEIF(NUMLEN.EQ.LEN)THEN
                 IF(IDEC.EQ.0)THEN
                          IANS=1
                 ELSEIF(IDEC.NE.0)THEN
                          IANS=2
                 ENDIF
        ELSEIF(NUMLEN.LT.LEN)THEN
                 IF(IDEC.EQ.0)THEN
                          IANS=3
                 ELSEIF(IDEC.NE.0)THEN
                          IANS=4
                 ENDIF
        ENDIF
99999   CONTINUE
        RETURN
        END
```

```
      SUBROUTINE TBREAK(TMID,TINIT)
$LARGE
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION T(30),F(14)
      COMMON/TEMPFIT/T,F
      ITER=0
1     CONTINUE
      ITER=ITER+1
      TNEXT=TINIT-(FDIF(TINIT)/DFDIF(TINIT))
      IF(ABS(TNEXT-TINIT).GT.1.0)THEN
                TINIT=TNEXT
                IF(ITER.LT.500)GOTO 1
      ENDIF
      TMID=TNEXT
      RETURN
      END
```

```
      FUNCTION FDIF(X)
$LARGE
*
*   THIS FUNCTION RETURNS THE SUM OF THE SQUARE OF THE DIFFERENCE
*     BETWEEN VALUES OF TWO POLYNOMIALS AND THEIR DERIVATIVE AT A SPECIFIC
*     TEMPERATURE (X).
*
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION T(30),F(14)
      COMMON/TEMPFIT/T,F
      AX=(CPT(0,X,F)-CPT(7,X,F))
      BX=(DCPT(0,X,F)-DCPT(7,X,F))
      FDIF=(AX**2)+(BX**2)
      RETURN
      END
```

```
      FUNCTION DFDIF(X)
$LARGE
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION T(30),F(14)
      COMMON/TEMPFIT/T,F
      AX=(CPT(0,X,F)-CPT(7,X,F))
      BX=(DCPT(0,X,F)-DCPT(7,X,F))
      TERM1=(F(2)-F(9))+(2.0*X*(F(3)-F(10)))+(3.0*(X**2)*(F(4)-F(11)))
     + +(4.0*(X**3)*(F(5)-F(12)))
      TERM2=(2.*(F(3)-F(10)))+(6.*X*(F(4)-F(11)))
     + +(12.*(X**2)*(F(5)-F(12)))
      DFDIF=(2.*AX*TERM1)+(2.*BX*TERM2)
      RETURN
      END
```

```
          SUBROUTINE CSTRING(S,N)
$LARGE
          CHARACTER*(*) S(*)
100       CONTINUE
          DO 200 I=1,N
          IF(S(I).EQ.' ')THEN
                    DO 300 I1=I+1,N
300                 S(I1-1)=S(I1)
                    N=N-1
                    GOTO 100
          ENDIF
200       CONTINUE
          RETURN
          END
```

```
          SUBROUTINE DCREAL(X,N)
$LARGE
          IMPLICIT REAL*8(A-H,O-Z)
          DIMENSION X(*)
100       CONTINUE
          DO 200 I=1,N
          IF(X(I).EQ.0.0)THEN
                    DO 300 I1=I+1,N
300                 X(I1-1)=X(I1)
                    N=N-1
                    GOTO 100
          ENDIF
200       CONTINUE
          RETURN
          END
```

```
          SUBROUTINE DREADNUM(S1,DNUM,IFLG)
$LARGE
          CHARACTER*(*)S1,S2*20
          REAL*8 DNUM,COEF,EXPONENT
          IFLG=0
          call slen(s1,lstr)
          CALL INQNUM(S1,IANS,NLEN,LSTR,0)
          IF(IANS.EQ.0)THEN
                    IFLG=-10
                    RETURN
          ELSEIF(IANS.EQ.1.AND.LSTR.GT.10)THEN
                    IFLG=-5
                    DNUM=0.
                    RETURN
          ELSEIF(IANS.EQ.2.AND.LSTR.GT.11)THEN
```

