EFFECTS OF TEMPERATURE AND HEATING RATE ON OFF-GAS COMPOSITION AND PYRENE REMOVAL FROM AN ARTIFICIALLY-CONTAMINATED SOIL

by

Hiroshi Harlan Saito

B. S., Chemical Engineering and Chemistry
Carnegie-Mellon University, 1989

M. S. Chemical Engineering Practice
Massachusetts Institute of Technology, 1991

Submitted to the Department of Chemical Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February, 1995

© 1995 Massachusetts Institute of Technology. All rights reserved.

Signature of Author .................................................................
Department of Chemical Engineering
February, 1995

Certified by .................................................................
Jack B. Howard
Professor of Chemical Engineering
Thesis Supervisor

.................................................................
William A. Peters
Associate Director, MIT Energy Laboratory
Thesis Supervisor

Accepted by .................................................................
Robert E. Cohen
Professor of Chemical Engineering
Chairman, Committee for Graduate Students
EFFECTS OF TEMPERATURE AND HEATING RATE ON OFF-GAS COMPOSITION AND PYRENE REMOVAL FROM AN ARTIFICIALLY-CONTAMINATED SOIL

by

Hiroshi Harlan Saito

Submitted to the Department of Chemical Engineering on January 19, 1994, in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Chemical Engineering

Abstract

Contaminated soil is a widespread problem in the United States. Cleanup of these sites is necessary due to the rising costs and stricter regulations regarding landfilling of hazardous materials. Soil remediation technology faces the problems of high cost per ton of soil, high volumes of contaminated soil, public concern over emissions from remediation activities, and technical difficulties encountered while remediating sites. As a result, only 52 of 1320 sites on the Environmental Protection Agency's (EPA) National Priority List (or "Superfund" site list) had been completely remediated since the passage of the "Superfund Law" in 1980 (as of October, 1994). Of the Superfund sites where remediation technologies have been selected, tabulated data indicate that polycyclic aromatic hydrocarbons (PAH's) are contaminants in of order 10-15% of sites, with concentrations ranging from hundreds of ppm to several weight percent (average concentration of order 0.5 wt.%).

Incineration/thermal treatment is a promising technology to handle PAH's, which can show the properties of low volatility, limited solubility/extractability, and recalcitrance to microbial degradation. The advantages of thermal techniques, which have been frequently selected by the EPA, are the wide ranges of organic contaminants that can be handled, high destruction and removal efficiencies, and extensive industrial experience with similar systems. Incineration/thermal treatment will continue to be used in the near future since a majority of the contracted work has yet to be completed. However, high cost and low public confidence motivate development of alternative technologies, as well as continual scientific and engineering study of incineration and other thermal methods.

Fundamental understanding of chemical and physical processes by which contaminants and other organic compounds are destroyed, removed, sequestered, and generated during thermal treatment of contaminated soils is needed. Extensive work has been performed by many investigators regarding heat and mass transfer in thermal treatment processes as well as evaporation/thermal desorption of contaminants from soil. However, no literature was found concerning the possibility of secondary chemical reactions and soil decomposition at the elevated temperatures often found in these treatment systems. Hence the focus of this work was to examine the effects of temperature and heating rate on the rates and extents of contaminant removal and destruction, and the identities and generation rates of volatile by-products.
Volatile products yields (tars, CO, CO₂, and C₁-C₂ hydrocarbons), and rates and extents of weight loss as affected by temperature (400 - 1000°C at a 1000°C/s nominal heating rate) and by heating rate (100, 1000, or 5000°C/s to a 550°C nominal final temperature) have been determined for an EPA Synthetic Soil Matrix neat, or "contaminated" with 4.75 wt.% pyrene. Experiments were carried out in a modified electrical screen heater reactor which allows a thin batch sample of soil to be heated through prescribed temperature-time histories under inert or reactive gaseous atmospheres at various pressures. Here, the ambient atmosphere was helium at approximately 3 psig total pressure. Three chemical kinetic models were tested in an effort to capture trends in weight loss and selected products yield data: the multiple independent parallel reaction model (MiPR), and two competitive reaction models.

The results of the experimental and modeling work show conclusively that: (1) temperature strongly affects rates and extents of contaminant removal; (2) temperature and heating rate affect off-gas production from neat and contaminated soils; (3) presence of contaminant can significantly change off-gas compositions, presumably from soil-contaminant chemical reactions; (4) while apparently high extents of pyrene removal (approaches 100%) can be obtained by thermal treatment, thermogravimetric studies can provide misleading results under some circumstances, e.g., from soil-contaminant interactions; and (5) uncontaminated soil itself contributes extensively to volatiles evolved under thermal treatment. This work also suggests that: (1) pyrene removal and off-gas production occur through an elaborate set of reactions/processes; and (2) PAH presumably can undergo chemical modification under relatively mild treatment conditions (e.g., 500°C), potentially creating hazardous biologically active chemicals.

The results of this work can be relevant to many aspects of thermal remediation research technology. This work would be useful for understanding the effects of incinerator upset conditions which could create pyrolytic environments that can result in volatiles or char escaping destruction, or undergoing conversion to other potentially toxic compounds. Other potential applications include improved understanding of related thermal destruction technologies such as circulating fluidized beds and electric/infrared heaters, as well as newer or "innovative" thermal techniques such as pyrolysis, thermal desorption, in-situ vitrification, and microwave methods. Basic work on contaminant removal and identification of volatiles reaction pathways could also provide information helpful for process development, monitoring, and control for incineration and "innovative" processes operating under normal as well as upset conditions.

Thesis Supervisors: Jack B. Howard and William A. Peters
Titles (respectively): Professor of Chemical Engineering, and Associate Director, MIT Energy Laboratory
Dedicated to my parents,

Reisuke and Ayako Saito
Acknowledgments

First of all, I must say that my time here at MIT has been a stimulating, educational, and (as I hope this document will prove) productive time for me. From the help and friendships of many people in the Department, I have learned not only about research and problem-solving, but also about life, living, and the world around me. For this, I am eternally grateful to all of the people here I have had the fortune to meet and to those who have shared their time with me. Due to the sheer numbers, I apologize if I have missed some of you, but please be assured that I have not forgotten you.

First and foremost, I would like to thank my advisors Prof. Jack Howard and Dr. William Peters for their constant patience, kindness, support, and teaching during this project (and for showing me all of the things I can do with aluminum foil); Dr. Verónica Bucalá for her help and her open-minded, kind, supportive critiques of my various (sometimes crazy) ideas about the project and experimental work (and for letting me teach her the experimental ropes - watch out world!); my thesis committee (Prof. Adel Sarofim, Dr. Art Lafleur, and Prof. Mirella Flytzani-Stephanopoulos) for their constructive inputs to my work; and my UROP student, Eric Freeman, for his efforts and help in collecting some of the data in this work (and for allowing himself to be the first victim of my management style). I'd also like to thank everyone in the CORE Lab (Art Lafleur, Bill Busby, Elaine Plummer, Koli Tajhizadeh, Marcia Ross and John Durant) for their kindness and help whenever I was in there.

To those of you in and around the basement laboratory: Joe "Let me give you some advice - Just Do It" Marr and Jack "Oh no! I can't believe it!" Brouwer for teaching me some of their experimental and theoretical know-how, inspiring me with their great ideas and cheer, and sharing with me their time in sometimes not-so-productive pursuits like our daily political discussions; George Darivakis for showing me how to run the reactor and the intensity it takes to finish work quickly; Tony Modestino for his sound advice and spare parts which saved me on many occasions whenever I was in a real bind; Rich Shandross, Claes Tullin, Gernot Krammer, Christian Feldermand, Yanping Zhang, Wei Liu, Guido Sacchi, Alex Diaz, Scott Macaddam, and Bill Grieco for the many interesting conversations we've had about technologies and the world; Laura Giovane and Lori Yerkey for sharing with me their time, sunny personalities (definitely a must down in the basement!), and their experiences as undergraduates here at MIT; and Kathy Brownell for looking out for me (concerning Prof. Howard's schedule as well as free food) and for our always educational and amusing conversations about art and life.

To my "other" group: Professor Jeff Tester for showing me that while research is a serious enterprise, it should also be taken with a critical and humorous eye; Gabe Worley for his patience, perseverance, and humor while surviving a 10.40 TA and other athletic campaigns together (like our IM Ultimate Frisbee B-League championship); and Rick Holgate and Fred Armellini for sharing their wit and their technical expertise (well, I always got the former as well when I asked for the latter!).

To those of you who've helped balance my life away from research:

Die-hard Basketballers: Abdul "Automatic" Barakat for his cheer and camaraderie; Joy "Air" Mendoza for epitomizing fun and sportsmanship; Leo "Hey, Man!" Lue for showing me humor, determination and kindness all at once; Stathis Avgoustinatos for epitomizing passion and determination; Brian Laffey for his generosity on and off the
court; Dave Chang for his hilarity and an all-court game; Raju “Automatic from the Low Post” Raju for his good skill and nature; Michael “Fadeaway” Thompson - my nemesis (you know what I mean!); Xinjing Zhou and Luhong Bo for their kindness, sportsmanship, and great humor; and Nelson “In your face!” Lin for breaking my glasses (and his good-natured humor).

Volleyballers: To the die-hard Sunday volleyball players like Jim and Paige Mulholland, Rick and Laura Holgate, Mike and Heather Masonjones, Peter Nelson and Jaqui Lynch, Joy Mendoza, Mark Prausnitz, and Gil Huppert for making playing, organizing, and learning the game so exciting, fun, and worthwhile; and to the A-League Intramural Team (where one year we came so close to the championship) for giving me a chance at the big leagues and allowing me to develop my skills (with special kudos to Brian Kelley, Colin Wolden, Steve Spiegelberg, Dan Lasko, and Bruce Scruggs).

Dinner and a Movie friends: Shih-Tung and Susan Ngiam for always tasty dinner, sparkling conversation, our New England Fall Foliage excursion, and watching out for my personal life throughout my years here; Joe Marr for all of those dinner and movie excursions after work which allowed me to explore and develop my philosophical and political side (oh, the monster you did create!); Peter Nelson and Jaqui Lynch for all of those great pot-luck dinners, hilarious conversations, and for humbling me in "Doom"; Lars Genieser and Carlos Rojas-Guzman for sharing their time in helping me pursue the more artistic and philosophical side of me; Roy Kamimura, Sal Salim, Harsono Simka, and Angelo Kandas for the many funny dinners together at anyone’s or everyone’s expense, for introducing me to anime and Hong Kong films, and for preparing me for international travel; Dave Oda and Roger Tim not only for their hospitality and time, but for allowing me to express my more sarcastic side (remember "Stargate"?)

Lunch crowds: While there have been so many people who’ve made lunch in Walker Lounge so fun, "Most Memorable" Awards must go to Elaine Aufiero-Peters, Eddie Khouri, Matt DiPippo, Vikki Vlastnik, Tim Donahue, Vivek Mohindra, Rich Shandross, and Howard C. vert for their (shall I say) unique skills in how to completely change the tone of a conversation.

The Fall 1991 10.40 class for making learning how to teach students enjoyable and for keeping it relatively painless.

Lastly, while some would acknowledge Juan Valdez, for me it would be the Coca Cola Corporation. Without Coke and Dr. Pepper, I wouldn’t have made it through the late nights and rough mornings which can come often during graduate studies. Thanks must also go to the authors of "Calvin and Hobbes" (Bill Watterson) and "Dilbert" (Scott Adams) for giving me another humorous perspective on the engineering and scientific pursuit.

Financial support of this research by NIEHS Grant 1SO4675-06 (MIT-Superfund Hazardous Substances Basic Research Program), and by the U.S. E.P.A. Northeast Hazardous Substances Research Center is greatly appreciated. I am also indebted to Dr. Raymond M. Frederick of U.S. E.P.A. and Dr. Seymour Rosenthal from Foster Wheeler Envireponse Inc. for providing soil samples and information pertaining to the properties and composition of this soil.
# Contents

<table>
<thead>
<tr>
<th>Chapter 1: Introduction</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0 Introduction</td>
<td>15</td>
</tr>
<tr>
<td>1.1 Problem Statement</td>
<td>15</td>
</tr>
<tr>
<td>1.1.1 U. S. Soil Contamination Problem</td>
<td>15</td>
</tr>
<tr>
<td>1.1.2 PAH-Contamination at Sites</td>
<td>18</td>
</tr>
<tr>
<td>1.1.3 Thermal Methods for Soil Remediation</td>
<td>24</td>
</tr>
<tr>
<td>1.1.4 Chemical and Physical Processes of Soil Decontamination</td>
<td>28</td>
</tr>
<tr>
<td>1.2 Thesis Objectives</td>
<td>33</td>
</tr>
<tr>
<td>1.3 Literature Review</td>
<td>35</td>
</tr>
<tr>
<td>1.3.1 Electric Screen Heater Studies</td>
<td>36</td>
</tr>
<tr>
<td>1.3.2 Adsorption/Desorption Studies</td>
<td>37</td>
</tr>
<tr>
<td>1.3.3 Rotary Kiln Incinerator Studies</td>
<td>42</td>
</tr>
<tr>
<td>1.3.4 Soil Off-gas Emission Studies</td>
<td>45</td>
</tr>
<tr>
<td>1.4 Summary</td>
<td>46</td>
</tr>
<tr>
<td>1.5 References</td>
<td>47</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 2: Apparatus and Procedure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.0 Introduction</td>
<td>55</td>
</tr>
<tr>
<td>2.1 Experimental Approach</td>
<td>55</td>
</tr>
<tr>
<td>2.1.1 Overall Experimental Program</td>
<td>55</td>
</tr>
<tr>
<td>2.1.2 Selection of Experimental Parameters</td>
<td>56</td>
</tr>
<tr>
<td>2.2 Apparatus Description</td>
<td>61</td>
</tr>
<tr>
<td>2.3 Experimental Procedure</td>
<td>68</td>
</tr>
<tr>
<td>2.4 References</td>
<td>75</td>
</tr>
</tbody>
</table>
### Chapter 3: Experimental Data

3.0 Introduction 78

3.1 Temperature Study 78

3.1.1 Experimental Data 78

3.1.2 Discussion 88

3.2 Heating Rate Study 91

3.2.1 Experimental Data 91

3.2.2 Discussion 100

3.3 Summary 101

3.4 References 103

### Chapter 4: Data Analysis

4.0 Introduction 105

4.1 Soil Constituents 105

4.1.1 Soil Formation and General Makeup 105

4.1.2 Soil Minerals 107

4.1.3 Soil Organic Matter 111

4.2 Pyrolysis Studies 117

4.2.1 Pyrene Pyrolysis 117

4.2.2 Soil Pyrolysis 118

4.2.3 Other Possible Reactions 122

4.2.4 Mineral Effects on Pyrolysis 124

4.3 Modeling Approach 126

4.4 Model Theories 127

4.4.1 Multiple Independent Parallel Reaction (MIPR) Model 127

4.4.2 Competitive Reaction Model - Parallel Reactions (CRM-P) 130
4.4.3 Competitive Reaction Model - Series Reactions (CRM-S) 132

4.5 Data Analysis 133

4.6 References 143

Chapter 5: Conclusions 149

Chapter 6: Recommendations 151

Appendices 153

Appendix A - Detailed Experimental Run Procedure 153
Appendix B - Tables of Experimental Data 158
Appendix C - Debugged Computer Programs 177
Appendix D - Heat Transfer Calculations and Measurements 262
Appendix E - Intraparticle CO/CO₂ Equilibrium Calculation 271
List of Figures

1-1 Growth of number of sites placed on the National Priority List from 1982 to 1991.

1-2 Overview of ROD’s through fiscal year 1991 with respect to treatment technologies selected: “Innovative” vs. “Conventional”

1-3 Overview of soil remediation methods selected at Superfund sites through fiscal year 1991.

1-4 Schematic of a rotary kiln incinerator system.

1-5 Possible pathways for transport and reaction of soil contaminants during thermal treatment

2-1 Schematic of the modified electrical screen heater reactor: a) reactor from top view, b) reactor from side view, c) closeup of the molded hot stage

2-2 Schematic representations of temperature-time profiles obtainable with the electrical screen heater, captive sample reactor

2-3 Schematic of Reactor System

2-4 Soil contamination containers

2-5 Gas chromatograph configuration for light gas analysis

3-1 Mass loss data for neat and 4.75 wt. % pyrene-contaminated soil as affected by peak temperature (nominal heating rate = 1000°C/s, zero holding time)

3-2 Effect of peak temperature on implied pyrene removal determined as mass loss difference between neat and contaminated soil, assuming no interaction between pyrene and soil (nominal heating rate = 1000°C/s, zero holding time)

3-3 Mass of volatilized tars for neat and 4.75 wt. % pyrene-contaminated soil as affected by peak temperature (nominal heating rate = 1000°C/s, zero holding time)

3-4 Carbon monoxide release for neat and 4.75 wt. % pyrene-contaminated soil as affected by peak temperature (nominal heating rate = 1000°C/s, zero holding time)

3-5 Carbon dioxide release for neat and 4.75 wt. % pyrene-contaminated soil as affected by peak temperature (nominal heating rate = 1000°C/s, zero holding time)

3-6 Methane release for neat and 4.75 wt. % pyrene-contaminated soil as affected by peak temperature (nominal heating rate = 1000°C/s, zero holding time)
3-7 Acetylene release for neat and 4.75 wt. % pyrene-contaminated soil as affected by peak temperature (nominal heating rate = 1000°C/s, zero holding time)

3-8 Ethylene release for neat and 4.75 wt. % pyrene-contaminated soil as affected by peak temperature (nominal heating rate = 1000°C/s, zero holding time)

3-9 Weight loss data for neat and 4.75 wt. % pyrene-contaminated soil as affected by heating rate (nominal peak temperature = 550°C, zero holding time)

3-10 Mass of volatilized tars for neat and 4.75 wt. % pyrene-contaminated soil as affected by heating rate (nominal peak temperature = 550°C, zero holding time)

3-11 Carbon monoxide release for neat and 4.75 wt. % pyrene-contaminated soil as affected by heating rate (nominal peak temperature = 550°C, zero holding time)

3-12 Acetylene release for neat and 4.75 wt. % pyrene-contaminated soil as affected by heating rate (nominal peak temperature = 550°C, zero holding time)

3-13 Carbon dioxide release for neat and 4.75 wt. % pyrene-contaminated soil as affected by heating rate (nominal peak temperature = 550°C, zero holding time)

3-14 Methane release for neat and 4.75 wt. % pyrene-contaminated soil as affected by heating rate (nominal peak temperature = 550°C, zero holding time)

3-15 Ethylene release for neat and 4.75 wt. % pyrene-contaminated soil as affected by heating rate (nominal peak temperature = 550°C, zero holding time)

4-1 Scheme for the fractionation of humic substances

4-2 Scanning electron micrographs of humic acid and fulvic acid

4-3 Proposed chemical structures for humic acid

4-4 Proposed chemical structures for fulvic acid

4-5 Six bipyrenes observed by Mukherjee et al.

4-6 Condensation products of bipyrenes observed by Mukherjee et al.

4-7 Experimental data and MIPR (multiple independent parallel reaction) model “best fit” values for weight loss of neat EPA soil as affected by heating rate. Initial soil mass = 50 mg, nominal peak temperature = 550°C, zero holding time.

4-8 Experimental data and CRM-P (competitive reaction model - parallel reactions) model “best-fit” values for acetylene yields from a neat EPA soil as affected by heating rate. Initial soil mass = 50 mg, nominal peak temperature = 550°C, zero holding time.

4-9 Experimental data and CRM-P (competitive reaction model - parallel reactions) model “best-fit” values for acetylene yields (A_o = 8.16 μg, E_p = 55070 cal/gmole, E_b = 52140 cal/gmole, k_p0 = k_b0 = 10^{13} /s). Both temperature and heating rate data used for a single fit. Initial soil mass = 50 mg, nominal peak temperature = 550°C, zero holding time.
Experimental data and CRM-S (competitive reaction model - series reactions) model "best fit" values for acetylene yields from a neat EPA soil as affected by heating rate. Initial soil mass = 50 mg, nominal peak temperature = 550°C, zero holding time.

CRM-S (competitive reaction model - series reactions) model "best fit" values as affected by \( A_0 \), the minimum mass of soil constituent necessary for acetylene production. Initial soil mass = 50 mg, nominal peak temperature = 550°C, zero holding time.
List of Tables

1-1 PAH Contamination Levels Observed by VISITT Vendors
1-2 Treatment Effectiveness of Several Soil Cleanup Technologies
1-3 Sample Temperature-Time Profiles Leading to Complete Removal of Fuel Oil from Soil
2-1 Summary of Thermal Treatment Conditions Available and Used
4-1 Common non-silicate minerals in soils
4-2 Common silicate minerals in soils
4-3 Multiple independent parallel reaction model (MIPR) parameters obtained for weight loss and volatiles evolution from an EPA Synthetic Soil Matrix
4-4 Reactions of clay minerals at elevated temperatures
Chapter 1

Introduction

1.0 Introduction

This chapter introduces the contaminated soil problem present in the United States, surveys the prevalence of polycyclic aromatic hydrocarbon contaminants at contaminated sites, provides an overview of thermal soil remediation technologies as a means to cleanup contaminated soil, discusses relevant work found in the published literature studying various aspects of thermal treatment, identifies research needs, and specifies the focus of this work.

1.1 Problem Statement

1.1.1 U.S. Soil Contamination Problem

Public outcry over waste disposal mismanagement at sites such as New York's Love Canal and Kentucky's Valley of the Drums led to the passage of the Comprehensive Environmental Response, Compensation, and Liability Act (CERCLA), otherwise known as the Superfund Law, in 1980. CERCLA, through the Environmental Protection Agency (EPA), initiated a national program to identify and clean up the worst chemically contaminated sites across the United States. Since then and after its reauthorization under the Superfund Amendments and Reauthorization Act (SARA) in 1986, National Priority List (NPL) has grown to contain 1320 sites (as of October, 1994) [1], designating these sites to be highly threatening to the environment. After seeing the number of NPL sites
more than triple since 1982 (Figure 1-1), EPA currently expects the NPL to grow by 50 - 100 sites annually [2].

Progress at remediating NPL sites has been slow due to many factors, including finances [3,4] and technical difficulties [3,4], as well as delays due to litigation [5] (not discussed here). A recent General Accounting Office study [1] reports that although significant progress has been made at half of the NPL sites, work is complete at only 52 sites by the end of fiscal 1993 and that 40% of the fiscal 1993 funds obligated for cleanups went to just 13 sites.

Future prospects for fulfilling Superfund’s objectives of cleaning up the nation’s worst abandoned hazardous waste sites remains uncertain [3,4]. J. Winston Porter, former head of the EPA cleanup program, has estimated that the total number of sites that require some type of cleanup numbers in the tens of thousands [3], encompassing not only federal Superfund sites but also state-sponsored cleanup sites and RCRA (Resource Conservation and Recovery Act) sites. By 1991, the EPA had assessed over 30,000 sites of which 48% were undergoing further assessment to determine the necessity of cleanup [2]. Abandoned dump sites containing hazardous chemicals are still being discovered and characterized, with each location containing hundreds to hundreds of thousands of tons of contaminated soil [6]. Prospects for lowering the high cost of cleanup (on average, approximately $26 million (1991) per site [2]) is uncertain since EPA believes that these cost estimates will rise since some of the more complex sites have yet to be cleaned up [2].

Further adding to the task facing the EPA is the growing public concern over emissions and residuals remaining from cleanup work. Because soil remediation technologies often involve addition of chemicals to soils (soil flushing or solvent extraction) or creating air or other emissions (incineration, thermal desorption, vacuum/vapor extraction), the public is concerned that the treatment process does not do more harm than the original contaminated soil.
The Growth of the NPL
From the First List to the Present

![Graph showing the growth of the NPL from 1982 to 1991.]

Source: CERCLUS (1991)

Figure 1-1. Growth of number of sites placed on the National Priority List from 1982 to 1991. (proposed = recommended for addition, final sites = slated for cleanup, deleted = cleanup actions completed)

Landfills were the dominant means of disposal for hazardous chemicals and wastes until the late 1980's. Landfills have become a less attractive disposal option since landfill availability has fallen, landfill regulations have become more stringent, and many materials (particularly wastes with high concentrations of or highly mobile substances) are unsuitable for landfilling. Steadily rising landfilling costs have prompted development and use of soil cleanup technologies. For example, EPA saw the use of cleanup technologies increase at NPL sites from about 54% in 1987 to 79% by 1990 [2], with the remainder of sites being cleaned up using containment technologies (segregation of waste in a particular place).

But soil cleanup technology is still a new field. As more sites have been evaluated and cleaned up using a variety of techniques, more has been learned about the cleanup process. However, the technical uncertainties and surprises encountered at cleanup sites have made the evaluation and use of remediation technologies difficult [4]. These uncertainties in combination with the above economic and public concerns the have stimulated research to optimize and further understand the fundamental processes behind existing soil remediation techniques to: 1) reduce treatment cost, 2) maximize destruction, removal, or isolation, and 3) minimize toxic emissions from remediation activities.

1.1.2 PAH-Contamination at Sites

Polycyclic aromatic hydrocarbons (PAH's), a class of compounds containing fused benzene rings, are encountered at many hazardous waste sites, particularly those that have handled wood preservatives, coal and pine tars, and sludges [7,8]. Simonich et. al [9] recently published an interesting paper suggesting that surface soils absorb up to about 44±18% of airborne anthropogenic PAH originating from sources such as automobiles, residential heating, burning, and industrial processes. But whether PAH's observed at Superfund sites were deposited by this mechanism is unknown. PAH's are of practical
interest since these compounds may pose a threat to human health, and are difficult to remove from soil relative to many other organic contaminants.

Human exposure to PAH's from contaminated sites can occur through various pathways such as slow evaporation from sites, and contamination of surface water and ground water (affecting drinking water and aquatic organisms) [10]. Human or biotoxicity of these chemicals has been well documented in the literature [11-14]. Many PAH's are mutagenic, carcinogenic or both. Many PAH's are of high molecular weight (MW) and exhibit low volatility, recalcitrance to microbial breakdown, and limited solubility in, and limited extractability from soil into, many liquids such as water and surfactant-water solutions [7]. Thus, incineration [15] or other forms of thermal destruction may be the preferred means for remediating sites contaminated with high MW PAH. In the present study, pyrene was chosen as a model contaminant because it is one of the higher MW PAH often present in heavy oils, tars, and some sludges at hazardous waste sites.

Frequency of the occurrence and PAH concentrations at Superfund sites has been difficult to ascertain. Despite searching the literature and inquiring with our contacts at EPA [16] and the EPA Region I Library here in Boston, no tabulated data profiling all of the sites on the NPL with respect to frequency of contaminant occurrence and contaminant concentrations could be found. However using available information from these sources and other sources, a very approximate picture of the national scene was assembled and is discussed below.

**PAH Occurrence at Superfund Sites**

Only two sources could be found profiling all of the sites on the NPL: the 1991 ROD Annual Report and the 1990 Innovative Technologies Semi-Annual Status Report, both published by the EPA.
1991 ROD Annual Report [17]- This report is an accumulation of data concerning the 945 ROD's (Records of Decision) approved by EPA up to and including the fiscal year 1991. An ROD is a document certifying that a remediation process has been chosen for a Superfund site or part of a site following CERCLA guidelines. From the Cumulative ROD Keyword Lists, it was manually counted that 127 ROD's or 13.4% of ROD’s had PAH’s among the primary hazardous substances detected at an NPL site. Since one site could have more than one ROD, determination of the precise fraction of NPL sites containing PAH could not be ascertained. Also in this report, there was no information regarding contaminant level for each ROD.

1990 Innovative Technologies Semi-Annual Status Report [8]- Although an accumulation of ROD information through fiscal year 1990, this report focuses only on the use of "innovative" technologies, such as soil washing, solvent extraction, bioremediation, vacuum extraction, dechlorination, in-situ vitrification, chemical treatment, and thermal desorption as opposed to "established" technologies such as incineration, solidification/stabilization, soil aeration, in-situ flaming, and chemical neutralization. Innovative technologies are defined by EPA as those with insufficient data on performance and cost to allow adequate assessment of their potential use for Superfund applications [8]. Manual counting from the detailed site information provides an approximate idea of the frequency of PAH’s at these selected NPL sites. It was found that 28 of 140 ROD's (20%) or 29 of 137 sites (21%) contained PAH contaminated soils or sediments. Innovative technologies accounted for 40% of the ROD’s signed through fiscal year 1990 (Figure 1-2). This implies that at the least ~8-10% of all NPL sites have PAH’s listed among the primary contaminants.
Figure 1-2. Overview of ROD's through fiscal year 1991 with respect to treatment technologies selected: "Innovative" vs. "Conventional"

PAH Contamination Levels at NPL Sites

The only information obtained comes either from cleanup-related literature about specific sites, or from industry-wide type of sources discussing in general contamination levels observed at sites, not necessarily concerning PAH specifically.

*Site-specific data:* Acharya and Ives [18] from IT Corporation reported PAH contamination levels at a former wood preservation facility ranged from ppm levels to up to 2 percent in their paper describing their cleanup experience at the Bayou Bonfouca Superfund site in Louisiana. At another abandoned wood-treatment Superfund site in Florida, Tobia et. al [19] tested a soil washing technique on a soil containing an average of approximately 0.5 wt.% PAH, with some size fractions of the soil containing as much as 1.5 wt.%.. In work by Durant et. al [20,21], PAH levels of 36 to 92 ppm were found in lake sediments in the Aberjona watershed. This watershed region includes the Industriplex Superfund site located in Woburn, MA.