```
                IFLG=-6
                DNUM=0.0
                RETURN
        ELSEIF(IANS.EQ.1.OR.IANS.EQ.2)THEN
                CALL READNUM(S1,RNUM,num,IFLG)
                IF(IFLG.NE.0)THEN
                        RETURN
                ELSE
                        DNUM=DBLE(RNUM)
                        RETURN
                ENDIF
        ELSE
                S2=' '
                S2(:NLEN)=S1(:NLEN)
                CALL READNUM(S2,RNUM,num,IFLG)
                IF(IFLG.NE.0)RETURN
                COEF=DBLE(RNUM)
                S2=' '
                LSTR1=LSTR-(NLEN+1)
                S2(:LSTR1)=S1(NLEN+2:LSTR)
                IF(S2(:1).EQ.'+')THEN
                        S2(:1)=' '
                ENDIF
                CALL INQNUM(S2,IANS2,LEXP,LLSTR,0)
                IF(IANS2.NE.1.AND.S2(:1).NE.'-')THEN
                        IFLG=-20
                        RETURN
                ELSE
                        CALL READNUM(S2,RNUM,num,IFLG)
                        IF(IFLG.NE.0)RETURN
                        IF(NUM.GT.130.OR.NUM.LT.-130)THEN
                                IFLG=-30
                                RETURN
                        ENDIF
                        EXPONENT=10.0**NUM
                ENDIF
                DNUM=COEF*EXPONENT
        ENDIF
        RETURN
        END
```

```
      SUBROUTINE LINLSQ (X,Y,B,NRXNS,NEXP,NSPC,NRT,SSR,AVGERR)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION XT(15,31,1),X(31,15,1),XTX(15,15,1),BP(15,1,1)
      DIMENSION XINV(15,15,1),WORK(15,15,1),DELPCT(100)
      DIMENSION Y(31,25,5),B(16,25,5),XTY(15,1,1),YP(31,1,1)
C   XO IS THE INTERCEPT: ie.; DIMENSION = 1 + # OF VARIABLES
      NRXN=NRXNS+1
C   SET THE VECTOR X(1,N,1)=1.0
      DO 99 K=1,NEXP
   99 X(K,1,1)=1.0
C   DEFINE THE TRANSPOSE OF X : ie.; XT
      DO 2000 J=1,NRXN
      DO 1000 I=1,NEXP
 1000 XT(J,I,1)=X(I,J,1)
 2000 CONTINUE
C   COMPUTE THE MATRIX PRODUCT XTX
      CALL PMATRIX(XT,X,XTX,NRXN,NEXP,NRXN,1)
C   COMPUTE THE INVERSE OF XTX
      CALL INVERT(NRXN,XTX,XINV,WORK,1)
C   DEFINE THE VECTOR YP(I,1,1): THESE ARE THE FUNCTION VALUES
      DO 335 I=1,NEXP
  335 YP(I,1,1)=Y(I,NRT,NSPC)
C   COMPUTE THE MATRIX VECTOR PRODUCT XTY
      CALL PMATRIX(XT,YP,XTY,NRXN,NEXP,1,1)
C   COMPUTE THE VECTOR OF LINEAR COEFFICIENTS
      CALL PMATRIX(XINV,XTY,BP,NRXN,NRXN,1,1)
      DO 336 I=1,NRXN
  336 B(I,NRT,NSPC)=BP(I,1,1)
      SSR=0.
      DELSUM=0.
      AVGERR=0.
C   CALCULATE PREDICTED RESPONSE, AVG ABS ERROR, AND SUM OF SQUARES
C      WRITE(9,*)'LINEAR PREDICTION SUMMARY'
C      WRITE(9,*)'             Y              YCALC               %DELTA'
C      DO 444 J=1,NEXP
      YCALC=0.
      DO 333 I=1,NRXN
  333 YCALC=YCALC+(B(I,NRT,NSPC)*X(J,I,1))
      DEL= Y(J,NRT,NSPC)-YCALC
      IF(Y(J,NRT,NSPC).EQ.0.0)GO TO 334
      DELPCT(J)=(DEL/Y(J,NRT,NSPC))*100.
C      WRITE(9,*)Y(J,NRT,NSPC),YCALC,DELPCT(J)
  334 DELSUM=DELSUM+DABS(DELPCT(J))
  444 SSR=SSR+(DEL**2)
      AVGERR=DELSUM/NEXP
      RETURN
      END
      SUBROUTINE PMATRIX (A,B,C,I,J,K,N1)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(15,31,1),B(31,15,1),C(15,15,1)
      DO 103 M=1,I
      DO 102 N=1,K
      TSUM=0.
      DO 101 L=1,J
  101  TSUM=TSUM+(A(M,L,1)*B(L,N,N1))
  102  C(M,N,1)=TSUM
  103  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INVERT(NCON,C,CIN,B,N)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION C(15,15,1),CIN(15,15,1),B(16,15,1)
C  CREATE THE IDENTITY MATRIX B
      DO 10 I=1,NCON
      DO 20 J=1,NCON
      B(I,J,1)=0.0
      IF(J.EQ.I) B(I,J,1)=1.0
   20 CONTINUE
   10 CONTINUE
      NR=NCON
C      WRITE(7,*)'IDENTITY MATRIX ='
C      DO 30 I=1,NCON
C      WRITE(7,*)(B(I,J,1),J=1,NCON)
C   30 CONTINUE
      CALL GJELIM(NCON,NR,N,C,B,CIN)
      RETURN
      END
      SUBROUTINE GJELIM(NCON,NR,NA,C,B,CIN)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION C(15,15,1),B(16,15,1),CIN(15,15,1)
      IF(NCON.EQ.1) GO TO 1120
      LAST=NCON-1
      DO 1700 I=1,LAST
      M=I
      ITEMP=I+1
      DO 1100 J=ITEMP,NCON
      IF(ABS(C(M,I,NA)).GT.ABS(C(J,I,NA))) GO TO 1100
      M=J
 1100 CONTINUE
      IF(ABS(C(M,I,NA)).LT.1.E-15) GO TO 1130
      IF(M.EQ.I) GO TO 1400
      DO 1200 J=I,NCON
      TEMP=C(M,J,NA)
      C(M,J,NA)=C(I,J,NA)
 1200 C(I,J,NA)=TEMP
      DO 1300 J=1,NR
      TEMP=B(M,J,NA)
      B(M,J,NA)=B(I,J,NA)
 1300 B(I,J,NA)=TEMP
 1400 DO 1600 J=ITEMP,NCON
      TEMP=C(J,I,NA)/C(I,I,NA)
      C(J,I,NA)=0.0
      DO 1500 K=ITEMP,NCON
 1500 C(J,K,NA)=C(J,K,NA)-TEMP*C(I,K,NA)
      DO 1600 K=1,NR
 1600 B(J,K,NA)=B(J,K,NA)-TEMP*B(I,K,NA)
 1700 CONTINUE
      IF(ABS(C(NCON,NCON,NA)).LT.1.E-15) GO TO 1130
      DO 1800 J=1,NR
 1800 CIN(NCON,J,NA)=B(NCON,J,NA)/C(NCON,NCON,NA)
      K=NCON-1
 1900 DO 1111 J=1,NR
      IT=K+1
      TEMP=0.0
      DO 1110 I=IT,NCON
 1110 TEMP=TEMP+C(K,I,NA)*CIN(I,J,NA)
 1111 CIN(K,J,NA)=(B(K,J,NA)-TEMP)/C(K,K,NA)
      K=K-1
      IF(K.NE.0) GO TO 1900
      RETURN
 1120 CIN(NCON,NR,NA)=B(NCON,NR,NA)/C(NCON,NCON,NA)
      RETURN
 1130 CONTINUE
      WRITE(*,120)
  120 FORMAT(//10X,'*** MATRIX IS SINGULAR  ***'   )
      WRITE(*,*)'ERROR IN SUBROUTINE GJELIM: MATRIX FOLLOWS'
      DO 1066 I=1,NCON
      WRITE(*,*)(C(I,J,NA),J=1,NCON)
 1066 CONTINUE
      STOP
      END
```