*Industry-wide type data:* A compilation of site and technology performance data from an EPA survey of vendors using “innovative” treatment technologies produced a PC-based database called VISITT 3.0 (Vendor Information System for Innovative Treatment Technologies, version 3.0). While this database is not restricted to Superfund sites, the levels of contamination observed by the various vendors is potentially very informative (Table 1-1).
Table 1-1
PAH Contamination Levels Observed by VISITT Vendors

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Site (State)</th>
<th>Cont. Level (wt.%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENSR</td>
<td>AT&amp;T Site (IL)</td>
<td>0.01700</td>
</tr>
<tr>
<td>Waste Stream Technologies, Inc.</td>
<td>Pacific Place (Canada)</td>
<td>2.5000</td>
</tr>
<tr>
<td>Waste Stream Technologies, Inc.</td>
<td>Wastewater Tech. Center (Canada)</td>
<td>0.1800</td>
</tr>
<tr>
<td>Ecology Technologies Intl.</td>
<td>Waco GT (TX)</td>
<td>0.0265 - 0.3685</td>
</tr>
<tr>
<td>Chester Environmental</td>
<td>Confidential (NH)</td>
<td>0.1000 - 0.6000</td>
</tr>
<tr>
<td>MycoTech Corp.*</td>
<td>Montana Pole &amp; Treating (MT)</td>
<td>0.0468 - 0.2850</td>
</tr>
<tr>
<td>ABB Environmental Services</td>
<td>Boys Market (CA)</td>
<td>0.8000</td>
</tr>
<tr>
<td>Arc Tech</td>
<td>coal gasification plant (confidential)</td>
<td>0.0080</td>
</tr>
</tbody>
</table>

* Superfund site

Again, PAH are present typically in the hundreds to thousands of ppm levels, with some sites going as high as a few weight percent. A survey of contractors treating petroleum-contaminated soils appears to confirm this observation, except in their case contamination levels were as high as several percent [22].

Also in conversations with Dr. Raymond M. Frederick [16] of the U.S. E.P.A., he also indicated that typical levels of organics contamination at Superfund sites are usually in the parts per million, with some sites going as high as of order 5 wt.%. To provide an objective basis for evaluating various remediation technologies, the group in which Dr. Frederick works at EPA formulated some Synthetic Soil Matrices containing typical “low” (0.3 wt.%) and “high” (3.1 wt.%) levels of organic contaminant based on their observations at Superfund sites.

But clearly there are cases where contamination levels are far worse. Luthy et. al [7] report that at some former manufactured gas plant (MGP) sites, free tars have been found in wells and pits, although often most of the tars typically found at MGP sites are trapped by capillary forces in soil, occupying 5-25% of the pore volume.
PAH Contamination Level and Frequency Summary

If the above information is representative of Superfund sites, it appears that PAH's are found at of order 10-15% percent of NPL sites at levels typically of order 0.5 - 1.0 wt. % with levels being as high as of order 5 wt.% at some sites. While the data available are very piece-meal, it is clear that PAH's are found at a significantly large number of NPL sites in significant concentrations.

1.1.3 Thermal Methods for Soil Remediation

Site evaluations and cleanups have allowed for assessment of various remediation technologies. Many types of remediation technologies have been tested with differing degrees of success for a variety of chemical contaminants encountered in the field. Among these technologies are: 1) incineration and thermal destruction, 2) solidification, stabilization, and neutralization, 3) volatilization, soil aeration, and vacuum extraction, 4) soil washing and flushing, and 5) biotreatment. Of these five remediation technologies, incineration and thermal destruction has been found to be effective over the widest range of organic compounds encountered at contaminated sites as shown in Table 1-2. Well-designed incineration/thermal destruction systems provide the highest overall degree of destruction of many hazardous waste streams [23] (destruction or removal efficiencies, or DRE's, for specific hazardous constituents as high as 99.9999% [4,24] are achievable).

While the purpose of this section is not to argue whether thermal destruction and incineration techniques are the best techniques available, versatility in handling organic contaminants and extensive past experience with similar systems are two of many factors which have led to the selection of this class of technologies for remediation at many contaminated sites. Figure 1-3 illustrates how extensively incineration and thermal destruction techniques have been chosen in the past by the EPA for remediation actions.
Figure 1-3. Overview of soil remediation methods selected at Superfund sites through fiscal year 1991.

According to a 1991 Thermal Treatment Remediation Contractor Survey [6], the typical cost of decontaminating soil by incineration/thermal destruction techniques is estimated to be approximately $300 - $700 per ton of soil.

While thermal remediation techniques such as incineration and thermal destruction have favorable characteristics for cleaning up sites, public concern has recently moved the EPA away from this technology towards less severe ("innovative") techniques such as thermal desorption and bioremediation. Strong public concerns about the health and environmental impacts of process emissions [15], and about contaminant retention or toxicants creation in the soil residues during normal operations and particularly from process upsets [25,26] has led to the recent poor public acceptance of incineration and thermal destruction techniques. Another criticism is that these processes can leave the soil devoid of organic constituents after treatment, resulting in a soil which could not support vegetation ("dead soil"). Depending on the issues faced at a particular site and the planned future use of the contaminated site, these disadvantages may be outweighed by their advantages or the lack of a viable alternative.

Despite the current negative light, thermal remediation technologies will continue to be used in the next several years. According to a 1991 thermal treatment contractor survey sponsored by the Air and Waste Management Association [6], only about one-third of the contracted soil volume has been treated. Also because only a small fraction of the NPL site cleanups have been completed to date, it is likely that the much of the contracted work using these techniques have yet to be implemented. In addition, some in industry believe that thermal destruction techniques could be the least costly option for handling soil contaminated with PAH-containing materials such as heavy oils, tars, and selected sludges [15]; or that alternatives to thermal destruction for these sites may be difficult to find [7].

It is believed that further understanding of the soil remediation process by thermal destruction could increase public acceptance of incineration and other thermal destruction
techniques. But in the meantime, the long-term role of this technology in relation to competing technologies has yet to be determined, despite its historical support by EPA.

Table 1-2
Treatment Effectiveness of Several Soil Cleanup Technologies

<table>
<thead>
<tr>
<th>Treatability Group</th>
<th>Bioremediation</th>
<th>Immobilization</th>
<th>Dechlorination</th>
<th>Solvent Extraction</th>
<th>Thermal Destruction</th>
<th>Low-temp. Thermal Desorption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nonpolar halogenated aromatics</td>
<td>B</td>
<td>C\textsuperscript{a}</td>
<td>B\textsuperscript{a}</td>
<td>B</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>Halogenated PCB's, dioxins, furans and their precursors</td>
<td>B</td>
<td>C\textsuperscript{a}</td>
<td>B</td>
<td>B</td>
<td>A</td>
<td>C\textsuperscript{c}</td>
</tr>
<tr>
<td>Halogenated phenols, cresols, amines, thiols and other polar aromatics</td>
<td>B</td>
<td>C\textsuperscript{a}</td>
<td>B\textsuperscript{a}</td>
<td>B</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>Halogenated aliphatic compounds</td>
<td>A</td>
<td>X</td>
<td>B\textsuperscript{a}</td>
<td>B</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>Halogenated cyclic aliphatics, ethers, esters, and ketones</td>
<td>C\textsuperscript{b}</td>
<td>C\textsuperscript{b}</td>
<td>B\textsuperscript{b}</td>
<td>B\textsuperscript{b}</td>
<td>A</td>
<td>C\textsuperscript{c}</td>
</tr>
<tr>
<td>Nitrated compounds</td>
<td>A</td>
<td>C\textsuperscript{b}</td>
<td>C\textsuperscript{c}</td>
<td>B</td>
<td>A</td>
<td>C\textsuperscript{b}</td>
</tr>
<tr>
<td>Heterocycles and simple non-halogenated aromatics</td>
<td>A</td>
<td>X</td>
<td>C\textsuperscript{c}</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>Polynuclear aromatics</td>
<td>A</td>
<td>C\textsuperscript{a}</td>
<td>C\textsuperscript{c}</td>
<td>B</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>Other polar nonhalogenated organic compounds</td>
<td>A</td>
<td>C\textsuperscript{a}</td>
<td>C\textsuperscript{c}</td>
<td>B</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>Nonvolatile metals</td>
<td>X</td>
<td>A</td>
<td>C\textsuperscript{c}</td>
<td>B</td>
<td>C\textsuperscript{c}</td>
<td>C\textsuperscript{c}</td>
</tr>
<tr>
<td>Volatile metals</td>
<td>X</td>
<td>A</td>
<td>C\textsuperscript{c}</td>
<td>B</td>
<td>X</td>
<td>C\textsuperscript{c}</td>
</tr>
</tbody>
</table>

A = Demonstrated effective  
B = Potentially effective (in certain situations)  
C = No expected effectiveness  
X = Not recommended (potentially adverse effects to environment or process)

\textsuperscript{a} Data were screened from consideration for regulatory development purposes. These data suggest that this technology may be effective in certain situations.  
\textsuperscript{b} Data were not available for this treatability group. Data for Compounds with similar physical and chemical characteristics suggest that this technology may be effective in certain situations.  
\textsuperscript{c} The physical and/or chemical characteristics of the constituent of this treatability group suggest that this technology would not be effective.

1.1.4 Chemical and Physical Processes of Soil Decontamination

To illustrate the fundamental processes taking place in an industrial soil decontamination system and potential problems which have sparked public concern with thermal treatment technologies, a rotary kiln incineration system will be used as an example. While there are many thermal destruction techniques besides rotary kiln incineration such as circulating fluidized beds, electric/infrared heaters, pyrolysis, and in-situ vitrification, rotary kilns have constituted a large percentage of the thermal treatment equipment used or selected. Of the ROD’s approved by EPA in 1988 [27], 80% of those to employ incineration as the remediation technique used rotary kilns. Also in a 1991 thermal treatment contractor’s survey [6], 33 of 57 respondents (plus 10 more using rotary dryers) employed rotary kilns.

A rotary kiln incinerator system is typically made up of three sections (see Figure 1-4): the primary desorber (the rotary kiln), the secondary combustion chamber (SCC), and a train of air pollution control devices. The rotary kiln is a refractory-lined rotating cylinder tilted slightly (approximately 3° from horizontal). Contaminated soil is supplied into the upper end and moves through the cylinder by gravity and agitation. Residues are discharged at the lower end. A turbulent flame inside the kiln, co-current or countercurrent to the soil flow, maintains the kiln temperature and heats the soil to the desired treatment temperature. The kiln rotation helps transfer and mix the waste, enhancing heat and mass transfer. While the soil is in the kiln, contaminants are believed to be partially volatilized in a pyrolytic environment, and partially burned by reaction with oxygen in the kiln flame. Surviving volatiles flow to the secondary combustion chamber, which is an afterburner section where the gas phase oxidation is completed. The flue gas is then sent to air pollution control devices such as baghouses and venturi scrubbers for particulate and acid gas removal.
Figure 1-4. Schematic of a rotary kiln incinerator system.

Problems facing this technology are not only the economics of the process, but also public opposition. Regulatory bodies and the public are concerned about further degradation of the environment, the effects of long-term low-level exposure to chemicals, and the possibility that incinerator emissions may pose unnecessary hazards to humans and the environment. There is particular concern that DRE's will fall off and emissions of hazardous materials will increase during upset conditions [26] such as volatiles "puffs", mixing upsets, and temperature (or time) excursions.

"Puffs" are surges of gas phase volatiles, which may result from rapid pyrolysis of solids or from prompt evaporation of volatile contaminants. Too rapid introduction of waste, or waste introduced at too high a concentration can lead to the formation of "puffs" in rotary kilns. "Puffs", which may contain hazardous compounds, can disrupt normal operation by flooding regions of the incinerator chamber with more volatiles than can be burned out at the kiln temperature with the oxygen immediately available. Through better characterization of these "puffs", methods to facilitate their burnout can be suggested. Mixing upsets are conditions where, through inadequate mixing, insufficient oxygen is supplied to completely oxidize the volatilized waste. Mixing upsets can be caused by a variety of means such as "puffs", or the diversion or interruption of waste or oxidant flows. Despite a stable flame and adequate oxygen, complete oxidation of volatilized waste cannot occur if the incinerator temperature is not sufficiently high for an appropriate length of time. Less than desired temperatures can be due to wastes with low heating values or high water content, air preheater malfunctions, "puffs", or cold surfaces in the reactor. Mixing upsets, "puffs", or temperature (or time) deviations can create pyrolytic environments that can result in volatiles or char escaping destruction, or undergoing conversion to other potentially toxic compounds. Unless destroyed in downstream afterburner or cleanup equipment, these volatiles or toxic compounds could be emitted by the incinerator [28].
Through better technical understanding of the chemical and physical processes responsible for incinerator behavior during normal and upset conditions, process development, monitoring, and control will benefit and public confidence toward incineration processes could be improved. But soil decontamination is accomplished in industrial equipment through a potentially complex set of several coupled and/or competing physical and chemical processes (e.g. heat transfer, mass transfer, evaporation/desorption, secondary chemical reactions, and soil reactions) which may contribute to the steady state or upset performance. If contaminant removal from soil is viewed on the soil-particle scale, the map of potential processes taking place is complex (Figure 1-5). Then if interparticle processes (such as interparticle heat and mass transfer, desorption and readsorption of the contaminant between particles, soil mixing effects in ex-situ methods, etc.) are added as in practical-scale equipment, determination of the fundamental processes controlling various aspects of decontamination (for example, removal efficiency or by-products formation) becomes difficult.

Several prior studies have contributed to better understanding of these and related questions. Among these works, Tognotti et al. [23] and Flytzani-Stephanopoulos et al. [29] have studied adsorption and desorption of contaminants from individual particles representative of soils. Darivakis [30] reports that rapid heating of a thin, approximately single monolayer of clay particles at a nominal heating rate of 1000°C/s gave weight losses consistent with essentially complete removal of pyrene contamination at temperatures above 600°C. Bucalá et. al [31] performed a similar study with fuel oil contaminated soils, examining the effects of treatment temperature, heating rate, and treatment time. Wu et al. [10,32] investigated mass transfer in soil beds to determine equilibrium and mass transfer parameters for contaminants, such as gas-soil partition coefficients, heats of adsorption, axial dispersion coefficients, and intraparticle diffusion coefficients. Lighty et al. [33-36] studied contaminants adsorption to and desorption from soil beds. Studies of rotary kilns include examination of heat and mass transfer effects by Silcox et al. [37] and Owens et al.
Figure 1-5. Possible pathways for transport and reaction of soil contaminants during thermal treatment.
[38,39], as well as modeling of puff formation in terms of volatilization from a fracturing solid phase by Linak et al. [40], Wendt et al. [41]. Using GC-MS, McClennan et al. [42,43] detected a large number of by-product compounds when thick beds (~5 - 8 cm) contaminated with alkylbenzenes were heated in air. Detailed understanding of how these compounds formed remains to be established, but possible sources are soil decomposition and chemical reactions of contaminant or contaminant conversion products with soil. The studies, particularly those by McClennan [42,43], raise interesting questions regarding contaminant retention, by-products release, and potential toxicants creation due to competition between reaction kinetics and heat/mass transfer related processes.

1.2 Thesis Objectives

Some industrial and bench-scale equipment studies suggest that chemical transformation of the contaminant may be taking place under certain operating conditions simultaneously with contaminant removal. Public concerns over potential health effects from process emissions, as well as economic pressures to minimize cost of thermal treatment methods, provide a strong impetus for researching the effects of operating conditions on contaminant removal and possible reaction of the contaminant to other products. Process development, monitoring, and control will benefit from detailed knowledge concerning these issues. From previous experience with the rapid devolatilization of coal [44-46] and biomass [47] at MIT, soil cleanup efficiencies as well as yields and compositions of co-products are expected to depend strongly on the temperature-time history of the soil.

Based on this information, the focus of this work has been to provide some insight into the following questions which are of practical and scientific importance:
(1) how rapidly and efficiently are targeted contaminants removed from the soil, “destroyed”, or chemical transformed to benign products?

(2) are the original contaminants transformed to (other) toxicants?

(3) what is role of soil itself throughout the cleanup process? (does soil decompose to (non)toxic products? does the contaminant affect soil decomposition, e.g. by reacting with the soil or soil-derived volatiles?)

(4) how are (1) - (3) affected by operating conditions?

By systematically varying reactor conditions (heatup rate and final temperature) in an inert atmosphere, the purpose of this work was to monitor weight loss as well as volatiles and off-gas production from a thin layer of soil (for both an uncontaminated synthetic soil matrix and samples of the same soil artificially contaminated with pyrene) in an effort to provide some insight into the above questions.

The results of this work can be relevant to many aspects of thermal remediation research and technology. As mentioned in the previous section, this work would be useful for understanding the effects of rotary kiln incinerator upset conditions which could create pyrolytic environments that can result in volatiles or char escaping destruction, or undergoing conversion to other potentially toxic compounds. This work could be extended to related thermal destruction technologies such as circulating fluidized beds and electric/infrared heaters, as well as newer or “innovative” thermal techniques such as pyrolysis, thermal desorption, in-situ vitrification, and microwave methods.

These more novel techniques could potentially have advantages over the incineration techniques, but could also suffer from similar problems under certain circumstances. As pyrolytic processes use little or not air, the contaminant is converted to char and volatiles [48] and the gross volume of air emissions could be reduced by a factor of 5 to 100 over conventional thermal destruction techniques [27]. But if the pyrolysis
process is not carried out properly, hazardous off-gases or PAH’s could be produced as in combustion processes under fuel-rich conditions. A similar process is *in-situ* vitrification, involving heating contaminated soil in the ground by electrical resistance, thereby pyrolyzing organic constituents and liquefying the soil so that upon cooling any hazardous inorganic contaminants such as poisonous metals are sealed within a glassified mass. Thermal desorption units volatilize contaminants from the soil at milder conditions than incinerators, but localized “hot spots”, if created due to insufficient heat transfer, could produce conditions sufficient for chemical transformation of the contaminant.

Hence, basic work on contaminant removal and identification of volatiles reaction pathways could provide information helpful for process development, monitoring, and control for incineration and “innovative” processes operating under normal as well as upset conditions.

### 1.3 Literature Review

**Overview**

The literature examining various processes underlying soil remediation by thermal techniques will be discussed in four groups: 1) electric screen heater studies, 2) adsorption and desorption investigations with single soil particles and packed soil beds, and by thermogravimetry, 3) small- and field-scale rotary kilns studies, and 4) preliminary work identifying and quantifying volatiles and off-gases produced from volatilizing contaminants from soil. Previous work on topics such as soil pyrolysis and pyrolysis of pyrene will be discussed in Chapter 4.
1.3.1 Electric Screen Heater Studies

Darivakis [30] at MIT investigated pyrene removal efficiency from a clay soil using an electric screen heater. Artificially-contaminated and neat soil were heated to temperatures between 200 - 800°C at a heatup rate of 1000°C/s. By difference in soil weight loss, the experiments indicated that pyrene was completely removed at temperatures greater than 600°C. A multiple independent parallel reaction model was used to fit the weight loss data to obtain global rate parameters for pyrene removal. Hence for the model, the activation energy found for the removal of pyrene from the soil was calculated as 37.1 kcal/mole. This is almost three times the heat of vaporization at the boiling point for pyrene (13.9 kcal/mole) probably due to desorption processes. However, there are two experimental factors which may affect the results. First is that the soil particles broke up upon heating, so that some soil was lost leading to high weight loss and wide scatter in the data. The second is that the soil contamination process used often created pyrene crystals of a size of the same order of magnitude as the soil particles. These were simply crushed gently, and the resulting contaminated soil was used. The second factor could lead to a slightly lower temperature for which pyrene is removed from soil completely, while both factors could contribute to a low estimate of the activation energy for the removal of pyrene. Despite these limitations, Darivakis’ work forms a basis of this study due to the apparent success of rapid removal of pyrene from a clay soil.

Using a slight modification to the Darivakis apparatus (discussed in Chapter 2), Bucalá et. al [31] conducted a very similar investigation of the effects of heating rate (200 - 6000°C/s), final temperature (300 - 700°C), holding time (0 - 25 sec), and contamination level (4 and 7 wt.%) on removal of a no. 2 fuel oil from a synthetic soil matrix. Experiments showed that thermal treatment can substantially eliminate no. 2 fuel oil from contaminated soil, with treatment time, temperature, and heating rate affecting rates and extents of removal. Different combinations of conditions led to complete fuel oil removal:
Under some circumstances, tradeoffs in heating rate, temperature, or holding time can be made to obtain same extents of removal. Changes in initial contamination level from 7 to 4 wt.% showed little effect on fractional extent of decontamination for 1000°C/s heating to 300°C for various holding times. The volatilized fuel oil was observed to experience significant chemical transformations at temperatures above 500°C for 1000°C/s heating rates.

1.3.2 Adsorption/Desorption Studies

Single Particle Studies

Stephanopoulos et al. [23,29] used an electrodynamic balance (EDB) to study adsorption and desorption of toluene and carbon tetrachloride onto single particles of different pore sizes (montmorillonite, Spherocarb, and Carbopack) from a gas stream containing the contaminant. The EDB uses electric fields to suspend a 90 - 180 μm dia. particle in a gas stream at 25°C, measuring particle mass gain and loss over time using the measured electric field strengths required to suspend the particle.

Characteristic adsorption and desorption rates and adsorbed amounts differed for each particle type. The amount of contaminant adsorbed was not a function of particle BET
surface area, but a strong function of particle intrusion volume (total particle pore volume, for pores greater than 30 nm diameter) as measured by mercury porosimetry. Based on the mass adsorbed per gram of particle, montmorillonite adsorbed the most contaminant, followed by Sphercarb and Carbosphere. Sphercarb adsorbed less than montmorillonite, despite having the highest BET surface area of the three particle types. Sphercarb consists mainly of micropores (<3 nm) where the contaminant is highly likely to form a liquid phase, and hence reduce contaminant access to the particle's internal surface area. This argument was supported by the results on Carbosphere, which had large pores, low surface area, the smallest intrusion volume, and the lowest uptake of contaminant. Desorption studies showed that a significant fraction of contaminant is irreversibly adsorbed onto the particle. In an order of magnitude analysis comparing the amount of contaminant irreversibly adsorbed to that required to cover the particle with a monolayer, the contaminant was found to irreversibly adsorb onto the particles in roughly a monolayer.

**Thermogravimetric Analyzer (TGA) Studies**

Han [49] studied adsorption and desorption of toluene, monochlorobenzene, and dichlorobenzene onto montmorillonite and Sphercarb particles. The desorption process was found to depend on soil types, especially the presence of micropores in the soil particle. Results indicate that the micropores are responsible for the slow desorption of the final monolayer of adsorbed contaminant after the initial rapid desorption period, that heating of the soil is very effective for achieving complete desorption, and that different contaminant-particle combinations can lead to widely varying minimum temperatures required for complete contaminant removal. For soil particles like Sphercarb (containing extensive micropore structures), modeling results suggest that surface diffusion is the main
transport mechanism for diffusion from small pores and that pore diffusion is insufficient to describe the desorption process.

Lin et al [50] investigated the adsorption and desorption of benzene, trichloroethylene, and water vapor onto sand and two dried soils (an EPA synthetic soil matrix and a Superfund site soil). A porous sphere model taking into account intrapore diffusion coupled with a nonlinear Freundlich isotherm described reasonably well the trends of both sorption processes using the effective diffusivity as the only fitting parameter. Surface diffusion was not incorporated into the model. A parallel pore model, taking into account diffusion in both small pores as well as the larger ones, was found to improve the fit to the experimental data, but the inferred tortuosity of the pores required for the fit was found to be too small. This could due to the lack of adsorbent penetration beyond the outer surface of the particle during the adsorption process, where the resulting contaminated particle is then used for the desorption experiments.

Soil Bed Studies

Wu et al. [10,32] at the New Jersey Institute of Technology (NJIT) have investigated mass transfer in soil. Soil used in their studies were obtained from the ground surface in Newark, New Jersey. Through deposition experiments on a packed bed of soil in a modified gas chromatograph, chromatographic response analysis was used to experimentally determine parameters relevant to mass transfer such as equilibrium constants for the partition of contaminants between the vapor phase and the adsorbed phase on the soil, heats of adsorption, axial dispersion coefficients, and intraparticle diffusion coefficients [10]. Model predictions using experimentally determined parameters were found to match experimental adsorption profiles fairly well. Later work by Wu and Bozzelli [32] showed that soil particle size does not affect the contaminant equilibrium constants and heats of adsorption.
Cabbar et. al [51] has measured the effective diffusivities and adsorption equilibrium constants for monochloroethane, 1,2-dichloroethane, 1,1,1-trichloroethane, and 1,1,2-trichloroethane diffusing axially through a cylindrical packed soil pellet. In these experiments, equal pressure is maintained on both sides of the pellet (to eliminate convective flux) before the contaminant is introduced as a pulse on one side of the pellet. An analysis of moments on the contaminant breakthrough profile vs. time were used to obtain the desired quantities.

Lighty et al. [33-36] at the University of Utah carried out a number of adsorption and desorption studies using a clay soil obtained as dust from brick manufacture. Three apparatus were used to investigate contaminant mass transfer: a "Particle Characterization Reactor" (PCR), a "Bed Characterization Reactor" (BCR), and a small scale rotary kiln. The PCR consists of a packed bed of soil through which a gas (contaminated or not) would pass, thereby reducing interparticle effects. The "Bed Characterization Reactor" (BCR), a bed of soil over which the decontaminating gas would flow, was used to analyze largely interparticle effects.

PCR desorption studies of four model soils showed that nonporous materials such as sand and glass beads desorb p-xylene much more quickly than do the porous materials, peat and clay [34]. The desorption of p-xylene from porous materials took place in two stages: an initial steep region which probably pertains to external surface evaporation of the contaminant, followed by a much slower and rate-controlling process which suggested intraparticle diffusion or desorption. Experiments showed that increasing temperature led to greater evolution rates in both the evaporation and diffusion-desorption regimes, and to lower amounts of contaminant remaining in the soil after treatment. These results suggested that adsorption-desorption phenomena are the probable rate limiting steps at long times in the contaminant desorption process from soil [34]. When the PCR was used to adsorb p-xylene from the gas phase, it was found that the contaminants adsorb according to the Freundlich isotherm [34,35].
Factors other than reactor temperature and soil type were also found to be important in PCR experiments at the Univ. of Utah. When examining the evolution rate of naphthalene and p-xylene from a gasoline contaminated soil, the lighter compound selectively desorbed first and the higher molecular weight compound took longer to desorb [34]. Soil moisture was found to increase contaminant desorption rates, probably due to steam distillation of the contaminant from the soil or due to water occupying sites contaminant otherwise would [52].

BCR data confirmed that local bed temperature affects desorption and total evolution of contaminant [35]. When PCR and BCR data were compared with those of the small scale rotary kiln, the contaminant evolution rate of the rotary kiln was between those of the PCR and BCR. These results illustrate the importance of heating rate and gas-solid contacting on the total evolution of p-xylene [35], as well as the possible importance of the solid-solid contacting expected from the rotation-induced tumbling of soil in an actual kiln.

Experimental soil temperature vs. time and p-xylene evolution vs. time profiles from the PCR and BCR experiments have been modeled [36]. Using the Freundlich isotherm, energy and mass conservation equations were applied simultaneously to the PCR assuming negligible radial gradients and axial dispersion. The resulting model predicted well the soil temperature-time profiles for runs at three different temperatures. The model was also reasonably able to predict the concentration of p-xylene remaining in the soil after 1 hour of desorption for temperatures between 25 and 275°C.

The BCR was modeled as a one-dimensional heat transfer and mass-transfer system [36]. The energy conservation equation was applied assuming negligible heat loss due to contaminant evaporation (contaminant level was 0.5% by weight). The temperature at the bottom of the soil bed was fit as a function of time to the experimentally measured profile and applied as a boundary condition to the bed. The model fit the temperature-time profiles measured well near the bottom of the bed, but underestimated the bed temperatures near the top of the bed. A one-dimensional mass transfer equation was fit to the boundary
conditions of no flux at the bottom, and zero concentration at the bed top (due to the sweep gas) The Freundlich isotherm was applied to describe contaminant concentration in the gas and soil phases of the bed. After the model was fit to p-xylene evolution data at 240°C, the model performed well at 315°C but overestimated the evolution rate at 175°C.

1.3.3 Rotary Kiln Incinerator Studies

Previous rotary kiln work has focused on modeling the mass and heat transfer characteristics leading to volatilization of the contaminant. The bulk of this work has been performed at University of Utah, Louisiana State University, and by the U.S. E.P.A..

Based on their modeling of the PCR and BCR data, models of a rotary kiln have been developed at the University of Utah [37-39]. Silcox et al. [37] have created a model for an indirectly fired rotary kiln. The model examined burden (or loaded soil) mixing, but did not incorporate gas phase fluid mechanics, contaminant evolution, or decomposition kinetics. The kiln was broken down into zones, in which heat and mass transfer equations are employed. Radiation and convection were included in the heat transfer calculations. The effects of soil moisture content, kiln rotation rate, fill fraction, and kiln temperature were examined with the model and compared with data obtained with a 130 kW natural gas-fired pilot-scale rotary kiln [38]. The model generally predicted the trends in the data for all four of the variables, but could not accurately predict the data for any of the variables except for kiln temperature.