```
             FUNCTION HF(I,T,F)
$LARGE
             IMPLICIT REAL*8 (A-H,O-Z)
             DIMENSION F(*)
             HF = F(1+I)+(F(6+I)/T)+((F(2+I)/2.)*T)+((F(3+I)/3.)*(T**2))
     + +((F(4+I)/4.)*(T**3))+((F(5+I)/5.)*(T**4))
             RETURN
             END

             FUNCTION DHF(I,T,F)
$LARGE
             IMPLICIT REAL*8 (A-H,O-Z)
             DIMENSION F(*)
             DHF=(F(2+I)/2.)+(-1.*F(6+I)/(T**2))+((2./3.)*F(3+I)*T)+
     + ((3./4.)*F(4+I)*(T**2))+((4./5.)*F(5+I)*(T**3))
             RETURN
             END


             FUNCTION CPT(I,T,F)
$LARGE
             IMPLICIT REAL*8 (A-H,O-Z)
             DIMENSION F(*)
             CPT=F(1+I)+(F(2+I)*T)+(F(3+I)*(T**2))+(F(4+I)*(T**3))+
     +  (F(5+I)*(T**4))
             RETURN
             END

             FUNCTION DCPT(I,T,F)
$LARGE
             IMPLICIT REAL*8 (A-H,O-Z)
             DIMENSION F(*)
             DCPT=F(2+I)+(2.*F(3+I)*T)+(3.*F(4+I)*(T**2))+(4.*F(5+I)*(T**3))
             RETURN
             END

             FUNCTION ST(I,T,F)
$LARGE
             IMPLICIT REAL*8 (A-H,O-Z)
             DIMENSION F(*)
             ST=F(7+I)+(F(1+I)*LOG(T))+(F(2+I)*T)+((F(3+I)/2.)*(T**2))+
     +  ((F(4+I)/3.)*(T**3))+((F(5+I)/4.)*(T**4))
             RETURN
             END

             FUNCTION DST(I,T,F)
$LARGE
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION F(*)
      DST=(F(1+I)/T)+F(2+I)+(F(3+I)*T)+(F(4+I)*(T**2))+(F(5+I)*(T**3))
      RETURN
      END
```

```
             FUNCTION F6(I,HOFT,T,F)
$LARGE
             IMPLICIT REAL*8(A-H,O-Z)
             DIMENSION F(*)
      X=HOFT-((T*F(1+I))+((F(2+I)/2.)*(T**2))+((F(3+I)/3.)*
     #(T**3))+((F(4+I)/4.)*(T**4))+((F(5+I)/5.)*(T**5)))
             F6=X
             RETURN
             END




             FUNCTION F7(I,SOFT,T,F)
$LARGE
             IMPLICIT REAL*8(A-H,O-Z)
             REAL*8 T,F7
             DIMENSION F(*)
      X=SOFT-((F(1+I)*DLOG(T))+(F(2+I)*T)+((F(3)/2.)*(T**2))
     #+((F(4+I)/3.)*(T**3))+((F(5+I)/4.)*(T**4)))
             F7=X
             RETURN
             END


             SUBROUTINE CVVIB(FREQ,T,CV,THETA)
$LARGE
             IMPLICIT REAL*8(A-H,O-Z)
             REAL*8 HP,KB,C,X,THETA,TEST
             HP=6.62517E-34
             KB=1.3804E-23
             C=2.998E+10
             IF(FREQ.GT.25000.0.OR.FREQ.LT.1.)THEN
                          CV=1000.
                          RETURN
             ENDIF
             IF(FREQ.NE.0.0)THEN
                          THETA=((HP*C)/KB)*FREQ
             ELSE
                          IF(THETA.EQ.0.)THEN
                                       CV=1000.
                                       RETURN
                          ENDIF
             ENDIF
             X=THETA/T
             IF(X.GT.230.)THEN
                          CV=1000.
                          RETURN
             ENDIF
             TEST=DEXP(X)
             IF(TEST-1.0.LT.10.0E-6)THEN
                          CV=100.
                          RETURN
             ENDIF
```

```
                TOP=EXP(X)
                BOTTOM=(TOP-1.0)**2
                CV=(X**2)*(TOP/BOTTOM)
                RETURN
                END
```

```
                SUBROUTINE SIGMACP(B)
                IMPLICIT REAL*8(A-H,O-Z)
                CHARACTER*70 FILPRN
                DIMENSION B(*),CONST(3)
                COMMON/CONST/CONST,NTERMS
                WRITE(*,*)'ENTER PRN FILENAME'
                READ(*,70)FILPRN
70              FORMAT(A70)
                OPEN(12,FILE=FILPRN,STATUS='UNKNOWN')
                T=100.
                DO 100 I=1,60
                CP=0.
                DO 200 J=1,NTERMS
                THETA=0.0
                IF(NTERMS.EQ.1)THEN
                          CALL CVVIB(B(2),T,CV,THETA)
                ELSE
                          CALL CVVIB(B(NTERMS+J-1),T,CV,THETA)
                ENDIF
                IF(J.NE.NTERMS.OR.NTERMS.EQ.1)THEN
                          TERM=B(J)*CV
                ELSE IF(J.EQ.NTERMS.AND.NTERMS.NE.1)THEN
                          SUM=0.0
                          DO 234 IK=1,NTERMS-1
234                       SUM=SUM+B(IK)
                          TERM=(CONST(2)-SUM)*CV
                ENDIF
                CP=CP+TERM
200             CONTINUE
                CP=(CP+4.0)*1.987*CONST(1)
                WRITE(12,*)T,CP
                T=T+100.
100             CONTINUE
                CLOSE(12,STATUS='KEEP')
                RETURN
                END
```

```
          FUNCTION CPSERIES(B,T)
          IMPLICIT REAL*8(A-H,O-Z)
          DIMENSION B(6),CONST(3)
          COMMON/CONST/CONST,NTERMS
          CP=0.0
          CV=0.0
          DO 100 I=1,3
                     CALL CVVIB(B(I+3),T,CV,THETA)
                     CP=CP+(B(I)*CV)
100       CONTINUE
          IF(CONST(3).EQ.0.)THEN
*                    NON-LINEAR
                     CP=(CP+4.0)
          ELSE
*                    LINEAR
                     CP=(CP+(7./2.))
          ENDIF
          CPSERIES=CP
          RETURN
          END
```

```
      SUBROUTINE FOFX(I,X,B,Y,Z)
$LARGE
             IMPLICIT REAL*8(A-H,O-Z)
      DOUBLE PRECISION B,CONST,XIN,THETA,CV,TERM(3),SUM
      DIMENSION X(*),B(*),CONST(3)
      COMMON /CONST/ CONST,NTERMS
             XIN=X(1)
*            write(*,*)x(1),y
             Z=0.0
             DO 100 II=1,NTERMS
             THETA=0.0
             IF(NTERMS.EQ.1)THEN
                           CALL CVVIB(B(2),XIN,CV,THETA)
             ELSE
                           CALL CVVIB(B(NTERMS+II-1),XIN,CV,THETA)
             ENDIF
             IF(II.NE.NTERMS.OR.NTERMS.EQ.1)THEN
                           TERM(II)=B(II)*CV
             ELSE IF(II.EQ.NTERMS.AND.NTERMS.GT.1)THEN
                           SUM=0.0
                           DO 234 IK=1,NTERMS-1
234                        SUM=SUM+B(IK)
                           TERM(II)=(CONST(2)-SUM)*CV
             ENDIF
             Z=Z+TERM(II)
100          CONTINUE
             IF(CONST(3).EQ.0.)THEN
*                          non-linear molecule
                           Z=(Z+4.0)*1.987*CONST(1)
             ELSE
*                          linear molecule
                           Z=(Z+(7./2.))*1.987*CONST(1)
             ENDIF
      RETURN
      END
```