To treat interphase transport effects, a model of toluene desorption from a montmorillonite clay adsorbent was developed by Owens et al. [39]. Heat and mass transfer in the soil bed in the kiln were modeled to describe the effects of rotation rate and fill fraction on the toluene desorption vs. time profile. Trends were fairly well predicted at 189°C and 332°C using three adjustable parameters. The parameters pertained to an Arrhenius-type expression used to describe local toluene desorption rates in the bed. The
Arrhenius type expression incorporated effects of slumping frequency (the frequency at which the soil is turned over in the kiln due to rotation) as well as fill fraction and rotation rate. However, this model could not be used for higher temperatures since the charring of toluene was not included in the model. A sensitivity analysis of this model showed that the local generation term controlled the overall desorption rate, suggesting rate controlling processes to be on the individual particle level.

The U. S. E. P. A. has also created computer models of the rotary kiln incinerator based on experiments performed with a pilot-scale rotary kiln reactor [40,41]. Linak et al. [40] studied the generation of transient puffs during batch treatment of sorbents laden with volatile organics. Formation of puffs was attributed to exposure of fresh sorbent surface area to the hot kiln environment as the contaminated sorbent bed underwent slumping motions and breakup caused by kiln rotation. Transient puff intensity increased with kiln temperature, rotation speed, and charge mass [40,41]. In combination with transient heat and mass transfer description for a heated bed of soil, Wendt and Linak [41] created a sorbent fragmentation model to describe puff behavior. A batch charge of contaminated sorbent was assumed to fracture into two aggregates after a characteristic time, with each aggregate further fracturing after subsequent characteristic times. Each fracture allows for release of contaminant as the sorbent surface area to volume ratio increases. The model provided qualitative agreement with experimental data. A factor leading to only qualitative agreement could be because the model does not include mixing of the soil layer by kiln action.

Wendt et al. [53] did further work to provide mechanistic grounds for observed profiles in puff formation at the outlet of the rotary kiln: coarse structure of puffs evolution over time was directly attributed to kiln rotation exposing new surface, while fine structure was probably due to turbulent gas phase mixing, indicating transport and not only vapor pressure driving forces control evolution of waste into the gas phase. Experimental kiln data were fit using same model above, but with slightly different boundary conditions and
with the assumption that the sorbent aggregate broke into four pieces only initially with the surface area being "renewed" after characteristic times. Again, the model does not include mixing due to kiln rotation. Using this model, Lemieux et al. [54] predicted optimal performance (minimizing transient puff formation) for a rotary kiln by operating at a low rotation rate and low temperature, with controlled oxygen enrichment such as with time-pulsed addition of oxygen.

Lemieux et al. [55] also showed that transient puffs in a rotary kiln can be significantly reduced using containers with small compartments which would individually break over time vs. a single large container which breaks open at once. The compartments serve to distribute exposure of contaminant rich sorbent over a longer period of time, thus minimizing sudden local depletion of oxygen which leads to transient puffs formation.

Chern and Bozzelli [56] at NJIT has used a bench-scale rotary kiln to study thermal desorption of 1-dodecene and 1-hexadecene from sand. Sand was utilized because of its lack of particle pore structure. A maximum removal efficiency was found to be 97.4% for test runs between 50 and 250°C for 6-20 minute residence times. Trends for removal efficiency (increases with temperature and residence time) were similar to those found in previous soil bed studies.

At Louisiana State University, field-scale rotary kiln studies of contaminant evolution have been performed. Cook et. al [57] examined dichloromethane evolution from sorbent packs as affected by kiln rotation rate and the presence of turbulence air. Slow kiln rotation rate yields contaminant evolution in distinct periods relating to the motion of the remnants of the pack, while faster rotation yields a more continuous evolution profile. Faster kiln rotation rates also appeared to give slightly better dichloromethane removal from the sorbent packs. Material balance uncertainties make the effects of turbulence air difficult to determine (the carbon balance closure with the turbulence air on is of order 70%, while a value of order 100% is obtained without turbulence air). Factors leading to material balance problems could be incomplete
dichloromethane evolution, conversion of the contaminant to PAH's which would not be measured in gas phase sampling, or assumptions leading to misleading kiln outlet estimates. Leger et. al [58,59] have followed up on Cook’s work using a different calculational technique, but were unable to determine further the effects of kiln rotation rate and presence of turbulence due to continued mass balance difficulties. However, they found that clay/polyethylene sorbent packs loaded with approximately 41 wt. % toluene evolved the middle 80% of material over an average time span of 141 seconds.

1.3.4 Soil Off-gas Emission Studies

Some preliminary work has focused on identifying specific compounds formed during the thermal treatment of soil. While treating a toluene contaminated soil in air, GC/MS of evolved gases indicated the presence of benzene, ethylbenzene, xylene, benzaldehyde, benzyl alcohol, quinone, benzonitrile, and several halogenated hydrocarbons [42,43]. The source of these chemicals is uncertain, particularly in the case of halogenated hydrocarbons since halogenated hydrocarbons were not used as a contaminant. McClennan et. al. [42] have verified that the detected compounds were not impurities in the solvent used to contaminate the soil. These products suggest degradation, oxidation, or other types of reactions with the soil substrate [43].

At Louisiana State University (LSU), total hydrocarbon, oxygen, CO, and CO₂ concentrations exiting an industrial scale kiln were examined when incinerating xylene-contaminated sorbent packs [60,61]. Other LSU studies using toluene/sorbent packs have tracked compounds leaving the kiln in the gas phase such as toluene, benzene, and lighter hydrocarbons up to C₂'s [62]. Researchers at the Univ. of Utah have also monitored these gases in a small-scale rotary kiln treating toluene-contaminated sorbents [63]. Several polycyclic aromatic hydrocarbons, chlorinated compounds, and dioxins have been detected.
in the off-gases from a rotary kiln simulator firing toluene and toluene-carbon tetrachloride mixtures by Linak et al. [40].

1.4 Summary

Public outcry over waste disposal mismanagement led to the passage of the Superfund Law in 1980, which organized a national program to identify and clean up the worst chemically contaminated sites in the United States. Since the Superfund program’s inception, approximately 1320 sites have been placed on the National Priority List (NPL) designating them to be highly threatening to the environment. Each site contains hundreds to thousands of tons of contaminated soil. Polycyclic aromatic hydrocarbons (PAH’s) have been found at a significant number of NPL sites at concentrations ranging from ppm levels to of order 5 wt.%. PAH’s are of public concern since 1) many PAH’s are mutagenic, carcinogenic or both, and 2) are among the more difficult organic compounds to remove from soil.

Incineration and related thermal destruction technologies have shown great applicability and versatility in handling organics-contaminated soils, while destruction or removal efficiencies (DRE’s) as high as 99.9999% have been demonstrated for specific hazardous constituents. Major problems facing incineration are the high cost (typically between $300 - $700 per ton of soil) and public opposition due to concerns over emissions during operation. It is believed that public confidence in incineration processes can be improved through better technical understanding of the complex set of chemical and physical processes responsible for incinerator behavior.

The majority of the studies of the thermal decontamination process has focused on extents of removal of contaminants from soil, adsorption onto and desorption from soil of the contaminant, and heat and mass transfer in soil beds and rotary kilns. There have been
a few studies examining the off-gases from soil being treated at industrially relevant conditions suggesting that chemical transformation of the contaminant may be taking place during the removal process. Here, a project has been proposed to 1) determine if there are treatment conditions where the contaminant is transformed to other products, 2) ascertain the role of soil in the decontamination process, and 3) understand how treatment conditions affect rates and extents of contaminant removal from soil.

1.5 References


Chapter 2

Apparatus and Procedure

2.0 Introduction

Chapter 1 having specified the conceptual focus of this work, Chapter 2 discusses the overall experimental program, justifies the selection of experimental conditions and parameters as well as the choice of apparatus, and lastly describes the experimental apparatus and procedures for soil sample treatment, volatiles collection, and volatiles analysis.

2.1 Experimental Approach

2.1.1 Overall Experimental Program

A modified screen heater reactor was used to heat samples of an EPA Synthetic Soil Matrix (neat or artificially contaminated with 4.75 wt. % pyrene) in an inert bath gas to study the first several seconds of thermal soil treatment. Light gases and tars volatilized from the treated samples were captured, the light gases (CO, CO₂, CH₄, C₂H₂, C₂H₄, and C₂H₆) were analyzed, and weight loss was monitored to estimate extents of contaminant removal. The temperature study consisted of heating both neat and contaminated soil samples to final temperatures between 400 and 1000°C at a nominal heating rate of 1000°C/s. In the heating rate study, soil samples were heated at nominal heating rates between 100 and 5000°C/s to a nominal temperature of 550°C. By studying weight loss and the quantities of volatiles produced as affected by temperature and heating rate, insight
into the fundamental processes and chemistry taking place during thermal treatment were to be elucidated.

2.1.2 Selection of Experimental Parameters

A number of variables could potentially affect rates and extents of contaminant removal as well as the quantities and identities of the organics volatilized from a contaminated soil when thermally treated. Before beginning experiments, decisions about four sets of parameters were made: thermal treatment equipment, soil and soil conditions, contaminant and contaminant level, and lastly, thermal treatment conditions.

Equipment

The experimental apparatus used in this work is a modified screen heater reactor (described in detail in Section 2.2). This reactor heats a thin layer of soil on a hot stage heated by electrical resistance, where volatilized products can be collected in a series of traps connected to the reactor effluent. A thin layer of soil was thought to be a good compromise between the very low quantities of volatilized materials available from single soil particle experiments and the added heat and mass transfer effects present with thick soil beds (e.g., volatilized contaminant from a single soil particle readsorbing onto neighboring particles).

Experimental work by Darivakis [1] showed that soil (63 - 90 μm particle size) treated under these constraints can lead to complete removal of the contaminant in short treatment times (of order 1 second). The treatment temperature-time profiles used by Darivakis were zero holding time experiments with a peak temperature between 200 - 800°C at a heatup rate of 1000°C/s. The results (i.e., soil weight loss corrected for weight loss from uncontaminated soil) indicated that pyrene was completely removed (in a helium environment) at temperatures greater than 600°C. The screen heater has been used
previously at MIT by several investigators to study pyrolysis of coal (e.g., Anthony [2], Suuberg [3]) and biomass (Hajaligol [4]) and soil decontamination (Darivakis [1] and Koch [5]). The screen heater reactor has established itself as a means to gather quantitative pyrolysis data. Mass as well as elemental carbon and hydrogen balances have been closed to within 5%.

Helium was selected as the gas atmosphere in which to conduct our thermal treatment experiments for two main reasons: 1) an atmosphere with oxygen would oxidize the hot stage interfering with the weight loss measurements, and 2) a pyrolytic environment simplifies the analysis of possible chemical pathways leading to the formation of observed products.

For rapid heating, the selection of this reactor, however, does fix the time domain to the first several seconds of thermal treatment, due to equipment constraints. But the study of this time domain is not necessarily a disadvantage considering that a significant portion of soil contaminant is volatilized as the soil temperature increases to that of a kiln, and that this time domain has not been extensively studied by previous workers. High heatup rates are likely to be experienced by soil particles below ~1 mm diameter upon entering a high temperature thermal system, and represent the more severe heating conditions likely to be encountered by contaminated soil in thermal treatment systems. This reactor is ideal for studying this time region due to its ability to achieve high temperatures at very high heating rates and to maintain temperatures for a period of several seconds or more.

Soil

At the beginning of this project, there were four soils available for use: University of Utah Soil, NJIT (New Jersey Institute of Technology) Soil, a commercial potting soil (Hyponex brand), and an EPA Synthetic Soil Matrix. The Utah soil is a clay dust obtained from a brick factory [5]. The NJIT soil is a Newark, NJ ground surface soil that was
washed in tap water and baked at 200°C overnight [6,7]. The Hyponex-brand potting soil is a commercial soil purchased at a local hardware store. The EPA soil is a formulation created to serve as a basis soil to evaluate cleanup technologies, based on an average of the soil compositions found at sites relating to 86 ROD’s and an independent study of the composition of Eastern U.S. soils [8,9].

At the time, no published literature was found concerning thermal decontamination or pyrolysis of potting soil. Some preliminary weight loss experiments and commercial organic carbon analysis revealed that the bulk of the weight loss was moisture and that the organic carbon content may be too high to represent typical soils found at Superfund sites. The Utah and NJIT soils, which were used in studies at University of Utah and New Jersey Institute of Technology described in Chapter 1, were obtained locally, but have not been used extensively in studies outside their institutions. Because of the potential future and wider use of the EPA Soil and the basis for its composition, it was thought that data based on the EPA Synthetic Soil Matrix would have greater industrial relevance.

Small particle sizes (63 - 125 μm) were chosen to minimize thermal and mass transfer gradients within soil particles. See Appendix D for a calculation showing that soil particles are spatially isothermal at experimental conditions.

Lighty et al. [10,11] have shown that soil moisture content is an important parameter in decontamination, proposing that the presence of moisture serves to steam distill the contaminant from the soil during heat treatment. Since a soil with a stable moisture content is needed to minimize this effect, desiccated soil was chosen as the basis since ambient relative humidity in the laboratory can vary widely from day to day. Using Drierite® (anhydrous calcium sulfate) as the desiccant, soil was dried for two days in a desiccator prior to use in experiments. The use of dry soil had the added advantage of minimizing the contribution of hydrolysis reactions during thermal treatment [12].
Contaminant

As mentioned in Chapter 1, pyrene was chosen as a model contaminant because it is one of the higher MW PAH often present in heavy oils, tars, and some sludges at hazardous waste sites. A single hydrocarbon contaminant was chosen initially in hopes that chemical analyses would be performed on the tars, should time permit. Pyrene pyrolysis products are known and have been analyzed previously at MIT (discussed in Chapter 4). Similar to the study by McClennen et. al [13,14] where a soil containing toluene was heated and produced chlorinated and oxygenated products, the possible reaction of the contaminant with the soil would be more easily ascertained with a simple hydrocarbon contaminant. If there were extensive contaminant-soil chemical interactions, a single hydrocarbon contaminant would also reduce the number of possible pyrolytic products to be analyzed. Experimentally, another advantage of having selected pyrene is its very low vapor pressure at room temperature, so vaporization losses would be negligible during the reactor flushing steps of the experiment.

The pyrene contamination level of 4.75% was chosen since five wt.% is representative of the higher range of organic contaminant loadings at Superfund sites (discussed in Chapter 1), and a high initial pyrene loading would provide enough material to facilitate quantification of decontamination efficiencies and by-product yields by weight loss measurements.

Thermal Treatment Conditions

Since auxiliary fuel is an important cost issue in thermal remediation technologies, the required maximum operating temperature and minimum required treatment time for destruction/removal will strongly affect the economics of the process. As noted previously, these parameters are also important variables in determining when various products are formed and volatilized.
The soil treatment conditions attainable with the modified screen heater reactor setup used in this study are summarized in Table 2-1:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Equipment Maximum Range</th>
<th>Experimental Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>heatup rates</td>
<td>100 - 6000 °C/s</td>
<td>100 - 5000°C/s</td>
</tr>
<tr>
<td>max. soil temperature</td>
<td>30 - 1200 °C</td>
<td>400 - 1000°C</td>
</tr>
<tr>
<td>heating times (heatup + holding times)</td>
<td>0.05 - 25.00* s</td>
<td>0.4 - 6 s</td>
</tr>
</tbody>
</table>

* maximum permitted holding time decreases with increasing temperature

The maximum heating rate and heating times were set by the maximum power that can pass through the variacs controlling power into the reactor system without overheating them, while maximum temperature was set by the foil’s ability to maintain structural integrity. The lower heating rate available is limited by the experimenter’s ability to manually control heatup to an approximately linear profile. In the experiments, the maximum values for the three parameters were used with a conservative margin for safety, while the minimum temperature (and hence heating time) was set by the boiling point of pyrene (404°C).

These considerations also reflect practical operating conditions. Heatup rates expected to be encountered by soil particles can vary from of order 10°C/min. to 10000°C/s, where the selected experimental heating rates are in the higher end of the expected range. The boiling point of the contaminant is often selected as a minimum operating temperature for thermal desorption processes, and 1200°C is close to the upper range of typical exit temperatures (1300°C) quoted by Linak et al. [15] for full-scale rotary kilns, while 100°C would go slightly below most thermal decontamination studies for desorbing volatile organic compounds. Heating times of 20 seconds are several times kiln gas residence times of 1-3 seconds [15] and are expected to be sufficient for virtually
complete release of contaminant from soil of the particle sizes to be studied. Pyrolysis studies using this reactor where particles were heated for times as short as 0.05 seconds have been performed previously.

Other considerations

It was necessary to perform experiments with artificially contaminated soils (soil spiked with contaminant) instead of a real site sample. Due to potential interactions between soil constituents and the contaminant, separate runs of the clean and contaminated soils are needed to identify products inherent to the soil itself in addition to the contaminant. To simulate industrial thermal remediation processes, the soil was not treated to remove any of the organics contained in the soil.

The last consideration was the length of time the contaminant was allowed to sorb to the soil. The Utah workers also performed aging studies, where a pre-weighed soil sample was contaminated with p-xylene to 0.5 wt.% and the resulting mixture was stored for periods up to one year [16]. Upon comparing contaminant desorption curves from the PCR (described in Chapter 1) of samples that were aged for one day, six months, and one year, increasing exposure time appeared to slightly lower the contaminant desorption rates. However, the difference was found to be sufficiently small to justify selecting one day as the aging period. Also due to practical constraints, a one day aging period was selected for soils contaminated in this study.

2.2 Apparatus Description

Soil samples were heated in a modified electrical screen heater reactor (Figure 2-1). The reactor consists of a 9” dia. x 12” tall cylindrical Pyrex chamber, which is sealed at both ends by aluminum flanges. Hanging from the reactor top flange are two electrodes,
Figure 2-1. Schematic of the modified electrical screen heater reactor: a) reactor from top view, b) reactor from side view, c) closeup of the molded hot stage.
between which a hot stage heated by electrical resistance is mounted between clamps attached to the lower part of each electrode. Soil temperature-time histories are measured throughout each experiment, using a rapid response Type K thermocouple (0.0005” thick film “Cement-On” Thermocouple, Type II, Omega Engineering Inc.) placed beneath the hot stage. Heat transfer calculations imply that the thermocouple closely tracks the bulk mean temperature of the soil particles (e.g., conservatively to within about 30°C at most, at a nominal heating rate of 1000°C/s - see Appendix D for the calculation). The top flange of the reactor is raised and lowered to introduce and remove the soil samples. Detailed information regarding the reactor construction/design and the reactor electrical circuit diagram can be found elsewhere (Caron [17]).

Previously, this apparatus was used to study pyrolysis of coal [2,3] and biomass [4] where samples to be heated were contained within a folded stainless steel screen. Screens have been used in this reactor in the past as they allow for gas flow past the particles held in the screen. The gas flow facilitates separation, dilution, and cooling of newly evolved volatiles. However Darivakis [1] had difficulty containing soil in the screen due to soil particle breakup. To circumvent this problem, Koch [5] devised a molded metal sheet to replace the screen. In the present work, a molded stainless steel foil (1\(\frac{3}{8}\) x 5\(\frac{1}{8}\) x 0.001”, 302 stainless steel shim also shown in Figure 2-1) serves as the hot stage, confining the sample within a 0.79” I. D. x 0.11” deep cylindrical well. The well was created by pressing the stainless steel foil between a nut and a piece of pipe. This change to the molded steel foil led to better reproducibility of soil mass loss measurements.

With this equipment, thermally thin beds of solids can be subjected to independent variations in heating rate (100 - 6000°C/s), final temperature (300 - 1100°C), and time at final temperature (0 - 30 seconds), under inert or reactive gaseous atmospheres. Typical temperature-time histories are shown in Figure 2-2. The sample was cooled by natural convection and/or radiation to room temperature at initial heating rates of order 400 - 600°C/s [18]. The temperature profiles are obtained by electric power input controlled by
Figure 2-2. Schematic representations of temperature-time profiles obtainable with the electrical screen heater, captive sample reactor.
two variacs connected in series and electronic timers. As the screen is heated up, a single
variac is used to regulate power through the electrodes. To hold the screen temperature, the
second variac is then added in series to scale back power output from the first variac, since
less power is required to maintain the final temperature than for heatup. The times for both
the heatup and hold steps are set by electronic timers.

Ports in the reactor allow gas to be added and removed from the reactor chamber as
the metal sheet is being heated. Volatiles from the soil are readily cooled, diluted, and
separated from the high temperature zone surrounding the foil, thus reducing opportunities
for secondary reactions and for changes in the volatiles composition. Should secondary
reactions on soil be very important, the screen heater reactor also has a liquid nitrogen
quench option. This option (not used here) allows for liquid nitrogen to be poured directly
onto the metal foil to quickly cool it after the metal foil has attained the desired final
temperature.

As was done by Franklin [19], material volatilized from the soil can be collected by
purging the reactor gas through three series-connected traps (Figure 2-3), respectively
designed to collect tars, condensables (benzenes to naphthalene), and light gases (CO,
CO₂, and C₁-C₂ hydrocarbons). The tar filter consists of a 25 mm dia. 1 µm pore size
teflon filter paper (Costar® Corp. Filinert™ Membrane Filter, Cat. No. 130610) contained
within a stainless steel casing (Costar® Corp. 304 Stainless Steel 25 mm dia. Syringe
Holder, Cat. No. 421500). The condensables trap is made up of a 6 mm O. D. Pyrex tube
containing 1.5” each of 3% OV-17 on GasChrom Q (80/100 mesh) and Porapak Q (50/80
mesh) gas chromatographic packings to capture low boiling liquids. The light gas trap
consists of 15” of Porapak Q within a 1/4” O. D. stainless steel tubing. The reactor,
condensables and light gas traps are connected together using Swagelock® Quick-
connects, which allow attachment and detachment of the traps from the reactor system with
minimal air contamination.
Figure 2-3. Schematic of Reactor System
These tars, condensables, and light gas traps were tested for capture efficiencies. The collection efficiency of the light gas trap was tested by injecting a known amount of methane into the reactor during reactor purging, collecting the methane in the trap, and analyzing the captured gas by gas chromatography. The light gas trap collection efficiency was found to be 99%. The reactor purge consisted of a continuous helium flow through the reactor and traps while maintaining reactor pressure at 3 psig, with total gas throughput of approximately 5 reactor volumes at 0.5 l/min. like in the actual experimental runs. The tar filter (99% efficient) was tested by volatilizing a known amount of pyrene, collecting the material in the filter, and comparing the amount collected to the quantity of pyrene available for capture (initial minus the sum of the pyrene unvaporized and the pyrene lost to the reactor walls).

The condensables trap (used at room temperature) capture efficiency was found to be approximately 35%, by injecting a known volume of toluene just upstream of the trap and comparing the weight captured to the amount injected after passing 5 reactor volumes through the trap. If used at low temperatures (-40°C), capture efficiency of the condensables trap can increase to as much as 90% but at the cost of large scatter in the data likely caused by ambient moisture entering the trap during detachment from the reactor system at the end of each run. Based on some information provided by previous investigators, capture efficiencies appear to have been low for traps of this kind. For a trap filled with glass wool operated at -77°C, Hajaligol [4] reported low capture efficiencies between 10 and 40% for oxygenated organics such as methanol, acetone, ethanol, and acetaldehyde. Suuberg [3] observed that aromatics falling between benzene and naphthalene can be found in his condensables traps (same construction as the ones used here), but retention of the more volatile condensables like benzene is poor. As the quantity of condensables produced in these experiments was expected to be low, the low efficiency of the condensables trap was thought to be only a minor problem.
To minimize gaseous impurities entering the reactor, the helium used in the experiment was filtered through a prefilter consisting of 15" of Porapak QS (50/80 mesh) within a 3/8" O. D. stainless steel tube. The helium prefilter and light gas traps are immersed in Dewar flasks containing liquid nitrogen (77 K) during the experiment, while the condensables trap is operated at room temperature. A funnel constructed from heavy duty culinary aluminum foil was attached to the tar filter inlet to help direct volatiles flow into the traps. The inner surface of the reactor top and bottom flanges was covered with regular culinary aluminum foil to collect tar condensate.

A Bascom-Turner Instruments Model 4120 Strip Chart Recorder/Data Acquisition System used to collect and store the temperature-time profiles experienced by the soil layers. Temperatures were typically sampled by this system at 10 msec intervals at nominal heating rates of 1000°C/s and above, while 100 msec intervals were used for lower heating rates.

It is believed that the overall product collection protocols are quite satisfactory in that materials accountability (mass balances) of 95% has been closed. Mass balances for each experiment are tabulated in Appendix B.

2.3 Experimental Procedure

This section discusses briefly the procedure to be followed for 1) soil pretreatment, 2) soil contamination, 3) each experimental run, and 4) analysis of the pyrolysis products. This procedure was developed to allow: 1) a carbon balance to be performed by comparing the weight loss from the soil to the quantities of products obtained from the three traps, 2) the resulting treated soil to be available for extraction with an organic solvent to obtain organics formed and not vaporized during the heating process, and 3) extraction of organics in the trapped samples for chemical analysis by gas chromatography.
Soil Pretreatment

All soils used were preconditioned by drying over Drierite® (anhydrous calcium sulfate) in a desiccator for two days. In the case of contaminated soil, the soil to be contaminated was preconditioned, contaminated using the method described below, and then redried in the dessicator for two days prior to use. As mentioned above, the purpose of pretreatment was to minimize the changes in the soil moisture content due to changes in ambient conditions (e.g., humidity), either through loss or gain of moisture from the ambient atmosphere. Pretreatment was found to stabilize the soil moisture content to approximately 1 wt. % moisture.

Soil Contamination Method

As pyrene is a solid at room temperature, it was dissolved in a volatile organic solvent to allow intimate mixing with the soil. After allowing the pyrene to sorb onto the soil, the solvent must then be evaporated to avoid it contaminating the soil.

The technique used is a refinement of that developed by Darivakis [1], where he added a known solution of pyrene in methylene chloride to a thin layer of soil in a shallow aluminum pan and allowed the solvent to evaporate overnight. Darivakis ran into two difficulties with his technique: excess contacting of the contaminant-laden solvent to the soil container walls led to pyrene crystallization on the walls, and rapid evaporation of the solvent resulted in large pyrene crystals unadsorbed onto soil particles. The first problem was solved by using an inverted cylindrical weighing jar, where solvent condensation on the walls followed by gravity-induced flow back to the main reservoir “washed” pyrene from the lower part of the walls of the container. Slowing the methylene chloride evaporation by placing the open weighing jar within an inverted jar whose lid was coated with teflon (Figure 2-4) solved the second difficulty. This created a significant back
Figure 2-4. Soil contamination equipment setup
pressure of methylene chloride so that the rate of evaporation was largely controlled by slow diffusion of the solvent out through the lid.

Based on these refinements, the “best” method found to add pyrene uniformly to the EPA synthetic soil was to:

1) place 1.5 g of the soil in an unbroken thin layer in the cap of an inverted cylindrical weighing bottle (Kimax, 45 ml with 45/12 ground glass joint)
2) pipette gently (drop by drop) the contaminant-laden solvent to eventually create a liquid layer approximately 1-2 mm above the soil layer
3) seal the weighing bottle for 12 hours to allow the soil time to adsorb pyrene from the solvent
4) open the weighing bottle, and gently move the weighing bottle cap containing the soil onto a specially-treated cap of an inverted 9 oz. wide-mouth jar (taking care not to disturb either the soil or liquid layers), and seal the jar
5) allow the solvent to evaporate slowly over the next 8-10 hours

The cap of the 9 oz. wide-mouth jar was sprayed three times with Fluoroglidle™ (Norton Performance Plastics Corp.) before use to create a teflon lining to slow the diffusion of volatilized methylene chloride through the jar lid. To obtain the 4.75 wt. % contamination level used in this work, the solution ultimately used to introduce pyrene was a 3.29 wt.% pyrene solution in methylene chloride made from 75 mg pyrene and 1.66 ml methylene chloride. As pyrene tends to crystallize on any surface from a highly concentrated solution, half of the methylene chloride (~0.83 ml) was used to dissolve the 75 mg pyrene in a graduated centrifuge tube. This solution was pipetted onto the soil, followed by a rinse of the centrifuge tube with the other half of the required methylene chloride which was also pipetted gently onto the soil.
The actual contamination level of the soil was verified by extracting a known weight of the contaminated soil with methylene chloride. The resulting filtrate was then allowed to evaporate in a preweighed aluminum pan, and the remaining pyrene was weighed. The contamination level was within 2% relative (4.75% ± 0.10%) of the target.

**Experimental Run Procedure**

A typical run involved molding the foil, loading the soil sample into the reactor, flushing the reactor, heating the soil, and purging the reactor. The foil, prior to molding, was fixed to a peak temperature above 900°C in a helium environment to remove organics from the foil surface. The cylindrical well-like depression in the center of the foil (0.79" I. D. x 0.11" deep) was formed by pressing the foil between two pipe fittings. Soil (50 mg ± 3 mg) was weighed onto the foil which was then mounted between the electrodes. Soil was manually spread (by agitation of the foil, or by a pair of spatulas if necessary) into approximately a monolayer in the well. The tar filter, the tar filter funnel, and the condensables trap were then connected to or mounted within the reactor (preweighed aluminum foil top and bottom reactor liners were mounted in the reactor prior to loading the soil into the reactor). The reactor was then closed and flushed seven times with prefilled helium. A flush involves filling the reactor to 7 psig, and then evacuating the reactor to -30 in Hg using a mechanical vacuum pump. Liquid nitrogen was added to the light gas trap Dewar, and an equilibrium helium flow of 0.5 l/min. was then established at a reactor pressure of 3 psig of helium. This slight overpressure was used to prevent oxygen from leaking into the system by pressure-driven flow.

Each soil sample was heated through a preselected temperature-time history. The helium flow was continued for 2 hours after cessation of heating to collect materials volatilized from the soil (the cumulative helium purge, i.e. from after the heating of the soil to cessation of the helium flow, was equivalent to approximately 5 reactor volumes). The light gas trap was then removed from the reactor system and set aside for later analysis.
The foil and heat-treated soil, as well as the other traps, the liners, and the funnel were then removed from the reactor and weighed for mass gain/loss for material balance closure. Also included in the material balance was the yield of tars on the Pyrex reactor wall, estimated gravimetrically by wiping the surface with preweighed glass filters.

Prior to each run, the light gas and the helium prefilter traps are preconditioned for at least 30 minutes by connecting them to the reactor system and placing them both in boiling water while the reactor system is pumped down to -30 in Hg vacuum. Based on the experience of Griffin [20], the combination of temperature and low pressure is expected to efficiently remove adsorbed materials remaining in the traps from the previous run.

For a detailed step-by-step procedure for a typical experiment, see Appendix A.

Reactor Products Analysis

Light Gas Analysis: The light gas trap was connected to a Hewlett Packard 5890A Series II Gas Chromatograph (GC) via a six-port gas-actuated valve located inside the GC (Figure 2-5). With the valve in the OFF position, the light gas trap is heated in a separate oven at 110°C for 1 hour to desorb the captured gases from the internal packing. Then carrier gas is sent through the trap (valve ON) to transfer the trapped gases onto the head of the GC column. The valve is returned to the OFF position after 5 minutes.

Light gas analysis was performed on a 6' x 1/8" x 0.085" stainless steel, 80/100 mesh Carbosphere packed column. The column exit was connected to a thermal conductivity detector (TCD) followed by a flame ionization detector (FID). CO₂ and CO were quantified using the TCD and the combustible C₁-C₂ hydrocarbons by FID. The helium carrier gas flow rate was 10 ml/min., with a TCD reference flow rate of 15 ml/min. The FID gas flows were set according to Hewlett-Packard's specifications. The temperature program used for separation of the gases was as follows:

- -30°C initially for 35 minutes
Figure 2-5. Gas chromatograph configuration for light gas analysis
• 25°C/min. ramp to 115°C
• 115°C for 15 minutes
• 25°C/min. ramp to 150°C
• 150°C for 30 minutes

After every two or three runs, the Carbosphere column was baked out at 300°C for one hour to desorb heavier compounds sometimes collected in the light gases trap.

**Soil Extract and Tars Analysis.** Due to time constraints, the tars collected (by the tar filter, tar filter funnel, top and bottom liners and the glass filters) and extracts of the treated soil were not analyzed in this work. It was hoped that sufficient quantities of these samples could be generated for chemical analysis as well as for biotransformation studies to investigate production of carcinogenic or mutagenic compounds during the treatment process. These studies are being left for future work.

### 2.4 References


Chapter 3

Experimental Data

3.0 Introduction

This chapter 1) presents the weight loss, tar, and light gas data collected from the temperature and heating rate studies, where neat EPA Soil as well as EPA soil contaminated with 4.75 wt.% pyrene were thermally treated, and 2) suggests possible reaction pathways or sources leading to the experimental products and observations.

3.1 Temperature Study

3.1.1 Experimental Data

Effects of temperature on total weight loss, tar and light gas yields (Figures 3-1 to 3-8) were studied by heating neat or contaminated soil from room temperature to final temperatures between about 350°C to 1050°C. The nominal heating rate was 1000°C/s and there was no holding time at the final temperature. Cooling was by radiation and natural convection at an estimated initial rate of order 500 - 600°C/s. Unlike the laboratory experiments where the total sample of soil was about 50 ± 3 mg for both neat and contaminated soil runs, neat soil data are plotted on the basis of 50 mg of contaminant-free EPA soil, while contaminated soil data are plotted on the basis of 50 mg neat soil + 2.49 mg pyrene before the onset of heating (leading to a contaminated soil where pyrene accounts for 4.75 wt.% of the total soil mixture). The error bars in the figures are standard deviations of the mean calculated in cases where duplicate experiments were performed.
Figure 3-1. Mass loss data for neat (O) and 4.75 wt. % pyrene-contaminated soil (●) as affected by peak temperature (nominal heating rate = 1000°C/s, zero holding time). 50 mg initial soil mass for neat soil runs and 52.49 mg (50 mg neat soil + 2.49 mg pyrene) initial soil mass for contaminated soil runs. Curves are freehand drawn trendlines.
**Figure 3-2.** Effect of peak temperature on implied pyrene removal determined as the mass loss difference between neat and contaminated soil, assuming no interaction between pyrene and soil (nominal heating rate = 1000°C/s, zero holding time).
Figure 3-3. Mass of volatilized tars for neat (O) and 4.75 w. % pyrene-contaminated soil (●) as affected by peak temperature (nominal heating rate = 1000°C/s, zero holding time). Curves are freehand drawn trendlines.
Figure 3-4. Carbon monoxide release for neat (O) and 4.75 wt. % pyrene-contaminated soil (●) as affected by peak temperature (nominal heating rate = 1000°C/s, zero holding time). Curves are freehand drawn trendlines.
Figure 3-5. Carbon dioxide release for neat (O) and 4.75 wt. % pyrene-contaminated soil (●) as affected by peak temperature (nominal heating rate = 1000°C/s, zero holding time). Curves are freehand drawn trendlines.
Figure 3-6. Methane release for neat (O) and 4.75 wt. % pyrene-contaminated soil (●) as affected by peak temperature (nominal heating rate = 1000°C/s, zero holding time). Curves are freehand drawn trendlines.
Figure 3-7. Acetylene release for neat (O) and 4.75 wt. % pyrene-contaminated soil (●) as affected by peak temperature (nominal heating rate = 1000°C/s, zero holding time). Curves are freehand drawn trendlines.
Figure 3-8. Ethylene release for neat (O) and 4.75 wt. % pyrene-contaminated soil (■) as affected by peak temperature (nominal heating rate = 1000°C/s, zero holding time). Curves are freehand drawn trendlines.
Data points were grouped, or considered "duplicate", on the basis of similar peak temperatures and when the deviations in the data points were judged not to be attributable to temperature effects (see Appendix B).

Figure 3-1 shows weight loss from neat and contaminated soils. Assuming no interaction between the pyrene and the soil, the difference in the weight loss curves for contaminated and neat soil can be taken as the weight of pyrene removed from the soil with increasing temperature. On this basis, Figure 3-2 indicates that pyrene is completely removed from the soil above 530°C, but also that above this temperature weight loss from contaminated soil exceeds the sum of the control weight of pyrene plus the corresponding weight loss from neat soil. A plausible explanation is presented in the Discussion section. Figure 3-3 shows the yield of tars, which for contaminated soil may include pyrene. An overall increase in tar release with increasing treatment severity is indicated, but difficult to confirm given the data scatter. Tars collected from experiments on contaminated soil were of different color at different heating severities: white (which is also the color of pyrene) in experiments below 500°C, yellow-tinged pink between 500-750°C, and brown above 750°C. These color differences suggest, but do not prove, that higher severity heating may cause chemical transformation of some of the pyrene during its release from the soil. Follow-up work will test this hypothesis by detailed chemical analysis of product tars from the different heating severities.

Carbon monoxide yields (Figure 3-4) are about the same for neat and contaminated soils up to about 450°C, but are somewhat larger for contaminated soil from 500 - 700°C and significantly greater above 800°C. Little or no difference is apparent in yields of carbon dioxide (Figure 3-5), methane (Figure 3-6), acetylene (Figure 3-7), or ethylene (Figure 3-8) from neat and contaminated soils. Ethane yields exhibited so much scatter that no systematic trends with temperature or with the presence of contaminant could be readily discerned. No significant amount of condensables (benezene to naphthalene) were collected in any of the neat or contaminated soil runs.
3.1.2 Discussion

Figure 3-2 indicates that pyrene is completely removed from the soil at temperatures above 530°C. This observation is consistent with results from similar experiments by Darivakis [1] using a pyrene-contaminated clay soil, where essentially total pyrene removal based on weight loss measurements was indicated at approximately 600°C at a nominal heating rate of 1000°C/s. Figure 3-2 implies that despite the short time available during heatup under rapid heating conditions, a major fraction of pyrene (~75%) can be removed by heating at 1000°C/s to roughly the boiling point of pyrene (404°C), and that essentially complete removal of the pyrene at this heating rate can be achieved by continuing heating to 530°C. However, from the following discussion, it will become clear that the decontamination process involves more than simple evaporation.

Figure 3-2 also shows that above 530°C, the implied pyrene removal exceeds 100%. Since this is impossible, the soil itself must be undergoing additional weight loss when heated in the presence of pyrene. Production of CO by reaction(s) of pyrene or pyrene-derived species with oxygen from the soil may contribute to this behavior. In support of this hypothesis: (1) CO yields are slightly larger for contaminated soil vs. neat soil between temperatures of about 550 to 650°C and dramatically larger at temperatures above about 850°C (Figure 3-3); and (2) the excess CO at ~950°C would account for approximately 2/3 of the excess weight loss. The existence of contaminant-soil interactions is also suggested by McClennen et. al.’s observations of oxygenated, halogenated, and nitrogenated species in volatiles generated by heating thick soil beds (of order 5 - 8 cm) artificially contaminated with alkylbenzenes like toluene and ethylbenzene [2,3]. Since their experiments were conducted in air, atmospheric oxygen may have contributed to oxygen in the volatiles, but it seems certain that chemical reactions between soil and
contaminant, or their decomposition products, must have been the source of the nitrogen- and halogen-containing volatiles.

A possible source of the CO produced from neat soil at high temperatures (Figure 3-4) is the reaction of CO_2 from the thermal decomposition of soil mineral carbonates with carbon from soil organic matter, via the reactions:

\[ \text{MCO}_3 \rightleftharpoons \text{MO} + \text{CO}_2 \]  
(1)

\[ \text{CO}_2 + \text{C} \rightarrow 2 \text{CO} \]  
(2)

where M is a metal. The 3.7 wt.% carbonate carbon present in the soil (which if completely calcined via reaction (1) would produce ~6800 μg of CO_2 per 50 mg uncontaminated soil) suggests an ample supply of CO_2 for reaction (2). Reaction of 90% of the soil organic carbon via reaction (2) would be sufficient to account for the maximum CO experimentally observed in the 1020°C run (Figure 3-4). Literature and thermodynamic data suggest the occurrence of this reaction sequence at high temperatures. Satterfield and Feakes [4] report that the presence of graphite or amorphous carbon facilitated barium carbonate decomposition through reactions (1) and (2). Production of CO in this manner is well known in fuel-rich zones of refractory-lined steel kilns fueled by coke or anthracite [5]. It is not clear whether reaction (2) contributes to the additional CO (Figure 3-4) observed from contaminated soil, e.g. by reactions of CO_2 with pyrene-derived carbon. This is because CO_2 yields are similar for neat and contaminated soil. For contaminated soil, further decomposition of the mineral carbonates via reaction (1) could buffer the local CO_2 concentrations within and adjacent to the soil, thereby compensating for CO_2 consumed by pyrene-derived carbon in reaction (2) and maintaining the net observed CO_2 production comparable to that for neat soil.

Further experiments are needed to determine the contributions of these hypothesized sources to the observed CO and CO_2 yields (e.g. mineral carbonate assays on samples of thermally treated soils to determine by difference the contribution of these minerals to CO_2 production). However, twenty or more runs would be needed to generate sufficient sample
for commercial assay. An in-house micro-analysis method for carbonate determinations was investigated but could not be implemented because of the imprecision of our technique and the deterioration of our gas chromatograph's detector caused by the acids used in the analysis. Another diagnostic experiment would be to repeat the above experiments using pyrene and mineral carbonates labeled with carbon-13 and then to assay for $^{13}$CO and $^{13}$CO$_2$ in the volatiles evolved upon heating the soil.

Thermal decomposition of pyrene may contribute to tar production from contaminated soil. Sharkey et. al [6] pyrolyzed samples of pyrene for four hours at 450°C in sealed glass capillary tubes that were evacuated prior to heating. The main products in the solid residue were believed to be dihydropyrene and dipyrenyl, while light gases such as hydrogen, methane, ethylene, ethane, and higher hydrocarbon gases were identified. At more severe conditions (900-1200°C), Mukherjee et. al [7] pyrolyzed pyrene using a drop tube furnace and found a variety of polycyclic aromatic hydrocarbons in the tars including bipyrenes, triphenylene, cyclopenta[cd]pyrene (CEP), cyclopenta[hi]acephenanthrylene (CPAP), and benzo[g,h,i]fluoranthene. Interestingly CPEP, which is a potent mutagen to bacterial cells [8] and human cells [9] in-vitro, is yellow-orange. While these experimental temperatures are much greater than the temperatures at which tars color changes are observed (530°C from white to yellow-pink and 750°C from pink to brown), it is possible that soil minerals might catalyze some pyrene conversion to CPEP. Detailed chemical analyses of the tar fractions are needed to test this hypothesis. Nevertheless, the observations of tar color changes, and of carbon monoxide yields exceeding what could be obtained from native organic matter in the soil, strongly suggest that secondary reactions of pyrene or its decomposition products with soil components or their decomposition products occur at rather modest treatment temperatures (> ~550°C). Further, the pyrene pyrolysis literature suggests that bioactive PAH could result with the proper conditions.

Thus while the details of the responsible chemistry are not known, there are indications that above a certain temperature, e.g. ~500°C for a heating rate of 1000°C/s,
polycyclic aromatic contaminants (or their decomposition products) will react chemically with soil or soil decomposition products to modify volatiles compositions, and that contaminant-soil interactions are sufficient to cause additional weight loss of the soil itself, i.e. the observed weight losses exceed those expected by adding the initial weight of contaminant to the weight loss of neat soil at the same heating conditions. Thus design, monitoring, control, and source apportionment protocols directed at technologies for thermal treatment of PAH-contaminated soils should consider the possibility of undesired side reactions of the contaminant.

3.2 Heating Rate Study

3.2.1 Experimental Data

Effects of heating rate on total weight loss, and on tar and light gas yields were studied by heating soil from room temperature to approximately 550°C at three different nominal heating rates of 100°C/s, 1000°C/s, and 5000°C/s. There was no holding time at the final temperature, and nominal initial cooling rates were typically in the range 500-600°C/s. The error bars in Figures 3-9 to 3-15 are standard deviations calculated separately for neat and contaminated soil from duplicate experiments performed at each heating rate. Again, all of the data are plotted on the basis of 50 mg of contaminant-free EPA soil, with an additional 2.49 mg of pyrene in the case of contaminated soil.

Weight loss (Figure 3-9) from neat or contaminated soil shows little change with increasing heating rate from 100 to 1000°C/s, but decreases when heating rate is increased to 5000°C/s. The implied pyrene removal as affected by heating rate was calculated from the difference in the curves in Figure 3-9 but no statistically significant effect was found. Tar yields from neat soil remain constant over the stated range of heating rates, while
Figure 3-9. Mass loss data for neat (O) and 4.75 wt. % pyrene-contaminated soil (●) as affected by heating rate (nominal peak temperature = 550°C, zero holding time). Curves are freehand drawn trendlines.
Figure 3-10. Mass of volatilized tars for neat (O) and 4.75 wt. % pyrene-contaminated soil (∙) as affected by heating rate (nominal peak temperature = 550°C, zero holding time). Curves are freehand drawn trendlines.
Figure 3-11. Carbon monoxide release for neat (O) and 4.75 wt. % pyrene-contaminated soil (●) as affected by heating rate (nominal peak temperature = 550°C, zero holding time). Curves are freehand drawn trendlines.
Figure 3-12. Acetylene release for neat (O) and 4.75 wt. % pyrene-contaminated soil (●) as affected by heating rate (nominal peak temperature = 550°C, zero holding time). Curves are freehand drawn trendlines.
Figure 3-13. Carbon dioxide release for neat (O) and 4.75 wt. % pyrene-contaminated soil (●) as affected by heating rate (nominal peak temperature = 550°C, zero holding time). Curves are freehand drawn trendlines.
Figure 3-14. Methane release for neat (O) and 4.75 wt. % pyrene-contaminated soil (●) as affected by heating rate (nominal peak temperature = 550°C, zero holding time). Curves are freehand drawn trendlines.
Figure 3-15. Ethylene release for neat (O) and 4.75 wt. % pyrene-contaminated soil (●) as affected by heating rate (nominal peak temperature = 550°C, zero holding time). Curves are freehand drawn trendlines.
release from contaminated soil increases slightly from 100 to 1000°C/s (Figure 3-10). The mean value of the tar yield determined from contaminated soil at 5000°C/s indicates a decline in the tar release when heating rate is further increased from 1000 to 5000°C/s. However the error bars in the data imply no statistically significant change from the tar yields at 1000°C/s. The tars collected in the contaminated soil runs were white with a very slight pink hue in all cases, except at 1000°C/s where the tars were yellowish pink. Color changes with temperature or heating rate may be a qualitative sign of chemical modification of the pyrene. Detailed chemical analysis of the tars is needed to assess this possibility. No significant amounts of condensables (benzene to naphthalene) were collected in any of the runs.

Light gas yields from neat and contaminated soils are shown in Figures 3-11 to 3-15. Carbon monoxide yields (Figure 3-11) were independent of heating rate for neat soil, and maximized at 1000°C/s for contaminated. Acetylene production from neat and contaminated soil (Figure 3-12) as well as CO₂, CH₄, and C₂H₄ production from contaminated soil (Figures 3-13 to 3-15 respectively) all show evidence of maximization at a heating rate between 100 and 5000°C/s. Carbon dioxide, methane, and ethylene yields for neat soil runs may also pass through a maximum at intermediate heating rates, but this trend is not well established because of the large amount of scatter in these data. Comparison of light gas production from neat and contaminated soil runs shows no statistically significant differences for CO, CO₂, CH₄, C₂H₂, or C₂H₄ at 100 and 5000°C/s (Figures 3-11, 3-13, 3-14, 3-12, 3-15 respectively), but clearly higher yields of CO and C₂H₂ from contaminated soil at 1000°C/s. Slightly higher CO₂ (Figure 3-13) and possibly C₂H₄ (Figure 3-15) yields from contaminated soil at 1000°C/s cannot be ruled out given the scatter in the data.
The present heating rate data (Figures 3-9 to 3-15) cannot be rationalized by the simple notion that with increasing heating rate, less time is needed to reach final temperature and thus less time is available for reaction. This notion is appropriate for phenomena governed by a single first-order kinetic pathway. However as shown *inter alia* by Darivakis et al. [10], heating rate effects become more complex even for relatively simple reaction networks, e.g.

\[ A \xrightarrow{(ab)} B \xrightarrow{(bc)} C \xrightarrow{(bd)} D \]  

(3)

Here the maxima in the yields of several of the light gases (e.g. CO, CO2, CH4, C2H2 in Figures 3-11, 3-13, 3-14, and 3-12 respectively) for neat and/or contaminated soil, with increasing heating rate, suggest that competing kinetic pathways may govern the net production of these species. As discussed above (reactions (1) and (2)), mineral carbonate decomposition and CO2 gasification of soil or pyrene-derived carbon may, for some heating conditions, have prominent roles in determining CO2 and CO production. Focusing on liquids production from coal pyrolysis, Darivakis et al. [10] point out that yields of product C in reaction (3) are highest when pathways ab and bc occur at kinetically identical rates and that changing, in their case increasing, heating rate is one way to bring about such “kinetic homogenization”. Similar effects may be occurring here for CO production from contaminated soil by the hypothesized reaction sequence (1) and (2) as heating rate is increased from 100°C/s (Figure 3-11). The absence of a corresponding heating rate effect for neat soil may reflect inadequate carbon to drive reaction (2).
Networks similar to (3), or more complex variations thereof, may account for the heating rate induced maxima in the yields of the other light gases noted above. We are not able to hypothesize specific pathways for all of the products, although CO production by a sequence such as (1) and (2) seems plausible. The enhanced yields of C₂H₂ (Figure 3-12) and C₂H₄ (Figure 3-15) from contaminated soil at 1000°C/s (vs. 100 and 5000°C/s) suggests a possible role for pyrene reacting in production of these and possibly other hydrocarbon species.

The apparent absence of a heating rate effect on total weight loss from neat and contaminated soil for the 100 to 1000°C/s range (Figure 3-9) is difficult to explain. It may reflect a fortuitous balancing of mass transfer and chemical reaction effects, i.e. at 1000°C/s, accelerated buildup of gas to drive pressure driven flow at the higher heating rate but less time for key chemical reactions. The decline in weight loss as heating rate increases from 1000 to 5000°C/s may then arise from dominance by the reduced time for chemical reaction contributions to weight loss.

3.3 Summary

The above results lead to the following conclusions regarding removal of pyrene, and by induction physico-chemically similar PAH contaminants, from soils by rapid heating under an inert atmosphere:

- Temperature strongly influences the rates and extents of contaminants removal, and the yields and release rates of the resulting volatiles.
- Yields of several light gases, e.g. CO, CO₂, C₂H₂, CH₄, and possibly C₂H₄ from pyrene-contaminated soil, pass through maxima with increasing
heating rate. However heating rate has little effect on pyrene removal efficiency.

- Pyrene, and presumably other polycyclic aromatic compounds of similar volatility and chemical reactivity, can undergo chemical modifications during thermal desorption from soil at higher temperatures, e.g., at or above ~500°C at a nominal heating rate of 1000°C/s.

- The presence of contaminant on the soil can lead to off-gas compositions significantly different from those obtained by heat treatment of uncontaminated soil, e.g., in the present case of pyrene contamination of an EPA Synthetic Soil, CO production was strongly enhanced and C2H2 showed modest increases for some heating conditions. This effect is apparently a consequence of chemical reactions between or among contaminant(s) and soil, or their decomposition products at elevated temperatures, e.g., above ~500°C at a heating rate of 1000°C/s.

- It is further tentatively concluded that there is potential for biologically active products to be formed during soil thermal treatment, either from decomposition of the contaminant itself or through chemical reactions between soil or soil decomposition products, and a PAH contaminant. Bioactivity testing and detailed chemical analyses of higher molecular weight products (e.g. tars) are needed to further test this inference.

- Thermal treatment can substantially reduce, and potentially virtually eliminate pyrene contamination in soils. Thus controlled thermal treatment should also provide high extents of removal from soils of other polycyclic aromatics similar in volatility and chemical reactivity to pyrene.

- At higher temperatures (e.g. around 530°C at 1000°C/s), extents of decontamination, calculated by comparing weight loss of contaminated soil (corrected for weight loss from the soil itself) to the initial weight of
contaminant, may exceed 100%. The apparent explanation is that chemical reactions between the soil and the contaminant, (or contaminant reaction products) cause additional volatilization of the soil. Thus, soil-contaminant reactions may cause gravimetric measurements to overestimate the true level of contaminant removal. Where possible, individual products yields as well as elemental and total material balances should be determined as further indicators contaminant removal efficiency.

3.4 References


5. Boynton, R. S., Chemistry and Technology of Lime and Limestone. J. Wiley and Sons, New York, pp. 244-6, 1980.
Chapter 4

Data Analysis

4.0 Introduction

This chapter discusses the chemical makeup of soil, provides an overview of pyrene and soil pyrolysis studies, introduces the models used to describe the data presented in Chapter 3, and discusses possible chemical reaction networks leading to the observed data trends based on the efficacy of these models.

4.1 Soil Constituents

While this section is not intended to provide a complete list of soil minerals and soil organics found in natural soils, it is intended to describe the complexity of the composition of both the organic and inorganic fractions of soil, and the various possible combinations of minerals and soil organics which could be present in soils at a particular Superfund site. As will be discussed in Sections 4.2 and 4.3, this complexity has impacted greatly soil pyrolysis studies and the approach to modeling of the data presented in Chapter 3.

4.1.1 Soil Formation and General Makeup [1]

Soils are a “finely divided rock-derived material containing an admixture of organic matter and capable of supporting vegetation. Soils are independent natural bodies, each with a unique morphology resulting from a particular combination of climate, living plants
and animals, parent rock materials, relief, the groundwaters, and age” [1]. Soil generally occurs as thick or thin horizons (or layers), more or less parallel to the surface. Each horizon differs from its neighbors in one or more properties such as color, texture, structure, consistency, porosity, and reaction.

Soil is formed through the accumulation of unconsolidated rock fragments (or parent rock) and subsequent formation of horizons. Natural processes promoting accumulation could be glaciers, wind, gravity, water, or physical and chemical weathering of hard rocks. Horizon differentiation can come about due to a series of gains, losses, transfers, and transformations of organic matter, soluble salts, carbonates, silicate clay minerals, sesquioxides, and silica. Gains can come in the form of additions of organic matter, oxygen, and water through oxidation and hydration as well as from additional mineral deposition over time at the soil surface or deposition via groundwater. Losses are mainly through material being dissolved or suspended in water percolating through the soil. Transfers of minerals and organic materials, which commonly occur in soils, can occur by dissolution or suspension in water moving through the soil followed by deposition due to uptake by plant roots, evaporation, or precipitation due to changes in conditions between soil horizons. Horizon formation can be upset due to mixing of soil layers by burrowing animals (rodents and earthworms), creepage or sliding of soils on steep slopes, or by plants withdrawing metals from deeper horizons and depositing them on the surface via leaf litter.

The soil composition and structure ultimately depend on the relative strengths of the above processes which are determined not only by natural factors such as climate, plants and animals, groundwater, and the period of time during which these processes occur, but also the ways humans have used the soil (agriculture, building, etc.). As a result, soil compositions and makeup can vary significantly from site to site as well as within a site.
4.1.2 Soil Minerals [2]

While the information of this subsection is summarized in Tables 4-1 and 4-2, the four major mineral classes as well as the particular properties and commonly encountered minerals of each class are discussed below.

*Halide, sulfate, and carbonate minerals* - This group is characterized by the simple chemical structures, physical softness, and solubility in water. The major minerals in the group are halite (NaCl), gypsum (CaSO₄·2H₂O), calcite (CaCO₃), and dolomite (CaMg(CO₃)₂).

*Sulfides* - Pyrite (FeS₂) is the most common of this class of compounds and is usually not present in large quantities in most soils. Pyrite is unstable in oxidizing environments, weathering to form sulfuric acid and the minerals jarosite (KFe₃(SO₄)₂(OH)₆) and gypsum when mining leaves pyritic material on the surface or when areas containing sulfides are drained.

*Oxides, hydroxides, and oxyhydroxides* - This class of minerals is formed when certain types of minerals break down during weathering and release cations and anions, which recombine to form more stable minerals. Common examples are Gibbsite (Al(OH)₃), Geothite (FeOOH), Hematite (Fe₂O₃), manganese oxides and hydroxides, and TiO₂.