```
             SUBROUTINE PD(NDATA,X,NPARM,B,Y,ZZ,Z,PDERIV,NORMAL,WORK,POIN01)
$LARGE
             IMPLICIT REAL*8(A-H,O-Z)
             DOUBLE PRECISION NORMAL(7,7),WORK(*),B(*),DEL,POIN01,
     +CONST(3),U,U1,DU,DU1,V,V1,DV,DV1,ALPHA,PRD,TERM1,TERM2,TERM3,
     +TERM4,HP,KB,C
             DIMENSION X(1,20),Y(*),PDERIV(6,20),Z(*)
             INTEGER NDATA,NPARM
             COMMON/CONST/CONST,NTERMS
             HP=6.62517E-34
             KB=1.3804E-23
             C=2.998E+10

C             INITIALIZE NORMAL TO 0.0,  ONLY THE PART OF
C             THE ARRAY THAT WILL BE USED
C
              DO 101 I = 1,NPARM+1
         DO 100 J = I,NPARM+1
         NORMAL(J,I) = 0.0
100          CONTINUE
101          CONTINUE
********************************************************************
*          REPLACE THIS SECTION WITH EXPLICIT JACOBIAN
*
C
C             CALCULATE THE PARTIAL DERIVATIVES OF THE FUNCTION
C             VALUES WITH RESPECT TO THE PARAMETER VALUES FOR
C             EACH SET OF DATA POINTS.  THESE DERIVATIVES WILL BE
C             STORED IN PDERIV.
C
         DO 200 J = 1,NDATA
             ALPHA=(HP*C)/(KB*X(1,J))
             U2=EXP(ALPHA*B(5))
             TERM3=(ALPHA*B(5))**2
             TERM4=(U2/((U2-1.0)**2))
             PRD=TERM3*TERM4
             DO 150 I=1,2
             U1=EXP(ALPHA*B(I+2))
             TERM1=(ALPHA*B(I+2))**2
             TERM2=(U1/((U1-1.0)**2))
             PDERIV(I,J)=(TERM1*TERM2-PRD)*1.987*CONST(1)
150          CONTINUE
             DO 1000 I=3,5
             U=(ALPHA*B(I))**2
             DU=2.0*(ALPHA**2)*B(I)
             U1=EXP(ALPHA*B(I))
             DU1=ALPHA*U1
             V1=(U1-1.0)**2
             DV1=(2.0*(U1-1.0)*DU1)
             V=U1/V1
             DV=((1.0/V1)*DU1)-((U1/(V1**2))*DV1)
             PDERIV(I,J)=(U*DV)+(V*DU)
             IF(I.LT.5)THEN
                          PDERIV(I,J)=PDERIV(I,J)*B(I-2)*1.987*CONST(1)
              ELSE
                           PDERIV(I,J)=PDERIV(I,J)*(CONST(2)-(B(1)+B(2)))*1.987*
     +CONST(1)
              ENDIF
1000          CONTINUE
200           CONTINUE
**********************************************************************
C
C              COMPUTE THE  NORMAL EQUATIONS
C
      DO 240 J1 = 1,NDATA
         DO 230 J2 = 1,NPARM
230         WORK(J2) = PDERIV(J2,J1)
         WORK(NPARM+1) = Y(J1) - Z(J1)
         DO 240 J3 = 1,NPARM+1
            DO 240 J4 = J3,NPARM+1
240            NORMAL(J4,J3) = NORMAL(J4,J3) + WORK(J3) * WORK(J4)
              RETURN
              END
```