*Silicates* - This is a very large and important class of minerals accounting for nearly 40% of the common minerals, constituting well over 90% of the earth's crust, and comprising the bulk of most soils. In sand- (75 μm - 4.75 mm) and silt- (2-20 μm) sized fractions of soils, minerals such as olivines, pyroxenes, amphiboles, micas, feldspars, and quartz predominate. Phyllosilicate (clay) minerals dominate the clay fraction (≤ 2 μm).
### Common non-silicate minerals in soils

<table>
<thead>
<tr>
<th>Mineral class</th>
<th>Mineral</th>
<th>Chemical formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Halides</td>
<td>Halite</td>
<td>NaCl</td>
</tr>
<tr>
<td>Sulfates</td>
<td>Gypsum</td>
<td>CaSO₄·2H₂O</td>
</tr>
<tr>
<td></td>
<td>Jarosite</td>
<td>KFe₂(SO₄)₃(OH)₆</td>
</tr>
<tr>
<td>Sulfides</td>
<td>Pyrite</td>
<td>FeS₂</td>
</tr>
<tr>
<td>Carbonates</td>
<td>Calcite</td>
<td>CaCO₃</td>
</tr>
<tr>
<td></td>
<td>Dolomite</td>
<td>CaMg(CO₃)₂</td>
</tr>
<tr>
<td></td>
<td>Nahcolite</td>
<td>NaHCO₃</td>
</tr>
<tr>
<td></td>
<td>Trona</td>
<td>Na₄CO₅·NaHCO₅·2H₂O</td>
</tr>
<tr>
<td></td>
<td>Soda</td>
<td>Na₄CO₅·10H₂O</td>
</tr>
<tr>
<td>Oxides and Hydroxides</td>
<td>Gibbsite</td>
<td>Al(OH)₃</td>
</tr>
<tr>
<td>Aluminum</td>
<td>Nordstrandite</td>
<td>Al(OH)₃</td>
</tr>
<tr>
<td></td>
<td>Boehmite</td>
<td>AlOOH</td>
</tr>
<tr>
<td></td>
<td>Diaspore</td>
<td>AlOOH</td>
</tr>
<tr>
<td></td>
<td>Corundum</td>
<td>Al₂O₃</td>
</tr>
<tr>
<td>Iron</td>
<td>Hematite</td>
<td>Fe₃O₄</td>
</tr>
<tr>
<td></td>
<td>Goethite</td>
<td>FeOOH</td>
</tr>
<tr>
<td></td>
<td>Lepidocrocite</td>
<td>FeOOH</td>
</tr>
<tr>
<td></td>
<td>Maghemite</td>
<td>Fe₃O₄</td>
</tr>
<tr>
<td></td>
<td>Ferricydrite</td>
<td>Fe₃O₄(OH)·4H₂O</td>
</tr>
<tr>
<td></td>
<td>Magnetite</td>
<td>Fe₃O₄</td>
</tr>
<tr>
<td>Manganese</td>
<td>Lithiophorite</td>
<td>(Al₆Li)MnO₄(OH)</td>
</tr>
<tr>
<td></td>
<td>Birnessite</td>
<td>variable</td>
</tr>
<tr>
<td></td>
<td>Hollandite</td>
<td>Ba₃MnO₆</td>
</tr>
<tr>
<td></td>
<td>Pyrolusite</td>
<td>MnO₂</td>
</tr>
<tr>
<td></td>
<td>Todorkite</td>
<td>variable</td>
</tr>
<tr>
<td></td>
<td>Manganite</td>
<td>MnOOH</td>
</tr>
<tr>
<td>Titanium</td>
<td>Rutile</td>
<td>TiO₂</td>
</tr>
<tr>
<td></td>
<td>Anatase</td>
<td>TiO₂</td>
</tr>
<tr>
<td></td>
<td>Ilmenite</td>
<td>FeTiO₃</td>
</tr>
</tbody>
</table>

Table 4-1. Common non-silicate minerals in soils

<table>
<thead>
<tr>
<th>Silicate class, unit composition, arrangement of SiO₄ tetrahedra†</th>
<th>Mineral</th>
<th>Ideal formula†</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Nesosilicates (SiO₄)⁻</strong></td>
<td>Olivine</td>
<td>(Mg, Fe)₂SiO₄</td>
</tr>
<tr>
<td></td>
<td>Forsterite</td>
<td>Mg₂SiO₄</td>
</tr>
<tr>
<td></td>
<td>Fayalite</td>
<td>Fe₂SiO₄</td>
</tr>
<tr>
<td></td>
<td>Zircon</td>
<td>ZrSiO₄</td>
</tr>
<tr>
<td></td>
<td>Sphene</td>
<td>CaTiO(SiO₄)</td>
</tr>
<tr>
<td></td>
<td>Topaz</td>
<td>Al₂SiO₄(F, OH)</td>
</tr>
<tr>
<td></td>
<td>Garnets</td>
<td>X₃Y₂(SiO₄)₆, where X = Ca, Mg, Fe²⁺, Mn²⁺ Y = Al, Fe³⁺, Cr³⁺</td>
</tr>
<tr>
<td></td>
<td>Andalusite</td>
<td>Al₂SiO₄</td>
</tr>
<tr>
<td></td>
<td>Sillimanite</td>
<td>Al₂SiO₄</td>
</tr>
<tr>
<td></td>
<td>Kyanite</td>
<td>Al₂SiO₄</td>
</tr>
<tr>
<td></td>
<td>Staurolite</td>
<td>Fe₅Al₂O₆(SiO₄)₃(O, OH)</td>
</tr>
<tr>
<td><strong>Sorosilicates (Si₃O₉)⁻</strong></td>
<td>Epidote</td>
<td>Ca₄(Al, Fe)₂Al₂O₆(SiO₄)₃(SiO₄)₃(OH)</td>
</tr>
<tr>
<td><strong>Cyclosilicates (Si₄O₁₁)⁻</strong></td>
<td>Beryl</td>
<td>Be₃Al₂Si₆O₁₆</td>
</tr>
<tr>
<td></td>
<td>Tourmaline</td>
<td>(Na, Ca)(Li, Mg, Al)(Al, Fe, Mn)(BO₃)(SiO₄)₃(OH)</td>
</tr>
<tr>
<td><strong>Inosilicates (single chains) (SiO₄)⁻</strong></td>
<td>Pyroxenes</td>
<td>(Ca, Na)(Mg, Fe, Al)(Si, Al)O₄</td>
</tr>
<tr>
<td></td>
<td>Augite</td>
<td>MgSiO₄</td>
</tr>
<tr>
<td></td>
<td>Enstatite</td>
<td>MgSiO₄</td>
</tr>
<tr>
<td></td>
<td>Hypersthene</td>
<td>(Mg, Fe)SiO₄</td>
</tr>
<tr>
<td></td>
<td>Diopside</td>
<td>NaMgSiO₄</td>
</tr>
<tr>
<td></td>
<td>Hedenbergite</td>
<td>CaFeSiO₄</td>
</tr>
<tr>
<td></td>
<td>Pyroxenoids</td>
<td>CaSiO₄</td>
</tr>
<tr>
<td></td>
<td>Wollastonite</td>
<td>Ca₅SiO₄</td>
</tr>
<tr>
<td></td>
<td>Rhodonite</td>
<td>MnSiO₄</td>
</tr>
<tr>
<td><strong>Inosilicates (double chains) (SiO₄)⁻</strong></td>
<td>Amphiboles</td>
<td>(Ca, Na)₅(Mg, Fe, Al)₄ Si₄</td>
</tr>
<tr>
<td></td>
<td>Hornblende</td>
<td>(Ca, Na)₅(Mg, Fe, Al)₄Si₄(OH)</td>
</tr>
<tr>
<td></td>
<td>Tremolite</td>
<td>Ca,Mg,SiO₄(OH)</td>
</tr>
<tr>
<td></td>
<td>Actinolite</td>
<td>Ca,Mg,SiO₄(OH)</td>
</tr>
<tr>
<td></td>
<td>Cummingtonite</td>
<td>Mg, Fe,SiO₄(OH)</td>
</tr>
<tr>
<td></td>
<td>Grunerite</td>
<td>Fe,SiO₄(OH)</td>
</tr>
</tbody>
</table>

Table 4-2. (continued on next page)
### Table 4-2. Common silicate minerals in soils

4.1.3 Soil Organic Matter

Organic content in soils can vary from 0.1% in desert soils to nearly 100% in organic soils [3], depending on drainage and clay content of the soil at a particular site [4]. Soils containing less than 12-20% organic carbon are termed mineral, while the remainder is termed organic soil. To serve as a reference, the average organic matter content in most good agricultural soils is about 1 - 5% [1]. Organic matter content in soil varies depending on soil texture, climate (precipitation and temperature), vegetation, and soil management. For example, organic matter content in soils increases with annual rainfall simply because of increased plant growth, while it decreases with increased soil temperature since higher temperatures stimulate microbial consumption of these materials.

Soil organic matter, or humus, is one of the most complex materials found in nature [5], containing plant and animal residues at various stages of decomposition, cells and tissues of soil organisms, and substances synthesized by the soil microbial population [1]. This soil fraction is formed through numerous complex biochemical processes characteristic of the decomposition of organic matter [1,6], beginning simply with fracturing of the cell walls [6]. Humus is generally a complex mixture of organic compounds which can be broken down into two major polymer groups: (1) “humic substances” or the more resistant fats, waxes, lignins, and oils derived from residues of higher plants, and (2) microorganism-synthesized compounds such as polysaccharides and polynuronides. Humic substances constitute 70-80% of soil organic matter, with polysaccharides (made up of sugars and sugar-type constituents) making up most of the remainder [7]. Carbohydrates, proteins, peptides, amino acids, fats, waxes, resins, pigments, and other low-molecular weight organic substances are also present, but in very small quantities since they are easily attacked by microorganisms in soil and have relatively short survival rates [8].
Chemical investigations on humic substances have been performed by soil scientists over the last 200 years [8-10], but much is unknown concerning their origin, synthesis, chemical structure, reactions, and their functions in the environment [3,4]. Since soil organics are often tightly bound in soils, various techniques have been employed to extract organics from soils and to analyze the extracts [3,9,10]. Currently hypothesized structures of humic substances are based on a classical extraction technique using dilute base and acid solutions (Figure 4-1), classifying humic substances into three categories: humic acid, fulvic acid, and humin. The soil is initially extracted with an alkali to separate out the humin (insoluble fraction tightly bound to soil), and the extract is treated with an acid to separate the humic acid and fulvic acid fractions.

Scanning electron micrographs show that, in aqueous solutions, humic and fulvic acids are present as elongated, multi-branched filaments 20 - 100 nm in width (Figure 4-2) [3,7]. Chemical analyses and studies of these fractions using a wide variety of analytical techniques (e.g., elemental analysis, infrared spectroscopy, x-ray analysis, size-exclusion gel filtration, chemical oxidation, pyrolysis-gas chromatography, biodegradation [2,3,9,10]) have led to proposals for the chemical structures of humic (Figure 4-3) and fulvic (Figure 4-4) acids. To date, there is no consensus on the correct structures of these substances.

As for the humins (unextractable fraction of the soil organic matter), a comprehensive picture concerning its structure and chemical makeup is still undetermined. Some elemental analysis and functional group chemical analyses have yielded some information [8]. Optical examination [6] has shown: (1) that a substantial amount of the humin is enclosed in microaggregates in the soil, (2) the presence of polysaccharides and other biosynthesized products such as resins, waxes, and lipids accounting for more than 10% of the humin, and (3) the presence of products which have been irreversibly bound to soil constituents such as clays.
Figure 4-1. Scheme for the fractionation of humic substances

Figure 4-2. Scanning electron micrographs of humic acid and fulvic acid

Figure 4-3. Proposed chemical structures for humic acid

Figure 4-4. Proposed chemical structures for fulvic acid

4.2 Pyrolysis Studies

A brief literature review of pyrene and soil pyrolysis studies is presented to provide an overview of the possible contributions of each component to the products chemistry potentially obtainable from the thermal treatment of pyrene-contaminated soil.

4.2.1 Pyrene Pyrolysis

Limited literature is currently available concerning pyrene pyrolysis. In 1957, Lang and Buffleb [11] pyrolyzed a 1:4 molar mixture of pyrene and benzene at approximately 1000 K. Several products were identified, but it is believed that the excess benzene may have suppressed products that would be observed if pyrene were singly pyrolyzed [12].

Madison and Roberts [13] in 1958 pyrolyzed liquid-phase pyrene in sealed glass capillary tubes at 475°C for 1 hour. Pyrene was found generally not to break down into smaller compounds, but no heavy residues were found in the tubes following heating. A small amount of gaseous products was formed (moles gas collected to moles pyrene initially charged was 0.0011), but the amount was too small to be analyzed. In a later study by Sharkey et al. [14] using a comparable technique under identical heating conditions, pyrene pyrolysis products were identified by mass spectrometry. Gas products, which made up 0.01% (wt.) of the initial charge, were found to be predominantly hydrogen, with the remainder made up of methane, ethane, ethylene, propane, and propylene. Dihydropyrene and dipyrenyl were tentatively identified in the residues, which constituted 2.2% (wt.) of the initial charge.

Mukherjee [12] used a drop tube reactor to study the pyrolysis of pyrene at 1213, 1303, 1393, and 1483 K. The large number of products observed was categorized into four groups: bipyrenes, their condensation products, fragmentation products, and soot. At the time of writing, data relating only to the first two groups were written up and will be
documented here. At the lowest two temperatures, bipyrenes were the major product with all 6 isomers present. The six isomers are shown in Figure 4-5. The relative quantities of the isomers could be explained by a statistical/steric argument based on the degeneracy of sites around the pyrene rings. Condensation products obtained, such as dibenzoperopyrene, are shown in Figure 4-6. The pathways shown in Figure 4-6 are deduced from the observation that the yields of these condensation products rose during the depletion of bipyrene. More information concerning the last two categories will be available when analysis of the data is completed.

From the work done to this date, pyrene pyrolysis products and pathways leading to their formation have been suggested and in some cases confirmed. However, kinetic information about these reactions have yet to be determined.

4.2.2 Soil Pyrolysis

In a companion study, Bucalá et al. [15] studied weight loss, tars, and light gas (CO, CO₂, C₁-C₂ hydrocarbons) evolution in 3 psig helium from the uncontaminated EPA Synthetic Soil Matrix for the temperature range 350-1050°C at a nominal 1000°C/s heating rate. Carbon dioxide, carbon monoxide, and tars were the major volatiles observed. It was hypothesized that: (1) CO₂ was a result of calcination of the carbonates present in the soil, (2) side reaction of carbonate-derived CO₂ with organic carbon present in the soil was a major CO source, and (3) the small quantities of hydrocarbons observed are produced from the small organic fraction present in the soil. Material and carbon balances in this study were good. Oxygen balances were fair and hydrogen balances were poor, possibly because yields of chemical water were not determined. This work did show that a multiple independent parallel reaction (MIPR) model can satisfactorily describe total weight loss as affected by temperature. Model parameters obtained from the data are listed in Table 4-3.
Figure 4-5. Six bipyrenes observed by Mukherjee et al.


Figure 4-6. Condensation products of bipyrenes observed by Mukherjee et al.

Kinetics Parameters for Total Weight Loss and Individual Products Release from Rapid Heating of Soil Under 3 psig Helium.

<table>
<thead>
<tr>
<th>Product</th>
<th>E. Kcal/g-mole</th>
<th>σ. Kcal/g-mole</th>
<th>k₀ (sec⁻¹)</th>
<th>V*, wt%</th>
<th>Std error of est. wt%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight loss</td>
<td>20785</td>
<td>3787</td>
<td>7.38E4</td>
<td>21.4000</td>
<td>1.1884a</td>
</tr>
<tr>
<td>CO</td>
<td>11934</td>
<td></td>
<td>35.62</td>
<td>4.1274</td>
<td>0.1951b</td>
</tr>
<tr>
<td>CO₂)₁c</td>
<td>18303</td>
<td></td>
<td>2.48E5</td>
<td>0.5851</td>
<td>0.0705b</td>
</tr>
<tr>
<td>(CO₂)₂d</td>
<td>35547</td>
<td></td>
<td>4.61E6</td>
<td>9.3310</td>
<td>0.9126b</td>
</tr>
<tr>
<td>CH₄</td>
<td>9754</td>
<td></td>
<td>2.65E2</td>
<td>0.0759</td>
<td>0.0051h</td>
</tr>
<tr>
<td>C₂H₂</td>
<td>14327</td>
<td></td>
<td>2.64E3</td>
<td>0.0128</td>
<td>0.0013h</td>
</tr>
<tr>
<td>C₂H₄</td>
<td>6748</td>
<td></td>
<td>6.65</td>
<td>0.0364</td>
<td>0.0041h</td>
</tr>
<tr>
<td>C₂H₆</td>
<td>3373</td>
<td></td>
<td>0.88</td>
<td>0.0052</td>
<td>0.0073h</td>
</tr>
<tr>
<td>Tars</td>
<td>4522</td>
<td></td>
<td>2.50</td>
<td>3.35</td>
<td>0.7010h</td>
</tr>
</tbody>
</table>

|^| Defined as \[ \sum_{j=1}^{n} \left( \frac{(V_{j\text{mode}l} - V_{j\text{expt}})^2}{(n - 4)} \right)^{1/2} \]

|^| Defined as \[ \sum_{j=1}^{n} \left( \frac{(V_{j\text{mode}l} - V_{j\text{expt}})^2}{(n - 3)} \right)^{1/2} \], where n is the number of data points

|^| Temperature range of fit: 354-626°C.

|^| Temperature range of fit: 626-1033°C.

Table 4-3. Multiple independent parallel reaction model (MIPR) parameters obtained for weight loss and volatiles evolution from an EPA Synthetic Soil Matrix

Other soil pyrolysis literature is derived from the use of pyrolysis-mass spectrometry by soil scientists as a means to derive structural information about soil organic matter. Whole samples of an agricultural soil and a forest soil, as well as of their organic fractions (humic acid, fulvic acid, and humin), were pyrolyzed by Schnitzer and Schulten [16,17]. Using a field ionization mass spectrometer, they identified compound classes (e.g., phenols, fatty acids, methyl naphthalenes) present in the soil and in individual soil organic fractions. Schulten and Schnitzer [18] identified over 200 compounds in the humic acid fraction of these soils, including n-alkanes, n-alkenes, aromatics and their alkylated derivatives, alcohols and phenols, and ketones. Earlier work by Hempfling and Schulten et al. [19] with humic soils have also suggested formation of organic acids, esters, and lignins in the pyrolysates. Saiz-Jimenez and De Leeuw [20] have also identified over 300 compounds in soil organic fractions from a Spanish brown soil. Compounds identified by mass spectrometry include light gases (carbon monoxide, carbon dioxide, methane, ethylene, ethane, propane, and up), alcohols (n-alkyl, phenols, aromatic), aromatics (unsubstituted, alkylated), fatty acids, and fatty acid esters among many others. Similar work has also been performed by Martin et al. [21] and Chiavari et al. [22]. While the amount of soil pyrolysis data is not extensive, Saiz-Jimenez strongly recommends caution in using soil pyrolysis data since there is often serious debate regarding interpretation of the data and potential problems with or differences between experimental conditions (pyrolysis temperatures varying from 300 to 700°C depending on the investigator) [23].

Among the common minerals cited in Section 4.1, many minerals such as carbonates and clays are also known to degrade or undergo transformations at elevated temperatures. Carbonates such as calcium carbonates and dolomite are well known to emit carbon dioxide to form their respective oxides (CaO and CaO-MgO) at elevated temperatures. Dissociation temperatures (dissociation beginning in a 100% CO₂ atmosphere at 1 atm. pressure) for calcite and dolomite are 898°C and 402°C [24], although in soil thermal treatment systems dissociation likely begins at much lower temperatures.
Clays are also known to undergo change (Table 4-4) over a wide temperature range [25],
starting with loss of chemical water below 400°C, dehydroxylation between 400 - 750°C,
phase transitions above 750°C, and metal ion oxidation reactions (even without external
oxygen) above temperatures starting at 400°C, depending on the minerals present.

4.2.3 Other Possible Reactions

Aside from the degradation reactions discussed in the previous subsection, other
reactions in the soil may also contribute to the products formed during thermal treatment of
soil.

Dragun [26] mentions that several organic chemical reactions are known to occur in
soil at ambient conditions: hydrolysis, substitution, elimination, oxidation, reduction, and
surface-catalyzed hydrolysis and oxidation. It is thought that as soil temperature increases,
the reaction rates will be accelerated under the appropriate conditions. As hydrolysis,
substitution and elimination occur only in an aqueous phase [27], these reactions are
expected to contribute little for conditions where free-standing water is not likely to exist.
Saltzman [28] indicates that surface-catalyzed hydrolysis occurs at high temperature, and
that the reaction is enhanced as moisture is removed from the surfaces of clays like
kaolinite. Some soils, clays, and minerals have the ability to catalyze oxidation of many
organic chemicals at ambient conditions as well as at higher temperatures and pressures
[29]. The role of reduction reactions is unclear since reduction is the least understood and
least studied reaction of those listed [27].

Other literature suggests chemical reactions between soil minerals and organic
compounds. As mentioned in Section 3.1.2, reactions between metal carbonates and
carbon fuels to form carbon monoxide suggest the possibility of similar reactions with
organic soil contaminants or with soil organic carbon. Researchers at Battelle’s Pacific
Northwest Laboratories have found a novel approach of using sodium bicarbonate to treat
### Table 7.1(a) Summary of thermal reactions - dioctahedral minerals

<table>
<thead>
<tr>
<th>Mineral</th>
<th>Reaction</th>
<th>Temperature</th>
<th>Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kaolinite</td>
<td>Prehydroxylation — 450-550 °C</td>
<td>1:1 type</td>
<td>Al₂Si₂O₅(OH)₄</td>
</tr>
<tr>
<td>Montmorillonite</td>
<td>250 °C</td>
<td>2:1 type</td>
<td>Al₂Si₂O₅(OH)₄</td>
</tr>
<tr>
<td>Montmorillonite</td>
<td>750 °C</td>
<td>2:1 type</td>
<td>Al₂Si₂O₅(OH)₄</td>
</tr>
<tr>
<td>Montmorillonite</td>
<td>1150 °C</td>
<td>2:1 type</td>
<td>Al₂Si₂O₅(OH)₄</td>
</tr>
<tr>
<td>Montmorillonite</td>
<td>230 °C</td>
<td>2:1 type</td>
<td>Al₂Si₂O₅(OH)₄</td>
</tr>
<tr>
<td>Montmorillonite</td>
<td>900-1250 °C</td>
<td>2:1 type</td>
<td>Al₂Si₂O₅(OH)₄</td>
</tr>
</tbody>
</table>

**Note:**
- With separation of silica and other phases.
- Ideal formula M₂⁺ₓH₂O (Al₁₋ₓ, Siₓ)₂O₅(OH)₄
- Temperatures variable according to chemical composition, crystal size, and conditions of heating.

### Table 7.1(b) Summary of thermal reactions - trioctahedral minerals

<table>
<thead>
<tr>
<th>Mineral</th>
<th>Reaction</th>
<th>Temperature</th>
<th>Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serpentinite</td>
<td>550 °C</td>
<td>1:1 type</td>
<td>Mg₃Si₂O₅(OH)₄</td>
</tr>
<tr>
<td>Saponite</td>
<td>550 °C</td>
<td>2:1 type</td>
<td>Mg₃Si₂O₅(OH)₄</td>
</tr>
<tr>
<td>Vermiculite</td>
<td>500-800 °C</td>
<td>2:1 type</td>
<td>Mg₃Si₂O₅(OH)₄</td>
</tr>
<tr>
<td>Chlorete</td>
<td>450-550 °C</td>
<td>2:1 type</td>
<td>Mg₃Si₂O₅(OH)₄</td>
</tr>
</tbody>
</table>

**Note:**
- With separation of silica and other phases.
- Ideal formula M₂⁺ₓH₂O Mg₃Si₂₋ₓAlₓO₅(OH)₄
- Temperatures variable according to chemical composition, crystal size, and conditions of heating.

### Table 4.4 Reactions of clay minerals at elevated temperatures

PCB (polychlorinated biphenyls) contaminated soils [30]. In this process, sodium bicarbonate is mixed with the contaminated soil before treatment in a rotary reactor at 350°C. The hydrogen from the sodium bicarbonate is used to replace chlorine and other halogens in the PCB's leaving NaCl salt and a non-toxic biphenyl. Similar reactions could occur during thermal treatments of soil as sodium bicarbonate is present in the natural minerals Nahcolite and Trona.

4.2.4 Mineral Effects on Pyrolysis

Work potentially relevant to soil decontamination studies relates to the effects of native minerals or inorganic additives on pyrolysis. Mineral effects on the pyrolysis of coal and a variety of hydrocarbons have been studied. Studies, some of which are described below, have documented the effect of minerals and added inorganic constituents on pyrolysis. As contaminated soil is largely inorganic with some organic constituents, it is important to direct attention to possible mineral effects on thermal decontamination of soil.

Ellig et al. [31] studied the effects of calcium oxide on the pyrolysis of n-heptane, benzene, toluene, and 1-methylnaphthalene between 550 to 950°C, by passing vapor of the pure compound through packed beds of calcium oxide/quartz mixtures or of quartz in control experiments. Upon comparing results, calcium oxide was observed to significantly increase the rates and extents of pyrolysis of the three aromatics. The temperature required for a given percentage conversion of a compound was found to decrease by 140 - 170°C in the presence of CaO compared to quartz alone. Global activation energies derived assuming first order kinetics showed that the activation energy over CaO was almost a factor of two lower than over quartz.

Lai [32] followed up Ellig's work by studying effects of CaO on the pyrolysis of benzene, 1-methylnaphthalene, 1-methylnaphthalene, and 9-methylnaphthalene. Again, the presence of CaO significantly increased rates and extents of aromatics pyrolysis, reducing
the temperature required for a given conversion by at least 140°C. The magnitude of the effect was found to increase with molecular size of the compound. Coke was a major product of these reactions suggesting that condensation reactions are important over CaO.

Effects of minerals on the pyrolysis of coal have also been studied. Yeboah [33] studied the effect of the addition of high calcium quicklime and of calcined dolomitic stones on coal pyrolysis. Coal pyrolysis runs with and without each mineral in a fluidized bed revealed that the presence of calcium oxide or dolomite lowers tar yield, and increases total gas, hydrogen, methane, and C₂+ hydrocarbon yields.

Franklin [34,35] initially extracted the majority of inorganics from coal using HF and HCl, and proceeded to add known amounts of minerals such as calcite and clays. When adding calcite and lime to the demineralized coal [34], the treated coal showed lower weight losses and higher char yields upon pyrolysis compared to the demineralized sample. Tar and lighter hydrocarbon yields were also reduced while yields of carbon oxides greatly increased.

Franklin later studied the effects of adding clays, quartz, and other minerals [35]. Pyrite, montmorillonite, and kaolinite reduced volatile yields by a statistically significant amount. Montmorillonite reduced tar yield by a statistically significant but slight extent. Kaolinite suppressed yields of light liquid hydrocarbons that were mainly benzene-toluene-xylene hydrocarbons. Calcium minerals were found to increase light hydrocarbon yields more than other minerals. Studies of the C₃ to C₈ hydrocarbons [36] indicated that C₃ hydrocarbons were unaffected, except in the case of calcite and lime where yields were reduced by 20%. Calcite and lime lowered yields of C₄ - C₆ hydrocarbons. Montmorillonite, kaolinite, and calcite reduced C₆ - C₈ volatile yields, while pyrite had less of an effect.

Pyrolysis has been demonstrated to be affected by the presence of a variety of inorganic compounds. Opportunity exists for mineral effects to be directly relevant to this soil decontamination study, since montmorillonite and kaolinite are known to be present in
the EPA soil, and calcined calcite and dolomite can be produced during thermal treatment from the calcium carbonate and dolomite also present.

4.3 Modeling Approach

Modeling the observed experimental data trends was used to shed light on proposed chemical processes taking place during the thermal treatment of neat and pyrene-contaminated soil. A global modeling approach seeking to describe the trends observed in the data through representative reactions sets was adopted. A global modeling approach, as opposed to a detailed compound-specific kinetic model, was also chosen because of the: 1) vast number of compounds present in the soil initially, a large fraction of which cannot be well-characterized or identified, which will produce a larger number of products on pyrolysis, 2) large number of uncharacterized reactions potentially taking place within the contaminant, soil organic, and soil mineral phases individually as well as potential chemical interactions between phases, 3) possible catalytic effects of the minerals on the kinetics which would be difficult to identify and quantify, and 4) the transient nature of soil particle structure and chemical constituents during the thermal treatment process, potentially affecting fundamental processes such as mass transfer (pore structure and size distribution changing due to calcination of carbonates) and catalytic effects (as clay minerals are chemically transformed).

Global modeling began with the employment of the multiple independent parallel reaction (MIPR) model which has been widely used for modeling coal pyrolysis at MIT, as coal and soil share the four characteristics listed above. The MIPR model has been used to describe the evolution rate of total volatiles (Anthony [37]), gaseous products (Serio [38]), and tar (Serio [38], Ko [39,40], and Darivakis [41]). Mass transfer and secondary reactions of tar in addition to the chemical decomposition of coal have been modeled by
many investigators using this model, such as Ko [39] in his thesis work. Using the same analogy, the MIPR model has been used by Darivakis [42] to describe weight loss from a pyrene-contaminated clay soil, and by Bucalá et al. [15] to model weight loss from an uncontaminated EPA synthetic soil.

As will be discussed in Section 4.4, the MIPR model was not sufficient to describe all the data. The global competitive reaction sets:

1. \[
\begin{align*}
A & \quad \text{B (side product)} \\
\quad & \quad \text{Product}
\end{align*}
\]

\{P\}

2. \[
\begin{align*}
A & \rightarrow \text{Product} \rightarrow B \quad \text{(side product)}
\end{align*}
\]

\{S\}

were also tested since these reaction schemes were suggested by the pathways illustrated in Figure 1-5. The competitive reaction sets were kept to two reactions to minimize the number of adjustable parameters.

4.4 Model Theories

4.4.1 Multiple Independent Parallel Reaction (MIPR) Model

Tars, soil weight loss, and light gas products can potentially be formed from a variety of source compounds present in the soil (neat or contaminated). For example, carbon dioxide could evolve from calcite and dolomite present in the soil, as well as from decarboxylation of the soil organic matter [23]. Hence viewing the reaction

\[
\text{(un)contaminated soil} \rightarrow \text{Product X}
\]
as a set of parallel independent reactions forming Product X, with each is not unreasonable. 
Due to different chemical bond strengths among compounds within the soil, each reaction 
leading to a select product is expected to have different activation energies and pre-
exponential factors.

The MIPR model [37] describes this phenomena and, lacking detailed kinetic 
information about each individual reaction, makes the simplifying assumption that Product 
X is formed via a large number of independent reactions, each of which is first-order with 
respect to the amount of product yet to be formed by that reaction:

\[
\frac{dV_i}{dt} = k_i (V_i^* - V_i) \quad \{1\}
\]

where \( V_i \) is the cumulative amount of product generated up to time \( t \) from reaction \( i \), \( V_i^* \) is 
the ultimate amount of product generated by reaction \( i \) \((V_i \rightarrow V_i^* \text{ as } t \rightarrow \infty) \), and \( k_i \) is the rate 
constant for reaction \( i \), here assumed to have the Arrhenius temperature dependence

\[
k_i = k_{i0} \exp\left(\frac{-E_i}{RT}\right) \quad \{2\}
\]

where \( k_{i0} \) and \( E_i \) are respectively the pre-exponential factor and activation energy for 
reaction \( i \), \( R \) is the Universal Gas Constant, and \( T \) is the absolute temperature. Substituting 
eq \{2\} into \{1\} and integrating, the amount of product yet to be formed by reaction \( i \) at 
time \( t \) is

\[
(V_i^* - V_i) = V_i^* \exp\left(\int_0^t k_i \, dt\right) \quad \{3\}
\]

The time integral is retained since the rate constant is a function of time in the non-
isothermal temperature-time profiles described in Chapter 2.

Parameter values for \( k_{i0}, E_i \) and \( V_i^* \) for each reaction \( i \) cannot in general be predicted 
a priori, and further simplifications (assumptions) are necessary [37]: (1) all reactions have
the same pre-exponential factors \( k_0 = k_0 \), for all \( i \), making the differences between all \( k_i \) being the activation energy \( E_i \), and (2) the number of reactions is large enough to permit \( E \) to be expressed as a continuous distribution function \( f(E) \). The result of these simplifications is that \( V^* \) becomes a differential fraction of the total amount of Product X formed, \( V^* \), where

\[
\frac{dV^*}{dE} = V^* f(E) \tag{4}
\]

and \( f(E) \) is the probability density function, which is normalized, i.e.

\[
\int_0^\infty f(E) \, dE = 1 \tag{5}
\]

allowing the quantity \( V^*_i \) to be written as

\[
V^*_i = dV^* = V^* f(E) \, dE \tag{6}
\]

Substituting Eq. (6) and (2) into Eq. (3) and integrating over all values of \( E \), the amount of Product X yet to be released \( (V^* - V) \) is

\[
V^* - V = V^* \int_0^\infty \exp \left( -k_0 \int_0^t \exp \left( \frac{-E}{RT(t)} \right) \, dt \right) f(E) \, dE \tag{7}
\]

Since the nature of the distribution of activation energies to form Product X is unknown, a Gaussian distribution has historically been assumed [37-42,15] and was used as the probability distribution function in this case.

\[
f(E) = \frac{\exp \left( -\frac{(E-E_0)^2}{2\sigma^2} \right)}{\sigma \sqrt{2\pi}} \tag{8}
\]

Substituting eq. (8) into eq. (7),

129
\[ V^* - V = V^* \int_0^\infty \exp(-k_0 \int_0^t \exp\left(-\frac{E_{\text{RT}(t)}}{R}\right) dt) \frac{\exp\left(-\frac{(E - E_0)^2}{2\sigma^2}\right)}{\sigma \sqrt{2\pi}} dE \tag{9} \]

where \( E_0 \) is the mean activation energy and \( \sigma \) is the standard deviation of the energy distribution.

With eq. (9), the model utilizes four adjustable parameters: \( k_0, V^*, E_0, \) and \( \sigma \) respectively. Best fit values for the kinetic parameters were calculated by least squares fitting, comparing each experimental value of \( V \) with predicted values obtained by integrating eq. (9) over the temperature-time profile for that experiment. The non-linear least squares algorithm used was a FORTRAN subroutine VMP obtained from Dr. Verónica Bucalá [43]. The complete debugged code is listed in Appendix C.

4.4.2 Competitive Reaction Model - Parallel Reactions

The fundamental basis for examining this model was the possibility that the limited soil organics or minerals present in soil (neat or contaminated) could yield a variety of products, depending on the favorability of specific reactions over others under different thermal treatment conditions. One possibility is competition between degradation reactions for the consumption of the soil organic matter to form light hydrocarbons, carbon monoxide, or tars. This model has been used previously by Kobayashi [45] to describe weight loss of lignite and bituminous coals.

In selecting a reaction mechanism for a competitive reaction model, reaction sets such as reaction \{P\} in Section 4.3 and reaction \{3\} in Section 3.2.2, as well as using the MIPR model for each of these sets, were considered. While models with a larger number
of reactions would improve the fit of a model to the experimental data, the lack of information regarding possible reaction mechanisms made selection of a single reaction set difficult. Another consideration was to minimize the number of adjustable parameters. Thus the simplest possible parallel competitive reaction set (reaction set \( \{P\} \)) was believed to be the most suitable choice. The set of differential equations to be solved for Reaction set \( \{P\} \) are as follows:

\[
\begin{align*}
\frac{dA(t)}{dt} &= (k_b + k_p) A(t) \\
\frac{dB(t)}{dt} &= k_b A(t) \\
\frac{dP(t)}{dt} &= k_p A(t) \\
A(t=0) &= A_0 \\
B(t=0) &= P(t=0) = 0
\end{align*}
\]

where \( A \) is the quantity of starting material remaining to be converted, \( B \) is the quantity of side product produced, \( P \) is the quantity of product to be modeled, \( k_b \) and \( k_p \) are the reaction rate coefficients for the reactions producing \( B \) and \( P \) respectively, and \( A_0 \) is the minimum mass of \( A \) needed to produce product \( P(t=t) \) (i.e., the fraction of the total soil sample which can produce product \( P \) if there were no side reaction product \( B \)).

Substituting eq. (2) into equations 10-12 for non-isothermal reaction conditions,

\[
\begin{align*}
\frac{dA(t)}{dt} &= A(t) \left[ k_{po} \exp \left( \frac{-E_p}{RT(t)} \right) + k_{bo} \exp \left( \frac{-E_b}{RT(t)} \right) \right] \quad (15) \\
\frac{dB(t)}{dt} &= A(t) \left[ k_{bo} \exp \left( \frac{-E_b}{RT(t)} \right) \right] \quad (16) \\
\frac{dP(t)}{dt} &= A(t) \left[ k_{po} \exp \left( \frac{-E_p}{RT(t)} \right) \right] \quad (17)
\end{align*}
\]
This model utilizes up to 5 adjustable kinetic parameters: $k_{b0}$, $k_{p0}$, $E_b$, $E_p$, and $A_0$. Since calculation of the analytical solution for equations 13-17 was non-elementary, the FORTRAN subroutine LSODE (Livermore Solver for Ordinary Differential Equations) was used to solve for $A(t)$, $B(t)$, and $P(t)$ numerically. The non-linear least squares fitting algorithm used to fit $P(t)$ with the experimental data in this case was an implementation of the Levenberg-Marquardt Method via the FORTRAN subroutine MRQMIN [44]. A listing of the debugged code can be found in Appendix C.

4.4.3 Competitive Reaction Model - Series Reactions

The fundamental basis for examining this model was the possibility that the products quantified experimentally such as tars and light gases may be consumed to form other side products. An example is the possible competition between mass transfer of carbon dioxide out of the soil particle versus conversion to carbon monoxide cited in Section 3.1.2. Another possibility is the consumption of volatilized organic tars via secondary reactions, analogous to secondary reactions of coal tars to form light gases and light oils [38].

A two-reaction five-parameter model was chosen for the same considerations used for the parallel competitive reaction model. The set of differential equations to be solved for Reaction set $\{S\}$ are as follows:

\[
\begin{align*}
\text{A} & \rightarrow \text{Product} \rightarrow \text{B (side product)} \quad \{S\} \\
- \frac{dA(t)}{dt} &= k_p A(t) \\
\frac{dB(t)}{dt} &= k_b P(t) \\
\frac{dP(t)}{dt} &= k_p A(t) - k_b P(t)
\end{align*}
\]
\[ A(t=0) = A_0 \] \{21\}
\[ B(t=0) = P(t=0) = 0 \] \{22\}

where \( A \) is the quantity of starting material remaining to be converted, \( B \) is the quantity of side product produced, \( P \) is the quantity of product to be modeled, \( k_b \) and \( k_0 \) are the reaction rate coefficients for the reactions producing \( B \) and \( P \) respectively, and \( A_0 \) is the fraction of the total soil sample which can produce product \( P(t=0) \). Substituting eq. \{2\} into equations \{18\} - \{20\} for non-isothermal reaction conditions,

\[ - \frac{dA(t)}{dt} = A(t) \left[ k_{p0} \exp\left( - \frac{E_p}{R T(t)} \right) \right] \] \{23\}
\[ \frac{dB(t)}{dt} = P(t) \left[ k_{b0} \exp\left( - \frac{E_b}{R T(t)} \right) \right] \] \{24\}
\[ \frac{dP}{dt} = A(t) \left[ k_{p0} \exp\left( - \frac{E_p}{R T(t)} \right) \right] - P(t) \left[ k_{b0} \exp\left( - \frac{E_b}{R T(t)} \right) \right] \] \{25\}

Here, the adjustable kinetic parameters are \( k_{b0}, k_{p0}, E_b, E_p \), and \( A_0 \), and the "best fit" values were obtained using the FORTRAN subroutines LSODE and MRQMIN. Since the code is similar to that of the competitive parallel reaction model discussed in the previous subsection, a listing of only the modified subroutines can be found in Appendix C.

4.5 Data Analysis

For reasons discussed in Section 4.3, the study to model trends in the weight loss, volatilized tars, and light gases data began by examining the results of the "best fits" obtained by the multiple independent parallel reaction (MIPR) model. Due to the reasons which follow, analysis using the competitive parallel and series reaction schemes for
reaction models became necessary. The efficacy of the fits of these three models and their implications will also be discussed.

**MIPR Modeling Results**

In the past, the MIPR model has been used to model temperature and heating rate data obtained for coal [37-41]. Bucalá et al. [15] used this model to describe the trends in the effects of temperature on total weight loss from intense heating of soil in an inert atmosphere. The best fit obtained was found to be quite good, with the standard error of estimate between the model and experimental results to be within 1.2%.

Here, this model was applied to the heating rate data shown in Chapter 3. To determine whether the model could describe maxima or non-monotonic trends in heating rate data observed in Chapter 3, the model was first fit to heating rate data only, and then temperature data were included to ascertain whether the model can be applied for all data.

The test case chosen was whether the weight loss data for neat soil could be modeled by MIPR. The data show a constant weight loss from 100 - 1000°C/s before falling at 5000°C. Since many combinations of $k_0$ and $E_0$ can give the same model results, $k_0$ was assumed to be $10^{13}$/sec, because Darivakis [42] used this value in his soil weight loss modeling and previous investigators modeling coal pyrolysis using MIPR have selected this value (Ko [39], Boroson [47]). While this value may not be suitable for describing inorganic processes such as calcination, $10^{13}$/sec can at least be used to model product formation and new activation energies can be recalculated later based on better values for $k_0$. When heating rate data alone were used for a “best fit” (Figure 4-7), the absolute values predicted by the model were reasonably good, with scatter in the model predictions due to the scatter in the experimental peak temperatures in the temperature-time profiles used for the fit. However, the general trend predicted by the model was a slight
Figure 4-7. Experimental data and MIPR (multiple independent parallel reaction) model "best fit" values for weight loss of neat EPA soil as affected by heating rate. Initial soil mass = 50 mg, nominal peak temperature = 550°C, zero holding time.
decrease with increase heating rate. Addition of the temperature data to the fit only strengthened the decreasing trend.

**Competitive Reaction Model - Parallel Reactions (CRM-P)**

Since modeling heating rate trends became the focus in the MIPR work, this model was tested with acetylene data obtained for neat soil which showed a maximum in the yield at the intermediate heating rate. Like in MIPR modeling, $k_0$ was assumed to be $10^{13}$/sec. As Figure 4-8 shows, the model performs well predicting the absolute values of the data at the extreme heating rates, but underpredicts at 1000°C/s. The model trend shows a slight downward concavity, but the trend is not strong enough to conclude that this model describes the data well. Again the scatter in the model predictions is due to scatter in the experimental temperature-time profiles used in the model. When the temperature data were included with the heating rate data (Figure 4-9), the best fit values from the CRM-P model produced temperature and heating rate trends very different from those of the data. Hence this competitive parallel model cannot singly describe acetylene production from the pyrolysis of neat EPA soil.

**Competitive Reaction Model - Series Reactions (CRM-S)**

This model was also tested with acetylene data obtained for neat soil which showed a maximum in the yield. $k_0$ was again assumed to be $10^{13}$/sec. Figure 4-10 illustrates that the average of the values obtained by CRM-S model generally predicts the strong maximum with heating rate well, but the best-fit estimates exhibit significant scatter. As predicted acetylene production increases with increasing peak temperature, the scatter indicates that the model is very sensitive to temperature. For example, at 5000°C/s the experimentally
Figure 4-8. Experimental data and CRM-P (competitive reaction model - parallel reactions) model "best fit" values for acetylene yields from a neat EPA soil as affected by heating rate. Initial soil mass = 50 mg, nominal peak temperature = 550°C, zero holding time.
Figure 4-9. Experimental data and CRM-P (competitive reaction model - parallel reactions) model *best fit* values for acetylene yields ($A_0 = 8.16 \, \mu g$, $E_p = 55070 \, \text{cal/gmole}$, $E_b = 52140 \, \text{cal/gmole}$, $k_{po} = k_{bo} = 10^{13} \, /s$). Both temperature and heating rate data used for a single fit. Initial soil mass = 50 mg, nominal peak temperature = 550°C, zero holding time.
Figure 4-10. Experimental data and CRM-S (competitive reaction model -series reactions) model "best fit" values for acetylene from a neat EPA soil as affected by heating rate. Initial soil mass = 50 mg, nominal peak temperature = 550°C, zero hold time.
obtained peak temperatures were 550°C ± ~5°C due to difficulties in obtaining precise peak temperatures at such high heating rates. In runs where the peak temperature approached 500°C, the model predicts almost no acetylene production, while runs with peak temperatures near 600°C, the model either matched the data or overpredicted. This sensitivity to temperature may explain this worker’s inability to obtain a fit to the temperature data with this model.

Another interesting finding while working with this model was how the model trend changed with the adjustment of the value for \(A_o\), the minimum mass of natural soil sources necessary to produce acetylene. As the value for \(A_o\) was increased (Figure 4-11), the trend became increasingly similar to the trend for neat soil weight loss as affected by heating rate (Figure 4-7). The resulting activation energies were changed only slightly (~1 kcal). Since the EPA soil contains a significant amount of carbonates with carbon dioxide and carbon monoxide significantly contributing to weight loss, the finding that the CRM-S model can describe the weight loss trend may support the suggestion in Chapter 3 that carbon monoxide is formed in the soil by reactions:

\[
\begin{align*}
MCO_3 & \leftrightarrow MO + CO_2 \\
CO_2 + C & \rightarrow 2 CO 
\end{align*}
\]

\{26\} \quad \{27\}

Discussion

The modeling work with the MIPR, CRM-P, and CRM-S models strongly indicates that none of these reaction sets singly will describe heating rate effects on soil decomposition or (contaminated) soil weight loss and volatiles production. The MIPR (or the single reaction model, a simplification of MIPR where \(\alpha=0\)) model predicts well the temperature data trends, but is unable to do so for the heating rate data. The CRM-P and CRM-S models cannot describe temperature data, while succeeding with the heating rate
Figure 4-11. CRM-S (competitive reaction model - series reactions) model "best fit" values as affected by Ao, the minimum mass of soil constituent necessary for acetylene production. Initial soil mass = 50 mg, nominal peak temperature = 550°C, zero holding time.
data to varying degrees. CRM-S predicts the trends well, but is very sensitive to temperature. CRM-P is not as temperature sensitive, but cannot predict strong changes in volatiles production. The limited success of the three models suggests that each reaction set may contribute to the overall reaction mechanism leading to the production of volatiles.

However, the inadequacies of the models may be due to their inherent assumptions. All of the models have assumed that mass transfer effects are negligible, providing unrestricted flow of the products out of and away from the soil particle. At first inspection of the heating rate data, the maximum yield of CO$_2$ at 1000°C/s indicates that calcination occurred to the greatest extent at this heating rate. Since increased calcination is expected to raise the porosity of the soil particle, mass transfer limitations should be at a minimum at 1000°C/s. Since a maximum at 1000°C/s is a pervasive pattern in much of the heating rate data (e.g. CH$_4$, C$_2$H$_2$, and C$_2$H$_4$), mass transfer effects on the modeling results cannot be eliminated on this basis.

However if carbon monoxide is indeed produced via reaction (27) and products release from within the soil particle is mass transfer limited, an equilibrium CO$_2$/CO ratio would be expected. Using this hypothesis to ascertain whether mass transfer is limiting, an equilibrium calculation was carried out for a few select temperatures (1000°C/s nominal heating rate) using the CO$_2$/CO ratios obtained experimentally to determine the equilibrium pressure within the particle. As calculated in Appendix E for 800, 1000, and 1200 K, the pressures needed for equilibrium at the observed CO$_2$/CO ratios (for both neat and contaminated soils) are of order $10^9$ - $10^{10}$ atm., which is probably far greater than the pressures sustainable within the soil particle. Hence on the basis of this calculation, it appears that intraparticle mass transfer limitations are not important.

Other assumptions to consider are those regarding the distribution of activation energies for reactions in the models. The MIPR model assumes a normal distribution of activation energies, while the CRM-P and CRM-S models assume a single activation energy can represent the potentially many reactions leading to the formation of the observed
products. Since the true distribution of activation energies is unknown, the effects of this assumption is unknown.

Summary

Three models were used in an effort to describe the trends in the temperature and heating data shown in Chapter 3: the multiple independent parallel reaction model (MIPR), and two two-reaction competitive reaction schemes (parallel and series reaction models). The MIPR model modeled the temperature data well, while the competitive reaction models modeled the heating rate data best. These results suggest that none of these models singly will describe the reaction mechanisms leading to the trends in the observed products, but that each model may contribute to the overall reaction mechanism. The inadequacies of the models could also be due to the failure of assumptions such as negligible mass transfer limitations and assumptions about activation energy distributions. An equilibrium calculation based on experimental CO₂/CO ratios at selected temperatures suggests that intraparticle mass transfer is not limiting volatiles release from soil, but the effect of the latter assumption is currently unknown.

4.6 References


43. Bucalá, Verónica, Planta Piloto de Ingeniería Química (PLAPIQUI), 12 de Octubre 1842, (8000) Bahía Blanca, ARGENTINA.


Chapter 5

Conclusions

Regarding removal of pyrene (and by induction, physico-chemically similar PAH contaminants) from soils by rapid heating under an inert atmosphere, this work has shown conclusively that:

- Temperature strongly influences rates and extents of contaminants removal (calculated by weight loss), and the yields and release rates of volatiles.
- Yields of several light gases, e.g. CO, CO₂, C₂H₂, CH₄, and possibly C₂H₄ from pyrene contaminated soil, pass through maxima with increasing heating rate, while ultimate pyrene removal efficiency is little affected.
- The presence of contaminant on the soil can lead to off-gas compositions significantly different from those obtained by heat treatment of uncontaminated soil, e.g., in the present case of pyrene contamination of an EPA Synthetic Soil, CO production was strongly enhanced and C₂H₂ showed modest increases for some heating conditions. This effect is apparently a consequence of chemical reactions between or among contaminant(s) and soil, or their decomposition products at elevated temperatures, e.g., above ~500°C at a heating rate of 1000°C/s.
- Thermal treatment can substantially reduce, and potentially virtually eliminate pyrene contamination in soils. Thus controlled thermal treatment should also provide high extents of removal from soils of other polycyclic aromatics similar in volatility and chemical reactivity to pyrene.
• At higher temperatures (e.g. around 530°C at 1000°C/s), extents of
decontamination, calculated by comparing weight loss of contaminated soil
(corrected for weight loss from the soil itself) to the initial weight of
contaminant, may exceed 100%. The apparent explanation is that chemical
reactions between the soil and the contaminant, (or contaminant reaction
products) cause additional volatilization of the soil. Thus, soil-contaminant
reactions may cause gravimetric measurements to overestimate the true level
of contaminant removal. Where possible, individual products yields as well
as elemental and total material balances should be determined.

The results of this work also strongly suggest that:

• Individual light gas products from the thermal treatment of neat and contaminated
soils are formed by a very complex set reactions. Modeling of the heating
rate data suggest products are formed via competitive reactions at a specific
temperature, while temperature data modeling indicates the presence of one
or more independent parallel reactions.

• Pyrene, and presumably other polycyclic aromatic compounds of similar volatility
and chemical reactivity, can undergo chemical modifications during thermal
desorption from soil at higher temperatures, e.g., at or above ~500°C at a
nominal heating rate of 1000°C/s

• It is further tentatively concluded that there is potential for biologically active
products to be formed from soil treatment, either from decomposition of the
contaminant itself or through chemical reactions between soil or soil
decomposition products, and a PAH contaminant. Bioactivity testing and
detailed chemical analyses of higher molecular weight products (e.g. tars)
are needed to further test this inference.
Chapter 6

Recommendations

This work has raised the issues of contaminant transformation and contaminant-soil constituent chemical reactions occurring during thermal treatment of PAH-contaminated soils as well as proposing possible reactions and mechanisms leading to the formation of individual species. To examine these issues more closely and confirm more conclusively the conclusions of this work, possible follow-up studies to this work are:

• Chemical analysis and bioassays of the tars produced and soil extracts: This will allow for the calculation of a carbon balance to determine quantitatively how the contaminant is distributed between light gases, tars and soil during thermal treatment, identify the presence of any known carcinogens or mutagens in these fractions, and illuminate chemical pathways and mechanisms leading to the formation of individual species. Based on the chemical analysis, selected samples can then be submitted for bioassays to determine their toxicity and mutagenicity to determine possible impacts of these fractions on human health, as health impacts from operation of thermal treatment facilities and the resulting treated soil are of great public concern.

• Particle size and pressure effects studies: Investigation of the effects of these parameters will determine the effects mass transfer on PAH removal efficiencies and off-gas composition.

• Contamination level studies: By varying the contamination levels, soil-contaminant reactions hypothesized can be confirmed by studying changes in light gases such as carbon monoxide as affected by contamination level.
• Soil or pure mineral studies: Since the soil types and soil makeup can vary widely within a site as well as between sites, companion studies to this one using different soils or pure minerals would allow for the examination mineral effects, the contribution of soil organics to the reaction products, and possible identification other soil-contaminant interactions.

• Contaminant studies: Studies with other contaminants as well as contaminant mixtures would be helpful in determining the operating conditions required to remediate these types of soils. Studies of the off-gases could also shed some new light on soil-contaminant interactions.

• Moisture effects: Moisture has been identified to enhance contaminant removal, possibly via a “steam distillation” mechanism. Hydrolysis reactions occurring to a significant degree are also possible at elevated temperatures. Potential contributions of each factor could be illuminated in a study.
Appendix A

Detailed Experimental Run Procedure
for the Screen Heater Reactor (Soil Work)

Phase I - Prefire foil/begin weighing experimental materials

- Rinse foil (if not already washed) with methylene chloride, and let condensed moisture evaporate
- Mount foil in reactor with dip in center (first tighten left clamp with no foil is sticking out of the left side of the clamp, place dip in foil until no foil is sticking out of right side of the right clamp, and tighten right clamp)
- Close reactor, and finger-tighten nuts and bolts to fasten lid
- Remove traps, if any, connected to system (including helium prefilter)
- Flush reactor four times
  - Flush 1
    - Just pull vacuum on reactor (don’t wait for it to go to -30 in Hg)
    - Weigh out tar filter and place in tar filter casing and tighten casing
    - Tighten four sets of nuts and bolts on reactor to finger tightness to create best seal.
    - Fill reactor with helium slowly to 7-8 psig (over a period of ~1.5 minutes)
  - Flush 2
    - Slowly pull vacuum on reactor (don’t wait for it to go to -30 in Hg)
    - Weigh out tar filter funnel and attach to tar filter casing with clamp
    - Clean outside surface of a preconditioned condensables trap with methylene chloride dampened Kimwipe and set down to dry
    - Fill reactor with helium slowly to 7-8 psig (same rate as above)
  - Flush 3
    - Slowly pull vacuum on reactor (don’t wait for it to go to -30 in Hg)
    - Weigh condensables trap
    - Fill reactor with helium relatively slowly to 7-8 psig
  - Flush 4
    - Slowly pull vacuum on reactor (wait for it to go to -30 in Hg)
    - Fill reactor with helium slowly to 7-8 psig
- Prefire foil
  - Set heating variac to 16, and heating variac timer to 0.7 seconds. All other timers should be set to zero.
  - Fire foil (peak temperature of at least 850°C should be attained)
  - Pull vacuum on system (if desired) to remove trace quantities of organics and fill reactor slowly to 0 psig, OR just reduce pressure in system to 0 psig with vacuum.
  - Open reactor
  - Remove foil and make impression in it for soil. Let sit on clean foil surface until needed in Phase II
Phase II - Finish weighing experimental materials

- Bottom reactor liner
  - weigh
  - immediately place in reactor
- Top reactor liner
  - weigh
  - make cuts to make room for the electrodes
    - pick a fold going across entire liner
    - cut 1" deep lines (parallel to that fold) which are 1 1/16" from the fold (so that you have a pair of 1" cuts at each end)
    - fold 2 1/8" flaps down
  - cut slit to go around the thermocouple feedthrough
    - Hold liner so that have two tabs folding downward
    - At left side at an angle of 45° towards the front of foil, make a 1 1/2" cut.
  - mount liner into reactor so that thermocouple feedthrough is wrapped within the 1 1/2" cut, and the electrodes sit in downwardly folded 2 1/8" flaps. Use an old piece of tubing with length-wise slit to hold the liner up near top of reactor.
- Weigh foil and ~50 ± 3 mg soil into foil (going outside this range may significantly affect the temperatures and heating rates obtained)
- Mount foil into reactor so that the impression in the foil is parallel to the reactor bottom
- Take tar filter casing with the funnel already attached, and mount it above the foil using a piece of rubber tubing to hold them in place
- Close reactor gently
- Put into place the four sets of nuts and bolts and hand-tighten
- Add condensables (just in air) and light gas trap (in an unchilled dewar) to the outlet line from the reactor
- Add helium prefilter (in an unchilled dewar) to helium inlet

Phase III - Flushing reactor and Establishing flow
- Flush 1
  - Pull vacuum on reactor slowly to not overtax the pump
  - When vacuum is at -30 in Hg, close valve 1 (at He inlet to reactor)
  - Add liquid nitrogen to helium pretrap ONLY, until it is full
  - Open rotameter valve a couple of turns so that get gas into presfilter area
  - Open valve 1 very slowly until just get gas entering the reactor and fine tune rotameter so that the bottom of the ball is at 10.
  - Fill reactor at this rotameter setting until 7-8 psig. Adjust rotameter as necessary to maintain right setting.
  - Stop flow by closing valve 1. DON'T TOUCH THE ROTAMETER
- Flush 2
  - Pull vacuum on reactor slowly until -30 in Hg
  - Pour liquid nitrogen into helium pretrap to top
  - Open valve 1 SLOWLY until just get gas entering and adjust flow with rotameter to get bottom of ball at 10.
  - Fill reactor to 7-8 psig at this rotameter setting
  - Stop flow by closing valve 1. DON'T TOUCH THE ROTAMETER.
- Flushes 3 and 4
  - Close valve 2 and leave valve 3 wide open
  - Pull helium through traps (open vacuum pump valve VERY SLOWLY) for 10-15 minutes each (pressure should drop from 7 to approx. 3 psig over this time)
  - Open valve 2 SLOWLY and get reactor vacuum to -30 in Hg
  - Fill reactor to 7-8 psig as in Flushes 1 and 2
- Flushes 5 and 6
  - Use procedure for Flush 2
- Flush 7
  - Pull vacuum on reactor slowly until -30 in Hg
  - Pour liquid nitrogen into helium pretrap to top
  - Close valve 2 and 3
  - Fill reactor with the rotameter setting with the bottom of the ball at 2 until you get 3 psig. Process takes approximately 25 minutes.
    - Refill the helium pretrap every 10 minutes.
    - When reactor pressure is at 0 psig, fill the light gas trap dewar with liquid nitrogen to chill it.
    - DO NOT CLOSE VALVE 1 WHEN GET TO PRESSURE
- Establish helium gas flow
  - Open valve 3 at the most 2 full turns from closed position
  - SLOWLY open valve for vacuum pump
  - Stabilize flow with valve 3 until a stable pressure is obtained for 3 minutes (Pressure should be positive and as near to 3 psig as possible). This step is CRITICAL to prevent passing gases through the traps too quickly giving us low capture efficiencies. (0.5 l/minute flow rate)

Phase IV - Firing the soil in the foil

- Set the heating and hold variacs as well as the heating and hold timers to determined settings.
- Prepare the Bascom-Turner recorder for taking measurement
  - Make sure full scale is 50 mV and thermocouple leads placed into channel 1
  - Set plotting parameters by using following key strokes
    - STATUS-7-0-4-GO (should see "0 4" on digital readout) Setting the no. of data points to acquire to 2000.
- STATUS-7-1-4-GO (should see “14” on digital readout) Setting the no. of data points plotted per page to 2000.
- STATUS-1-1-1-0-GO (should see “10” on digital readout) Setting the sampling time to 10 milliseconds (10 ms/point)

- Turn on the electronic ice point.
- Press STATUS-4-1 to see if readout is ~0.20 (i.e. ~25°C or room temperature). If this is so, the thermocouple is working.
- Refill to top with liquid nitrogen both the light gas trap and helium prefilter dewar flasks.
- Turn on MAIN POWER SWITCH of reactor
- Press on recorder ACQ-1 (should see “Ac 1” on digital readout)
- Turn on POWER switch (should see light go on)
- Press on recorder GO
- Turn RUN switch on to fire the reactor. Watch the show!
- After the recorder is done recording the data, turn off the RUN, POWER, and MAIN POWER SWITCHES.
- To save the recorder data on disk, press on the recorder DISK-2-GO. A number will come onto the digital readout (“d2 XXX). Record this number (XXX) as it is the data file number for this run on this disk. (Also note disk number, which is on the label of the diskette in the drive).
- Turn off the electronic ice point and the Bascom-Turner recorder.

Phase V - Two hour reactor purge

- Every 10 minutes refill liquid nitrogen in light gas trap and helium prefilter dewar flasks
- Periodically monitor and adjust the reactor pressure and Valves 1 & 3 (as necessary) to be sure that you have the correct gas flows to maintain a constant pressure.
  - If the reactor pressure is holding steady above 3 psig, that's okay. At end of run, we'll just turn off the helium at the end of 2 hours and let reactor pressure drop to 3 psig before closing the vacuum pump valve.
  - If the reactor pressure is holding steady at significantly below 3 psig (like 1 psig), then close vacuum pump valve (NOT valve 3) until pressure gets to 3 psig and then re-open valve.
  - If the reactor pressure is continuing to rise and reaches 6-7 psig, close valve 1 and open valve 3 more (use small increments on the order of 1/16 turn, unless pressure rise is really fast then I'd use 1/8 turns). Record the amount of time valve 1 is closed and add this to the 2 hour purge time so that maintain constant amount gas used to purge the reactor in each run.
  - If the reactor pressure is continuing to drop and reaches 0-1 psig, close the vacuum pump valve and close valve 3 (by 1/8 of turn if drop is slow, or 1/4 turn if drop is fast). This case means that gas flow through the light gas trap was too fast, and hence the results of this run may yield low gas numbers. NOTE THIS IN YOUR NOTEBOOK IF THIS CASE ARISES. Continue with run and we'll see later if the data make any sense.
  - 30 minutes before the end of the run, start boiling water for trap preconditioning (2 liters water in an Erlenmeyer flask on the hot plate set at HI on the dial)

Phase VI - Post-run weighing/cleanup & Preparation for next run

- At the end of two hours, close valve 1 and the vacuum pump valve.
- Remove the light gas trap and the condensables trap. Set aside light gas trap for GC analysis later.
- Loosen the clamp holding the dewar for the helium pretrap, and detach this trap from the equipment panel.
- Pour liquid nitrogen from both dewars into the 2 l dewars for storage for the next run
- Open valve 2 and with the vacuum pump, reduce the reactor pressure to 0 psig.
- Open reactor after loosening the nuts and bolts, and remove tar filter & funnel from above soil and set aside.
- WEIGH SOIL FIRST (avoid collecting moisture in air as much as possible)
- Reactor glass wipedown
  - Weigh two 7.0 cm dia. glass filter papers (Whatman GF/A 7.0cm)
  - Take first, fold it in half and wet with methylene chloride. Wipe top half of reactor.
  - Fold first filter paper the other way and wipe the other half of the reactor.
  - Repeat process with the second filter
  - Let the methylene chloride evaporate for 20-25 minutes before weighing the “dirty” glass filter papers. IN THE MEANTIME, see next step and Phase VII.
- Weigh remainder of items
  - remove and weigh reactor liners (top and bottom)
  - weigh condensables trap (wipe with methylene chloride dampened Kimwipe and let dry before weighing, just like in Phase I)
  - weigh tar filter funnel
  - weigh tar filter after carefully removing filter from its casing

Phase VII - Trap preconditioning

- Take the boiling water from the 2 l Erlenmeyer flask and transfer it to the metal beaker.
- Move the hotplate to the equipment panel and place it below where the helium prefilter would go. Turn the hotplate on HI, and place the metal beaker with water on the hotplate.
- Take the helium prefilter trap and a light gas trap which has already had its contents analyzed (i.e. NOT THE LIGHT GAS TRAP YOU JUST FINISHED WITH) and attach it to the equipment panel and reactor (respectively) so that the two traps sit in the boiling water.
- Make sure valves 1, 2, and 3 are ALL OPEN and the rotameter valve and “Main Helium” valve on the equipment panel are BOTH OFF.
- Pull a vacuum on the reactor to -30 in Hg (heat + vacuum removes water, etc.)
- Precondition for 30 minutes
- End of Day
  - turn off hotplate
  - remove traps (keeping vacuum)
  - Leave reactor purging for next user
    OR
    - turn off pump, open vent and reactor vacuum valves and let in air into reactor to 0 psig and open reactor for next user.
  - Wash tar filter case for next user (rinse with methylene chloride and allow to dry)
Appendix B

Tables of Experimental Data
from the Temperature and Heating Rate Studies

Contents of this section

1. Raw data       Information such as initial and final mass of soil, gas chromatographic (GC) peak areas for the light gases, contributions towards total tars from each individual source, condensables quantities, and GC calibration curves

2. Processed data Tabulation of calculated values such as percentage weight loss, total tars, quantities of each light gas, and mass of condensables

3. Normalized data Processed data for neat EPA soil was normalized on the basis of 50 mg of soil initially. Contaminated EPA soil was normalized on the basis of 50 mg neat soil + 2.49 mg pyrene initially. This section also contains a detailed discussion of the statistical analysis of the experimental data.

LOCATION OF THE PRIMARY DATA: All experimental notebooks, notebooks on the modeling work, gas chromatographic data (paper copies and diskettes) can be found in the laboratory of Room. 66-059. Professor Jack B. Howard and Dr. William A. Peters have been notified of their location.
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Blank EPA Soil Data - Temperature Study (no hold at real temperature)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>(Raw Data in Absolute Numbers)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Run Date</td>
<td>Temperature (°C)</td>
<td>Heating Time (sec)</td>
<td>Heating Rate (°C/s)</td>
<td>Hold Temp. (°C)</td>
<td>Hold Time (sec)</td>
<td>Soil Weight (Init., mg)</td>
<td>Soil Weight (Final, mg)</td>
<td>Tar Filter (mg)</td>
<td>Tar Filter Funnel (mg)</td>
<td>Tar Liners (mg)</td>
</tr>
<tr>
<td>8</td>
<td>1/17/94</td>
<td>554</td>
<td>0.50</td>
<td>1100</td>
<td>0</td>
<td>0</td>
<td>50.43</td>
<td>49.04</td>
<td>0.00</td>
<td>0</td>
<td>0.46</td>
</tr>
<tr>
<td>9</td>
<td>6/9/93</td>
<td>518</td>
<td>0.50</td>
<td>988</td>
<td>0</td>
<td>0</td>
<td>50.71</td>
<td>52.48</td>
<td>0.03</td>
<td>0.00</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>7/19/93</td>
<td>518</td>
<td>0.50</td>
<td>988</td>
<td>0</td>
<td>0</td>
<td>50.71</td>
<td>52.48</td>
<td>0.03</td>
<td>0.00</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>6/19/94</td>
<td>542</td>
<td>0.60</td>
<td>885</td>
<td>0</td>
<td>0</td>
<td>51.32</td>
<td>49.29</td>
<td>0.07</td>
<td>0</td>
<td>1.5</td>
</tr>
<tr>
<td>12</td>
<td>7/22/93</td>
<td>555</td>
<td>0.60</td>
<td>930</td>
<td>0</td>
<td>0</td>
<td>50.19</td>
<td>47.24</td>
<td>0.01</td>
<td>0.01</td>
<td>0.15</td>
</tr>
<tr>
<td>13</td>
<td>1/26/94</td>
<td>582</td>
<td>0.60</td>
<td>1003</td>
<td>0</td>
<td>0</td>
<td>49.89</td>
<td>48.38</td>
<td>0.09</td>
<td>0.10</td>
<td>0.54</td>
</tr>
<tr>
<td>14</td>
<td>6/4/93</td>
<td>826</td>
<td>0.60</td>
<td>898</td>
<td>0</td>
<td>0</td>
<td>49.68</td>
<td>47.16</td>
<td>0.05</td>
<td>0.08</td>
<td>0.17</td>
</tr>
<tr>
<td>15</td>
<td>6/5/93</td>
<td>718</td>
<td>0.60</td>
<td>926</td>
<td>0</td>
<td>0</td>
<td>50.69</td>
<td>48.38</td>
<td>0.09</td>
<td>0.10</td>
<td>0.54</td>
</tr>
<tr>
<td>16</td>
<td>8/19/94</td>
<td>747</td>
<td>0.60</td>
<td>930</td>
<td>0</td>
<td>0</td>
<td>50.32</td>
<td>48.38</td>
<td>0.09</td>
<td>0.10</td>
<td>0.54</td>
</tr>
<tr>
<td>17</td>
<td>1/21/94</td>
<td>750</td>
<td>0.60</td>
<td>995</td>
<td>0</td>
<td>0</td>
<td>50.03</td>
<td>48.38</td>
<td>0.09</td>
<td>0.10</td>
<td>0.54</td>
</tr>
<tr>
<td>18</td>
<td>1/28/94</td>
<td>803</td>
<td>0.60</td>
<td>930</td>
<td>0</td>
<td>0</td>
<td>51.03</td>
<td>48.38</td>
<td>0.09</td>
<td>0.10</td>
<td>0.54</td>
</tr>
<tr>
<td>19</td>
<td>1/17/94</td>
<td>848</td>
<td>0.60</td>
<td>926</td>
<td>0</td>
<td>0</td>
<td>50.45</td>
<td>48.38</td>
<td>0.09</td>
<td>0.10</td>
<td>0.54</td>
</tr>
<tr>
<td>20</td>
<td>12/4/93</td>
<td>898</td>
<td>0.60</td>
<td>926</td>
<td>0</td>
<td>0</td>
<td>50.58</td>
<td>48.38</td>
<td>0.09</td>
<td>0.10</td>
<td>0.54</td>
</tr>
<tr>
<td>21</td>
<td>1/20/94</td>
<td>931</td>
<td>0.60</td>
<td>926</td>
<td>0</td>
<td>0</td>
<td>50.58</td>
<td>48.38</td>
<td>0.09</td>
<td>0.10</td>
<td>0.54</td>
</tr>
<tr>
<td>22</td>
<td>2/15/93</td>
<td>954</td>
<td>0.60</td>
<td>926</td>
<td>0</td>
<td>0</td>
<td>49.01</td>
<td>48.38</td>
<td>0.09</td>
<td>0.10</td>
<td>0.54</td>
</tr>
<tr>
<td>23</td>
<td>1/29/94</td>
<td>955</td>
<td>0.60</td>
<td>926</td>
<td>0</td>
<td>0</td>
<td>50.00</td>
<td>48.38</td>
<td>0.09</td>
<td>0.10</td>
<td>0.54</td>
</tr>
<tr>
<td>24</td>
<td>7/14/94</td>
<td>962</td>
<td>0.60</td>
<td>926</td>
<td>0</td>
<td>0</td>
<td>50.58</td>
<td>48.38</td>
<td>0.09</td>
<td>0.10</td>
<td>0.54</td>
</tr>
<tr>
<td>25</td>
<td>7/15/94</td>
<td>972</td>
<td>0.60</td>
<td>926</td>
<td>0</td>
<td>0</td>
<td>50.58</td>
<td>48.38</td>
<td>0.09</td>
<td>0.10</td>
<td>0.54</td>
</tr>
<tr>
<td>26</td>
<td>7/16/94</td>
<td>1019</td>
<td>0.60</td>
<td>926</td>
<td>0</td>
<td>0</td>
<td>50.58</td>
<td>48.38</td>
<td>0.09</td>
<td>0.10</td>
<td>0.54</td>
</tr>
<tr>
<td>27</td>
<td>1/28/94</td>
<td>1033</td>
<td>1.25</td>
<td>807</td>
<td>987</td>
<td>5</td>
<td>50.46</td>
<td>48.38</td>
<td>0.09</td>
<td>0.10</td>
<td>0.54</td>
</tr>
<tr>
<td>28</td>
<td>1/27/94</td>
<td>1033</td>
<td>1.25</td>
<td>807</td>
<td>1065</td>
<td>5</td>
<td>50.46</td>
<td>48.38</td>
<td>0.09</td>
<td>0.10</td>
<td>0.54</td>
</tr>
<tr>
<td>29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>All runs between 0 mg and -0.10 mg are considered 0 mg (based on errors of reading balance)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>Experiment key: S=Salto, E=Eric Freeman, V=Veronica Bucak</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>V. Bucak's raw data was unavailable at the time of compilation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>EPA Soil Data - Temperature Study (no hold at real temperature)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>Raw Data in Absolute Numbers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>Run Date</td>
<td>Temperature (°C)</td>
<td>Heating Time (sec)</td>
<td>Heating Rate (°C/s)</td>
<td>Hold Temp. (°C)</td>
<td>Hold Time (sec)</td>
<td>Soil Weight (Init., mg)</td>
<td>Soil Weight (Final, mg)</td>
<td>Tar Filter (mg)</td>
<td>Tar Filter Funnel (mg)</td>
<td>Tar Liners (mg)</td>
</tr>
<tr>
<td>42</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
<td>G</td>
<td>H</td>
<td>I</td>
<td>J</td>
<td>K</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>62</td>
<td>Rand EP A Soil Data - Heating Rate Study (no heat at initial temperature)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>(peak temperature = 550°C)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>Key: <strong>Italics</strong> = Deduced from similar runs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>65</td>
<td><strong>Bold</strong> = Negative weight measured</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>66</td>
<td>(Raw Data in Absolute Numbers)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>67</td>
<td>Run Date</td>
<td>Temperature (°C)</td>
<td>Heating Time (sec)</td>
<td>Heating Rate (°C/s)</td>
<td>Hold Temp. (°C)</td>
<td>Hold Time (sec)</td>
<td>Soil Weight (init, mg)</td>
<td>Soil Weight (final, mg)</td>
<td>Tar Filter (mg)</td>
<td>Tar Filter Funnel (mg)</td>
<td>Tar Lines (mg)</td>
</tr>
<tr>
<td>68</td>
<td>6/2/93 H</td>
<td>586</td>
<td>0.80</td>
<td>1003</td>
<td>0</td>
<td>0</td>
<td>49.85</td>
<td>48.54</td>
<td>0.09</td>
<td>0.10</td>
<td>0.54</td>
</tr>
<tr>
<td>69</td>
<td>6/3/93 H</td>
<td>542</td>
<td>0.80</td>
<td>863</td>
<td>0</td>
<td>0</td>
<td>54.22</td>
<td>52.64</td>
<td>0.03</td>
<td>0.00</td>
<td>0</td>
</tr>
<tr>
<td>70</td>
<td>7/2/93 H</td>
<td>555</td>
<td>0.60</td>
<td>893</td>
<td>0</td>
<td>0</td>
<td>51.32</td>
<td>49.79</td>
<td>0.07</td>
<td>0</td>
<td>0.61</td>
</tr>
<tr>
<td>71</td>
<td>7/22/93 H</td>
<td>582</td>
<td>0.60</td>
<td>530</td>
<td>0</td>
<td>0</td>
<td>50.19</td>
<td>47.74</td>
<td>0.01</td>
<td>0.15</td>
<td>0</td>
</tr>
<tr>
<td>72</td>
<td>7/26/93 H</td>
<td>582</td>
<td>0.60</td>
<td>930</td>
<td>0</td>
<td>0</td>
<td>50.19</td>
<td>47.74</td>
<td>0.01</td>
<td>0.15</td>
<td>0</td>
</tr>
<tr>
<td>73</td>
<td>New Calibration (since Jan '94)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>74</td>
<td>3/1/94 H</td>
<td>586</td>
<td>0.00</td>
<td>99</td>
<td>0</td>
<td>0</td>
<td>50.66</td>
<td>48.80</td>
<td>0.01</td>
<td>0.01</td>
<td>0.25</td>
</tr>
<tr>
<td>75</td>
<td>3/11/94 H</td>
<td>507</td>
<td>0.00</td>
<td>81</td>
<td>0</td>
<td>0</td>
<td>50.47</td>
<td>48.58</td>
<td>0.00</td>
<td>0.00</td>
<td>0.32</td>
</tr>
<tr>
<td>76</td>
<td>4/2/94 H</td>
<td>507</td>
<td>0.00</td>
<td>81</td>
<td>0</td>
<td>0</td>
<td>50.47</td>
<td>48.58</td>
<td>0.00</td>
<td>0.00</td>
<td>0.32</td>
</tr>
<tr>
<td>77</td>
<td>5/3/94 H</td>
<td>518</td>
<td>0.11</td>
<td>4491</td>
<td>0</td>
<td>0</td>
<td>50.66</td>
<td>49.99</td>
<td>0.00</td>
<td>0.00</td>
<td>0.17</td>
</tr>
<tr>
<td>78</td>
<td>5/5/94 H</td>
<td>590</td>
<td>0.11</td>
<td>4600</td>
<td>0</td>
<td>0</td>
<td>51.15</td>
<td>50.42</td>
<td>0.03</td>
<td>0.00</td>
<td>0.21</td>
</tr>
<tr>
<td>79</td>
<td>5/15/94 H</td>
<td>507</td>
<td>0.11</td>
<td>5500</td>
<td>0</td>
<td>0</td>
<td>50.90</td>
<td>48.42</td>
<td>0.01</td>
<td>0.00</td>
<td>0.14</td>
</tr>
<tr>
<td>80</td>
<td>5/21/94 H</td>
<td>521</td>
<td>0.09</td>
<td>5522</td>
<td>0</td>
<td>0</td>
<td>50.34</td>
<td>49.80</td>
<td>0.05</td>
<td>0.00</td>
<td>0.21</td>
</tr>
<tr>
<td>81</td>
<td>6/2/94 H</td>
<td>535</td>
<td>0.11</td>
<td>4910</td>
<td>0</td>
<td>0</td>
<td>50.66</td>
<td>49.99</td>
<td>0.00</td>
<td>0.00</td>
<td>0.17</td>
</tr>
<tr>
<td>82</td>
<td>6/5/94 H</td>
<td>507</td>
<td>0.11</td>
<td>5500</td>
<td>0</td>
<td>0</td>
<td>50.90</td>
<td>48.42</td>
<td>0.01</td>
<td>0.00</td>
<td>0.14</td>
</tr>
<tr>
<td>83</td>
<td>6/3/94 H</td>
<td>506</td>
<td>0.10</td>
<td>6200</td>
<td>0</td>
<td>0</td>
<td>50.24</td>
<td>49.55</td>
<td>0.00</td>
<td>0.00</td>
<td>0.21</td>
</tr>
<tr>
<td>84</td>
<td>85</td>
<td>86</td>
<td>87</td>
<td>88</td>
<td>89</td>
<td>90</td>
<td>91</td>
<td>92</td>
<td>93</td>
<td>94</td>
<td>95</td>
</tr>
<tr>
<td>90</td>
<td>Rand EP A Soil Data - Heating Rate Study (no heat at initial temperature)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>91</td>
<td>Peak temperature = 550°C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>92</td>
<td>Key: <strong>Italics</strong> = Deduced from similar runs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>93</td>
<td><strong>Bold</strong> = Negative weight measured</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>94</td>
<td>(Raw Data in Absolute Numbers)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>95</td>
<td>Run Date</td>
<td>Temperature (°C)</td>
<td>Heating Time (sec)</td>
<td>Heating Rate (°C/s)</td>
<td>Hold Temp. (°C)</td>
<td>Hold Time (sec)</td>
<td>Soil Weight (init, mg)</td>
<td>Soil Weight (final, mg)</td>
<td>Tar Filter (mg)</td>
<td>Tar Filter Funnel (mg)</td>
<td>Tar Lines (mg)</td>
</tr>
<tr>
<td>96</td>
<td>Calibrated Curve (Jan '94)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>97</td>
<td>5/11/93 H</td>
<td>572</td>
<td>0.40</td>
<td>857</td>
<td>0</td>
<td>0</td>
<td>51.31</td>
<td>47.78</td>
<td>0.74</td>
<td>0.09</td>
<td>1.01</td>
</tr>
<tr>
<td>98</td>
<td>5/15/93 H</td>
<td>598</td>
<td>0.60</td>
<td>857</td>
<td>0</td>
<td>0</td>
<td>49.63</td>
<td>45.92</td>
<td>0.69</td>
<td>0.11</td>
<td>0.99</td>
</tr>
<tr>
<td>99</td>
<td>5/20/93 H</td>
<td>548</td>
<td>0.00</td>
<td>87</td>
<td>0</td>
<td>0</td>
<td>51.46</td>
<td>47.85</td>
<td>0.61</td>
<td>0.04</td>
<td>0.93</td>
</tr>
<tr>
<td>100</td>
<td>6/1/93 H</td>
<td>570</td>
<td>0.00</td>
<td>91</td>
<td>0</td>
<td>0</td>
<td>50.17</td>
<td>46.53</td>
<td>0.63</td>
<td>0.06</td>
<td>0.92</td>
</tr>
<tr>
<td>101</td>
<td>5/11/94 H</td>
<td>588</td>
<td>0.11</td>
<td>5177</td>
<td>0</td>
<td>0</td>
<td>50.09</td>
<td>47.68</td>
<td>0.82</td>
<td>0.03</td>
<td>0.89</td>
</tr>
<tr>
<td>102</td>
<td>5/15/94 H</td>
<td>588</td>
<td>0.10</td>
<td>5740</td>
<td>0</td>
<td>0</td>
<td>50.04</td>
<td>47.57</td>
<td>0.81</td>
<td>0.00</td>
<td>0.80</td>
</tr>
<tr>
<td>103</td>
<td>5/25/94 H</td>
<td>596</td>
<td>0.11</td>
<td>5200</td>
<td>0</td>
<td>0</td>
<td>49.91</td>
<td>47.75</td>
<td>0.59</td>
<td>0.12</td>
<td>0.60</td>
</tr>
<tr>
<td>H</td>
<td>M</td>
<td>O</td>
<td>P</td>
<td>Q</td>
<td>R</td>
<td>S</td>
<td>T</td>
<td>U</td>
<td>V</td>
<td>W</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>Condensables Trap (mg)</td>
<td>CO Area Units</td>
<td>CO2 Area Units</td>
<td>H2O Area Units</td>
<td>CH4 Area Units</td>
<td>C2H6 Area Units</td>
<td>C2H4 Area Units</td>
<td>C2H6 Area Units</td>
<td>Soil Color</td>
<td>Tara Color</td>
<td></td>
<td></td>
</tr>
<tr>
<td>69</td>
<td>0</td>
<td>2.23E+06</td>
<td>1.35E+07</td>
<td>5.01E+07</td>
<td>1.05E+07</td>
<td>5.33E+05</td>
<td>1.10E+07</td>
<td>3.46E+06</td>
<td>gray</td>
<td>DNA</td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>0</td>
<td>2.44E+06</td>
<td>1.36E+07</td>
<td>4.91E+07</td>
<td>5.46E+06</td>
<td>4.33E+05</td>
<td>3.41E+06</td>
<td>8.61E+05</td>
<td>gray</td>
<td>DNA</td>
<td></td>
</tr>
<tr>
<td>71</td>
<td>0</td>
<td>4.28E+06</td>
<td>1.34E+07</td>
<td>8.13E+07</td>
<td>7.43E+06</td>
<td>3.47E+05</td>
<td>3.71E+06</td>
<td>7.24E+05</td>
<td>gray</td>
<td>DNA</td>
<td></td>
</tr>
<tr>
<td>72</td>
<td>0</td>
<td>4.48E+06</td>
<td>1.34E+07</td>
<td>7.53E+07</td>
<td>8.06E+06</td>
<td>3.79E+05</td>
<td>2.03E+06</td>
<td>7.24E+05</td>
<td>gray</td>
<td>DNA</td>
<td></td>
</tr>
<tr>
<td>73</td>
<td>0.00</td>
<td>1.70E+06</td>
<td>4.18E+05</td>
<td>1.86E+06</td>
<td>2.78E+06</td>
<td>1.27E+05</td>
<td>3.11E+06</td>
<td>1.15E+06</td>
<td>dark gray</td>
<td>DNA</td>
<td></td>
</tr>
<tr>
<td>74</td>
<td>0</td>
<td>2.41E+06</td>
<td>5.24E+05</td>
<td>1.21E+06</td>
<td>7.96E+06</td>
<td>2.08E+05</td>
<td>3.95E+06</td>
<td>1.19E+06</td>
<td>dark gray</td>
<td>DNA</td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>0.00</td>
<td>1.38E+06</td>
<td>2.81E+05</td>
<td>1.20E+06</td>
<td>1.86E+06</td>
<td>1.06E+05</td>
<td>2.06E+06</td>
<td>7.62E+05</td>
<td>gray</td>
<td>DNA</td>
<td></td>
</tr>
<tr>
<td>76</td>
<td>0.00</td>
<td>7.28E+05</td>
<td>1.47E+05</td>
<td>1.04E+06</td>
<td>1.36E+06</td>
<td>1.53E+05</td>
<td>3.54E+06</td>
<td>6.31E+05</td>
<td>gray</td>
<td>DNA</td>
<td></td>
</tr>
<tr>
<td>77</td>
<td>0.00</td>
<td>1.03E+06</td>
<td>2.65E+05</td>
<td>9.37E+05</td>
<td>2.13E+06</td>
<td>1.69E+05</td>
<td>3.50E+06</td>
<td>7.55E+05</td>
<td>gray</td>
<td>DNA</td>
<td></td>
</tr>
<tr>
<td>78</td>
<td>0.01</td>
<td>1.68E+06</td>
<td>2.43E+05</td>
<td>1.48E+06</td>
<td>5.03E+05</td>
<td>2.25E+05</td>
<td>8.39E+05</td>
<td>1.04E+06</td>
<td>gray</td>
<td>DNA</td>
<td></td>
</tr>
<tr>
<td>79</td>
<td>0</td>
<td>1.39E+06</td>
<td>2.29E+05</td>
<td>2.06E+06</td>
<td>3.48E+06</td>
<td>1.86E+05</td>
<td>3.61E+06</td>
<td>7.61E+05</td>
<td>gray</td>
<td>DNA</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>0</td>
<td>1.55E+06</td>
<td>2.15E+05</td>
<td>3.47E+06</td>
<td>1.43E+06</td>
<td>1.52E+05</td>
<td>4.19E+06</td>
<td>6.71E+05</td>
<td>gray</td>
<td>DNA</td>
<td></td>
</tr>
<tr>
<td>81</td>
<td>0.13</td>
<td>4.72E+06</td>
<td>7.80E+05</td>
<td>1.00E+07</td>
<td>8.71E+06</td>
<td>9.32E+05</td>
<td>1.16E+07</td>
<td>3.54E+06</td>
<td>gray-orange</td>
<td></td>
<td></td>
</tr>
<tr>
<td>82</td>
<td>0</td>
<td>3.40E+06</td>
<td>6.59E+05</td>
<td>1.97E+06</td>
<td>1.03E+07</td>
<td>9.11E+05</td>
<td>9.37E+06</td>
<td>1.21E+06</td>
<td>gray</td>
<td></td>
<td></td>
</tr>
<tr>
<td>83</td>
<td>0</td>
<td>2.49E+06</td>
<td>3.97E+05</td>
<td>3.09E+06</td>
<td>3.11E+06</td>
<td>2.03E+05</td>
<td>3.84E+06</td>
<td>1.24E+06</td>
<td>dark gray</td>
<td></td>
<td></td>
</tr>
<tr>
<td>84</td>
<td>0</td>
<td>1.59E+06</td>
<td>3.58E+05</td>
<td>6.01E+06</td>
<td>3.07E+06</td>
<td>2.06E+05</td>
<td>3.90E+06</td>
<td>1.04E+06</td>
<td>gray</td>
<td></td>
<td></td>
</tr>
<tr>
<td>85</td>
<td>0.71</td>
<td>1.91E+06</td>
<td>2.14E+05</td>
<td>4.92E+06</td>
<td>2.09E+06</td>
<td>2.94E+05</td>
<td>8.59E+06</td>
<td>2.05E+06</td>
<td>dark gray-black</td>
<td></td>
<td></td>
</tr>
<tr>
<td>86</td>
<td>0.00</td>
<td>1.46E+06</td>
<td>1.55E+05</td>
<td>2.81E+06</td>
<td>1.98E+06</td>
<td>1.99E+05</td>
<td>6.88E+06</td>
<td>1.10E+06</td>
<td>dark gray-black</td>
<td></td>
<td></td>
</tr>
<tr>
<td>87</td>
<td>0</td>
<td>2.58E+06</td>
<td>2.70E+05</td>
<td>1.07E+07</td>
<td>2.84E+06</td>
<td>2.18E+05</td>
<td>6.70E+06</td>
<td>1.51E+06</td>
<td>dark gray-black</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: The table seems to contain a mix of values and units that are not clearly defined, and the context is not clear from the data provided.
Processed Data
| Z | T | AA | AB | AC | AD | AE | AF | AG | AH | AI | AJ | AK | AL | AM |
|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| **6. (Calculated Raw Data in Absolute Numbers)** | | | | | | | | | | | | | | |
| Run Date | Temperature (°C) | Heating Time (sec) | Heating Rates (°C) | Hold Temp. (°C) | Hold Time (sec) | Wt. Loss (%) | Total Tars (mg) | Condensables (mg) | CO (µg) | CO2 (µg) | H2O (mg) | CH4 (µg) | C2H2 (µg) | C2H4 (µg) | C2H6 (µg) |
| 8/11/94 V | 354 | 0.50 | 1100 | 0 | 0 | 1.45 | 0.45 | 14.44 | 69.94 | 0.91 | 0.18 | 0.65 | 0.26 | 0.96 |
| 9/6/95 H | 615 | 0.40 | 978 | 0 | 0 | 1.39 | 0.48 | 7.38 | 61.28 | 0.12 | 0.75 | 0.34 | 0.64 | 0.65 |
| 10/1/95 V | 518 | 0.50 | 958 | 0 | 0 | 2.27 | 0.30 | 37.42 | 157.91 | 1.61 | 0.30 | 3.14 | 0.67 |
| 10/13/94 F | 453 | 0.60 | 863 | 0 | 0 | 2.91 | 0.03 | 21 | 293.54 | 0.06 | 2.28 | 0.43 | 2.10 | 0.52 |
| 11/6/95 H | 555 | 0.60 | 885 | 0 | 0 | 3.96 | 0.68 | 39.41 | 287.59 | 3.60 | 3.15 | 0.40 | 2.25 | 0.86 |
| 12/1/95 F | 587 | 0.60 | 930 | 0 | 0 | 4.88 | 0.17 | 41.43 | 287.57 | 2.54 | 3.33 | 0.41 | 2.68 | 1.03 |
| 12/4/95 H | 626 | 0.60 | 1003 | 0 | 0 | 5.11 | 0.73 | 21.81 | 297.91 | 0.13 | 4.72 | 0.48 | 4.09 | 1.49 |
| 12/15/95 H | 718 | 0.80 | 968 | 0 | 0 | 5.07 | 0.44 | 116.93 | 718.19 | 0.30 | 6.43 | 0.81 | 4.49 | 0.99 |
| 1/19/95 V | 747 | 0.75 | 964 | 0 | 0 | 4.87 | 0.87 | 177.25 | 698.69 | 0.08 | 6.43 | 0.81 | 4.49 | 0.99 |
| 1/21/95 V | 730 | 0.75 | 995 | 0 | 0 | 5.43 | 0.93 | 166.19 | 684.73 | 0.08 | 6.43 | 0.81 | 4.49 | 0.99 |
| 1/28/95 H | 805 | 0.80 | 930 | 0 | 0 | 5.60 | 0.63 | 244.62 | 574.32 | 0.97 | 6.43 | 0.81 | 4.49 | 0.99 |
| 1/17/96 H | 848 | 1.00 | 824 | 0 | 0 | 9.19 | 0.72 | 212.33 | 1041.84 | 0.31 | 14.87 | 1.58 | 7.37 | 0.97 |
| 2/2/96 V | 814 | 0.95 | 970 | 0 | 0 | 8.20 | 1.02 | 195.35 | 934.26 | 0.96 | 1.92 | 5.62 | 0.85 |
| 2/1/96 H | 931 | 1.00 | 907 | 0 | 0 | 12.40 | 0.44 | 402.38 | 2147.70 | 0.76 | 14.97 | 2.58 | 7.96 | 1.03 |
| 2/12/96 H | 934 | 1.05 | 867 | 0 | 0 | 9.45 | 1.02 | 290.57 | 1284.86 | 11.27 | 1.93 | 5.72 | 0.92 |
| 2/13/96 H | 951 | 1.00 | 931 | 0 | 0 | 12.92 | 0.79 | 573.18 | 3000.84 | 9.31 | 1.73 | 4.27 | 0.48 |
| 2/16/96 H | 962 | 1.00 | 938 | 0 | 0 | 11.28 | 0.95 | 162.32 | 1951.31 | 0 | 11.12 | 1.62 | 5.73 |
| 2/17/96 H | 972 | 1.00 | 948 | 0 | 0 | 12.80 | 1.19 | 199.10 | 2595.14 | 0 | 11.23 | 2.05 | 5.87 | 0.81 |
| 2/18/95 H | 1019 | 1.00 | 955 | 0 | 0 | 12.60 | 1.51 | 637.10 | 2486.12 | 0.14 | 14.49 | 2.73 | 9.02 | 1.68 |
| 2/20/95 H | 1053 | 1.25 | 807 | 987 | 5 | 21.42 | 1.94 | 1523.98 | 4955.97 | 42.67 | 8.16 | 25.6 | 0.7 |
| 2/27/95 H | 1053 | 1.25 | 807 | 1045 | 5 | 21.50 | 1.36 | 2085.33 | 5027.51 | 32.97 | 4.81 | 14.33 | 0.19 |

51. Total bars = tar filter + tar filter furnace + reactor tar samples + glass filters (reactorblings).
52. H2O numbers are corrected by a blank run for a baseline value of water (0.92 µg) obtained in trap after 2 flushes.
53. EPA Soil = 5% (wt.) Pyrene Data - Temperature Study (no hold at final temperature).
54. EPA Soil = 5% (wt.) Pyrene Data - Temperature Study (no hold at final temperature).
<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>62</td>
<td>44</td>
<td>4.68</td>
<td>50.84</td>
<td>2.64</td>
</tr>
<tr>
<td>63</td>
<td>50</td>
<td>4.85</td>
<td>50.97</td>
<td>2.64</td>
</tr>
<tr>
<td>64</td>
<td>55</td>
<td>4.99</td>
<td>51.11</td>
<td>2.62</td>
</tr>
<tr>
<td>65</td>
<td>60</td>
<td>5.12</td>
<td>51.22</td>
<td>2.12</td>
</tr>
<tr>
<td>66</td>
<td>65</td>
<td>5.25</td>
<td>51.38</td>
<td>-0.81</td>
</tr>
<tr>
<td>67</td>
<td>70</td>
<td>5.39</td>
<td>51.54</td>
<td>-1.18</td>
</tr>
<tr>
<td>68</td>
<td>75</td>
<td>5.53</td>
<td>51.70</td>
<td>-1.55</td>
</tr>
<tr>
<td>69</td>
<td>80</td>
<td>5.67</td>
<td>51.85</td>
<td>-1.92</td>
</tr>
<tr>
<td>70</td>
<td>85</td>
<td>5.81</td>
<td>52.01</td>
<td>-2.29</td>
</tr>
<tr>
<td>71</td>
<td>90</td>
<td>5.95</td>
<td>52.16</td>
<td>-2.66</td>
</tr>
<tr>
<td>72</td>
<td>95</td>
<td>6.09</td>
<td>52.32</td>
<td>-3.03</td>
</tr>
<tr>
<td>73</td>
<td>100</td>
<td>6.23</td>
<td>52.47</td>
<td>-3.40</td>
</tr>
<tr>
<td>74</td>
<td>105</td>
<td>6.37</td>
<td>52.63</td>
<td>-3.77</td>
</tr>
<tr>
<td>75</td>
<td>110</td>
<td>6.51</td>
<td>52.78</td>
<td>-4.14</td>
</tr>
<tr>
<td>76</td>
<td>115</td>
<td>6.65</td>
<td>52.93</td>
<td>-4.51</td>
</tr>
<tr>
<td>77</td>
<td>120</td>
<td>6.79</td>
<td>53.08</td>
<td>-4.88</td>
</tr>
<tr>
<td>78</td>
<td>125</td>
<td>6.93</td>
<td>53.23</td>
<td>-5.25</td>
</tr>
<tr>
<td>79</td>
<td>130</td>
<td>7.07</td>
<td>53.38</td>
<td>-5.62</td>
</tr>
<tr>
<td>80</td>
<td>135</td>
<td>7.21</td>
<td>53.53</td>
<td>-5.99</td>
</tr>
<tr>
<td>81</td>
<td>140</td>
<td>7.35</td>
<td>53.68</td>
<td>-6.36</td>
</tr>
<tr>
<td>82</td>
<td>145</td>
<td>7.49</td>
<td>53.83</td>
<td>-6.73</td>
</tr>
<tr>
<td>83</td>
<td>150</td>
<td>7.63</td>
<td>53.98</td>
<td>-7.10</td>
</tr>
<tr>
<td>84</td>
<td>155</td>
<td>7.77</td>
<td>54.13</td>
<td>-7.47</td>
</tr>
<tr>
<td>85</td>
<td>160</td>
<td>7.91</td>
<td>54.28</td>
<td>-7.84</td>
</tr>
<tr>
<td>86</td>
<td>165</td>
<td>8.05</td>
<td>54.43</td>
<td>-8.21</td>
</tr>
<tr>
<td>87</td>
<td>170</td>
<td>8.19</td>
<td>54.58</td>
<td>-8.58</td>
</tr>
<tr>
<td>88</td>
<td>175</td>
<td>8.33</td>
<td>54.73</td>
<td>-8.95</td>
</tr>
<tr>
<td>89</td>
<td>180</td>
<td>8.47</td>
<td>54.88</td>
<td>-9.32</td>
</tr>
<tr>
<td>90</td>
<td>185</td>
<td>8.61</td>
<td>55.03</td>
<td>-9.69</td>
</tr>
<tr>
<td>91</td>
<td>190</td>
<td>8.75</td>
<td>55.18</td>
<td>-10.06</td>
</tr>
<tr>
<td>92</td>
<td>195</td>
<td>8.89</td>
<td>55.33</td>
<td>-10.43</td>
</tr>
<tr>
<td>93</td>
<td>200</td>
<td>9.03</td>
<td>55.48</td>
<td>-10.80</td>
</tr>
<tr>
<td>94</td>
<td>205</td>
<td>9.17</td>
<td>55.63</td>
<td>-11.17</td>
</tr>
<tr>
<td>95</td>
<td>210</td>
<td>9.31</td>
<td>55.78</td>
<td>-11.54</td>
</tr>
<tr>
<td>96</td>
<td>215</td>
<td>9.45</td>
<td>55.93</td>
<td>-11.91</td>
</tr>
<tr>
<td>97</td>
<td>220</td>
<td>9.59</td>
<td>56.08</td>
<td>-12.28</td>
</tr>
<tr>
<td>98</td>
<td>225</td>
<td>9.73</td>
<td>56.23</td>
<td>-12.65</td>
</tr>
<tr>
<td>99</td>
<td>230</td>
<td>9.87</td>
<td>56.38</td>
<td>-13.02</td>
</tr>
<tr>
<td>100</td>
<td>235</td>
<td>10.01</td>
<td>56.53</td>
<td>-13.39</td>
</tr>
<tr>
<td>101</td>
<td>240</td>
<td>10.15</td>
<td>56.68</td>
<td>-13.76</td>
</tr>
<tr>
<td>102</td>
<td>245</td>
<td>10.29</td>
<td>56.83</td>
<td>-14.13</td>
</tr>
<tr>
<td>103</td>
<td>250</td>
<td>10.43</td>
<td>56.98</td>
<td>-14.50</td>
</tr>
<tr>
<td>104</td>
<td>255</td>
<td>10.57</td>
<td>57.13</td>
<td>-14.87</td>
</tr>
<tr>
<td>105</td>
<td>260</td>
<td>10.71</td>
<td>57.28</td>
<td>-15.24</td>
</tr>
<tr>
<td>106</td>
<td>265</td>
<td>10.85</td>
<td>57.43</td>
<td>-15.61</td>
</tr>
<tr>
<td>107</td>
<td>270</td>
<td>10.99</td>
<td>57.58</td>
<td>-15.98</td>
</tr>
<tr>
<td>108</td>
<td>275</td>
<td>11.13</td>
<td>57.73</td>
<td>-16.35</td>
</tr>
<tr>
<td>109</td>
<td>280</td>
<td>11.27</td>
<td>57.88</td>
<td>-16.72</td>
</tr>
<tr>
<td>110</td>
<td>285</td>
<td>11.41</td>
<td>58.03</td>
<td>-17.09</td>
</tr>
<tr>
<td>111</td>
<td>290</td>
<td>11.55</td>
<td>58.18</td>
<td>-17.46</td>
</tr>
<tr>
<td>112</td>
<td>295</td>
<td>11.69</td>
<td>58.33</td>
<td>-17.83</td>
</tr>
<tr>
<td>113</td>
<td>300</td>
<td>11.83</td>
<td>58.48</td>
<td>-18.20</td>
</tr>
</tbody>
</table>

Note: The table represents data with differences and calibration values.
Normalized Data
<table>
<thead>
<tr>
<th>Run Date</th>
<th>Bascom-Turner File No.</th>
<th>Bascom-Turner Recorder</th>
<th>Converted File Name</th>
<th>Converted from</th>
<th>Profile Description</th>
<th>Temperature (°C)</th>
<th>Heating Rate (°C/s)</th>
<th>Hold Temp. (°C)</th>
<th>Hold Time (sec)</th>
<th>Wt. Loss (%)</th>
<th>Mass loss (mg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6/15/90</td>
<td>D1 3 File 109</td>
<td>El 3 File 533</td>
<td>10</td>
<td>1554n</td>
<td>Mostly clean, slight noise</td>
<td>534</td>
<td>1100</td>
<td>0</td>
<td>0</td>
<td>1.45</td>
<td>0.72</td>
</tr>
<tr>
<td>6/15/90</td>
<td>D1 3 File 109</td>
<td>El 3 File 533</td>
<td>10</td>
<td>1415n</td>
<td>Very clean</td>
<td>415</td>
<td>978</td>
<td>0</td>
<td>0</td>
<td>1.39</td>
<td>0.69</td>
</tr>
<tr>
<td>6/15/90</td>
<td>D1 3 File 109</td>
<td>El 3 File 533</td>
<td>10</td>
<td>1515n</td>
<td>Very clean</td>
<td>518</td>
<td>988</td>
<td>0</td>
<td>0</td>
<td>2.27</td>
<td>1.14</td>
</tr>
<tr>
<td>6/15/90</td>
<td>D1 3 File 109</td>
<td>El 3 File 533</td>
<td>10</td>
<td>1542n</td>
<td>Mostly clean, with a few noisy spots</td>
<td>542</td>
<td>883</td>
<td>0</td>
<td>0</td>
<td>2.91</td>
<td>1.48</td>
</tr>
<tr>
<td>6/15/90</td>
<td>D1 3 File 109</td>
<td>El 3 File 533</td>
<td>10</td>
<td>1551n</td>
<td>Mostly clean, with a few noisy spots</td>
<td>551</td>
<td>883</td>
<td>0</td>
<td>0</td>
<td>3.96</td>
<td>1.96</td>
</tr>
<tr>
<td>6/15/90</td>
<td>D1 3 File 109</td>
<td>El 3 File 533</td>
<td>10</td>
<td>1562n</td>
<td>Mostly clean, with slight noise on heatup</td>
<td>562</td>
<td>950</td>
<td>0</td>
<td>0</td>
<td>4.88</td>
<td>2.44</td>
</tr>
<tr>
<td>6/15/90</td>
<td>D1 3 File 109</td>
<td>El 3 File 533</td>
<td>10</td>
<td>1573n</td>
<td>Mostly clean, with slight noise on heatup</td>
<td>573</td>
<td>1003</td>
<td>0</td>
<td>0</td>
<td>5.11</td>
<td>2.55</td>
</tr>
<tr>
<td>6/15/90</td>
<td>D1 3 File 109</td>
<td>El 3 File 533</td>
<td>10</td>
<td>1581n</td>
<td>Very clean</td>
<td>718</td>
<td>868</td>
<td>0</td>
<td>0</td>
<td>6.07</td>
<td>3.24</td>
</tr>
<tr>
<td>6/15/90</td>
<td>D1 3 File 109</td>
<td>El 3 File 533</td>
<td>10</td>
<td>1591n</td>
<td>Very clean</td>
<td>747</td>
<td>964</td>
<td>0</td>
<td>0</td>
<td>6.87</td>
<td>3.22</td>
</tr>
<tr>
<td>6/15/90</td>
<td>D1 3 File 109</td>
<td>El 3 File 533</td>
<td>10</td>
<td>1602n</td>
<td>Clean, with very slight noise</td>
<td>850</td>
<td>993</td>
<td>0</td>
<td>0</td>
<td>7.43</td>
<td>3.27</td>
</tr>
<tr>
<td>6/15/90</td>
<td>D1 3 File 109</td>
<td>El 3 File 533</td>
<td>10</td>
<td>1613n</td>
<td>Clean, with very slight noise</td>
<td>805</td>
<td>930</td>
<td>0</td>
<td>0</td>
<td>8.60</td>
<td>3.80</td>
</tr>
<tr>
<td>6/15/90</td>
<td>D1 3 File 109</td>
<td>El 3 File 533</td>
<td>10</td>
<td>1624n</td>
<td>Very clean</td>
<td>848</td>
<td>924</td>
<td>0</td>
<td>0</td>
<td>9.19</td>
<td>4.60</td>
</tr>
<tr>
<td>6/15/90</td>
<td>D1 3 File 109</td>
<td>El 3 File 533</td>
<td>10</td>
<td>1635n</td>
<td>Very clean</td>
<td>898</td>
<td>920</td>
<td>0</td>
<td>0</td>
<td>9.80</td>
<td>4.10</td>
</tr>
<tr>
<td>6/15/90</td>
<td>D1 3 File 109</td>
<td>El 3 File 533</td>
<td>10</td>
<td>1931n</td>
<td>Very clean</td>
<td>931</td>
<td>907</td>
<td>0</td>
<td>0</td>
<td>12.40</td>
<td>6.20</td>
</tr>
<tr>
<td>6/15/90</td>
<td>D1 3 File 109</td>
<td>El 3 File 533</td>
<td>10</td>
<td>1934n</td>
<td>Clean, with very slight noise</td>
<td>954</td>
<td>867</td>
<td>0</td>
<td>0</td>
<td>9.45</td>
<td>4.73</td>
</tr>
<tr>
<td>6/15/90</td>
<td>D1 3 File 109</td>
<td>El 3 File 533</td>
<td>10</td>
<td>1951n</td>
<td>Very clean, with only one noisy spot</td>
<td>950</td>
<td>987</td>
<td>0</td>
<td>0</td>
<td>12.92</td>
<td>6.49</td>
</tr>
<tr>
<td>6/15/90</td>
<td>D1 3 File 109</td>
<td>El 3 File 533</td>
<td>10</td>
<td>1972n</td>
<td>Mostly clean, with slight noise on heatup</td>
<td>972</td>
<td>948</td>
<td>0</td>
<td>0</td>
<td>12.90</td>
<td>6.40</td>
</tr>
<tr>
<td>6/15/90</td>
<td>D1 3 File 109</td>
<td>El 3 File 533</td>
<td>10</td>
<td>1983n</td>
<td>Mostly clean, with slight noise on heatup</td>
<td>985</td>
<td>958</td>
<td>0</td>
<td>0</td>
<td>12.90</td>
<td>6.30</td>
</tr>
<tr>
<td>6/15/90</td>
<td>D1 3 File 109</td>
<td>El 3 File 533</td>
<td>10</td>
<td>1994n</td>
<td>Mostly clean, with slight noise on heatup</td>
<td>986</td>
<td>968</td>
<td>0</td>
<td>0</td>
<td>13.10</td>
<td>6.40</td>
</tr>
<tr>
<td>6/15/90</td>
<td>D1 3 File 109</td>
<td>El 3 File 533</td>
<td>10</td>
<td>2005n</td>
<td>Mostly clean, with slight noise on heatup</td>
<td>995</td>
<td>978</td>
<td>0</td>
<td>0</td>
<td>13.10</td>
<td>6.40</td>
</tr>
<tr>
<td>6/15/90</td>
<td>D1 3 File 109</td>
<td>El 3 File 533</td>
<td>10</td>
<td>2016n</td>
<td>Mostly clean, with slight noise on heatup</td>
<td>996</td>
<td>988</td>
<td>0</td>
<td>0</td>
<td>13.10</td>
<td>6.40</td>
</tr>
<tr>
<td>6/15/90</td>
<td>D1 3 File 109</td>
<td>El 3 File 533</td>
<td>10</td>
<td>2027n</td>
<td>Mostly clean, with slight noise on heatup</td>
<td>997</td>
<td>998</td>
<td>0</td>
<td>0</td>
<td>13.10</td>
<td>6.40</td>
</tr>
<tr>
<td>6/15/90</td>
<td>D1 3 File 109</td>
<td>El 3 File 533</td>
<td>10</td>
<td>2038n</td>
<td>Mostly clean, with slight noise on heatup</td>
<td>998</td>
<td>999</td>
<td>0</td>
<td>0</td>
<td>13.10</td>
<td>6.40</td>
</tr>
</tbody>
</table>

Note: * Converted files all have 10 min time intervals
* signifies use of a FORTRAN code to interpolate between 100 ms pts to obtain 10 ms intervals

EPA Soil + 9% (w/w) Pyrene Data - Temperature Study (as held at Room Temperature)
<table>
<thead>
<tr>
<th></th>
<th>W</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>AA</th>
<th>AB</th>
<th>AC</th>
<th>AD</th>
<th>AE</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Total Tar (mg)</td>
<td>Condensates (mg)</td>
<td>CO (mg)</td>
<td>CO2 (mg)</td>
<td>H2O (mg)</td>
<td>CH4 (mg)</td>
<td>C2H2 (mg)</td>
<td>C2H4 (mg)</td>
<td>C2H6 (mg)</td>
</tr>
<tr>
<td>8</td>
<td>0.84</td>
<td>14.34</td>
<td>69.34</td>
<td>0.90</td>
<td>0.18</td>
<td>0.62</td>
<td>0.95</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.48</td>
<td>0.00</td>
<td>13.17</td>
<td>121.80</td>
<td>0.06</td>
<td>2.11</td>
<td>1.94</td>
<td>0.48</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1.01</td>
<td>0.51</td>
<td>0.51</td>
<td>9.09</td>
<td>56.72</td>
<td>1.07</td>
<td>0.06</td>
<td>0.80</td>
<td>0.59</td>
</tr>
<tr>
<td>11</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>12</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>13</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>14</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>15</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>16</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>17</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>18</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>19</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>20</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>21</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>22</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>23</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>24</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>25</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>26</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>27</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>28</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>29</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
<td>G</td>
<td>H</td>
<td>I</td>
<td>J</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>61</td>
<td>Blank EPA Soil Data - Heating Rate Study (no held at final temperature)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>62</td>
<td>Experimental data calculated on a 50 mg EPA soil sample base</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>63</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>65</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>66</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>67</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>68</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>69</td>
<td>Run Date</td>
<td>Bascom-Turner Recorder File No.</td>
<td>Sampling Interval (msec)</td>
<td>Converted File Name</td>
<td>Converted from point no.</td>
<td>Profile Description</td>
<td>Temperature (°C)</td>
<td>Heating Rate (°C/s)</td>
<td>Hold Temp. (°C)</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>68</td>
<td>6/3/93</td>
<td>Disc 3 File 129</td>
<td>10</td>
<td>N545m/1545m</td>
<td>317</td>
<td>Mostly clean, with a few noisy spikes</td>
<td>543</td>
<td>863</td>
<td>0</td>
</tr>
<tr>
<td>69</td>
<td>7/22/93</td>
<td>Disc 3 File 348</td>
<td>10</td>
<td>N555m/1555m</td>
<td>199</td>
<td>Very clean</td>
<td>555</td>
<td>845</td>
<td>0</td>
</tr>
<tr>
<td>70</td>
<td>7/22/93</td>
<td>Disc 3 File 350</td>
<td>10</td>
<td>N582m/1582m</td>
<td>237</td>
<td>Mostly clean, with slight noise on heated</td>
<td>582</td>
<td>930</td>
<td>0</td>
</tr>
<tr>
<td>71</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>72</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>73</td>
<td>5/13/94</td>
<td>Disc 3 File 168</td>
<td>100</td>
<td>N556m/556m</td>
<td>77</td>
<td>Very clean, but for spikes at peak temp</td>
<td>556</td>
<td>89</td>
<td>0</td>
</tr>
<tr>
<td>74</td>
<td>4/26/94</td>
<td>Disc 3 File 472</td>
<td>100</td>
<td>N507m/507m</td>
<td>58</td>
<td>Very clean</td>
<td>507</td>
<td>84</td>
<td>0</td>
</tr>
<tr>
<td>75</td>
<td>5/3/94</td>
<td>Disc 3 File 173</td>
<td>100</td>
<td>N507m/507m</td>
<td>41</td>
<td>Very clean</td>
<td>507</td>
<td>81</td>
<td>0</td>
</tr>
<tr>
<td>76</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>77</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>78</td>
<td>4/21/94</td>
<td>Disc 3 File 468</td>
<td>10</td>
<td>N521m</td>
<td>177</td>
<td>Very clean but for 1 small spike</td>
<td>521</td>
<td>522</td>
<td>0</td>
</tr>
<tr>
<td>79</td>
<td>4/26/94</td>
<td>Disc 3 File 471</td>
<td>10</td>
<td>N518m</td>
<td>151</td>
<td>Very clean but for 1 tiny spike</td>
<td>518</td>
<td>449</td>
<td>0</td>
</tr>
<tr>
<td>80</td>
<td>5/14/94</td>
<td>Disc 3 File 273</td>
<td>10</td>
<td>N518m/518m</td>
<td>XXXXX</td>
<td>Highly disrupted profile; replace</td>
<td>518</td>
<td>460</td>
<td>0</td>
</tr>
<tr>
<td>81</td>
<td>5/14/94</td>
<td>Disc 3 File 274</td>
<td>10</td>
<td>N518m/518m</td>
<td>XXXXX</td>
<td>Highly disrupted profile; replace</td>
<td>518</td>
<td>540</td>
<td>0</td>
</tr>
<tr>
<td>82</td>
<td>5/14/94</td>
<td>Disc 3 File 275</td>
<td>10</td>
<td>N518m/518m</td>
<td>XXXXX</td>
<td>Highly disrupted profile; replace</td>
<td>518</td>
<td>485</td>
<td>0</td>
</tr>
<tr>
<td>83</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>84</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>85</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>86</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>87</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>88</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>89</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>90</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>91</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>92</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>93</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>94</td>
<td>Run Date</td>
<td>Bascom-Turner Recorder File No.</td>
<td>Sampling Interval (msec)</td>
<td>Converted File Name</td>
<td>Converted from point no.</td>
<td>Profile Description</td>
<td>Temperature (°C)</td>
<td>Heating Rate (°C/s)</td>
<td>Hold Temp. (°C)</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>94</td>
<td>3/1/94</td>
<td>Disc 3 File 465</td>
<td>10</td>
<td>Very clean</td>
<td>513</td>
<td>83</td>
<td>0</td>
<td>0</td>
<td>8.88</td>
</tr>
<tr>
<td>95</td>
<td>3/10/94</td>
<td>Disc 3 File 466</td>
<td>10</td>
<td>Very clean</td>
<td>513</td>
<td>85</td>
<td>0</td>
<td>0</td>
<td>7.48</td>
</tr>
<tr>
<td>96</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>97</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>98</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>99</td>
<td>5/20/94</td>
<td>Disc 3 File 368</td>
<td>100</td>
<td>Very clean</td>
<td>548</td>
<td>87</td>
<td>0</td>
<td>0</td>
<td>7.42</td>
</tr>
<tr>
<td>100</td>
<td>5/10/94</td>
<td>Disc 3 File 276</td>
<td>100</td>
<td>Very clean, with slight noise spikes</td>
<td>570</td>
<td>91</td>
<td>0</td>
<td>0</td>
<td>7.49</td>
</tr>
<tr>
<td>101</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>102</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>103</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>104</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Converted file all time intervals

* signifies use of a FORTRAN program to interpolate between 100 ms points to obtain 10 ms intervals

* signifies use of a FORTRAN program to scale the HVH profile to the correct peak temperature to replace the experimentally-obtained profile

* signifies that the heated profile produced by FORTRAN program was manually interpolated to contain the correct heated time

EPA Soil + 550 (wt.) Pyrene Data - Heating Rate Study (no held at final temperature)

91 Experimental data calculated on a 50 mg EPA soil + 2.49 mg pyrene sample base

| Basis | 37.49 | Basis |
Statistical Analysis Methodology

Since the data in the temperature and heating rate studies were obtained using slightly different approaches, the methodology employed for the statistical analysis of each data set will be discussed separately. Each description will include the criteria selected for grouping data, issues which must be taken into account to fairly evaluate the data, and details for some of the more complicated cases.

Heating Rate Study

The data for the heating rate study was collected at three distinct heating rates simplifying the grouping of the data. However since obtaining exact peak temperatures was difficult, an important consideration was to ensure the average temperatures represented from each group is as close to the target 550°C as possible without affecting the trends observed. In cases to the contrary, data was selected or discarded to ensure that the average temperature trend across the heating rates would not overshadow the trends observed.

Hence, for the case of neat soil at ~1000°C/s heating rate, the run with a peak temperature of 626°C was removed as results from this run gave the highest values for weight loss and many light gases of the group. In the neat soil runs at ~5000°C/s heating rate, the 506°C peak temperature run could have been removed to raise the average temperature of the grouping from 537°C to 549°C. But the data points were consistently in the middle of the data points in this grouping leaving the trends unaffected, leading to it’s inclusion in the data set.

In the contaminated soil runs, the average temperatures for each data group increases with heating rate. More data points to change this trend were not obtained due to time constraints and difficulty at the time of experimentation to obtain temperatures precisely despite many efforts to do so. Since higher heating rates means less time at elevated temperatures, it was believed prior to experimenting that the quantities volatilized should decrease with heating rate. But the constant and decreasing trends in the volatiles observed with increasing heating rate can arguably be supported more strongly by our experimental trend of increasing peak temperature with increasing heating rate. But for cases where volatiles increase with heating rate from 100°C/s (avg. peak temp. = 531°C) to 1000°C/s (avg. peak temp. = 558°C), we could expect that the quantities volatilized may be slightly low at the lower heating rate. However whether this would be true is unclear since the temperature difference is relatively small and since volatiles collected did not systematically increase with temperature in the two contaminated soil runs at 100°C/s. For these reasons, it is believed that the data can be used for our study.

With regard to discordant data, there was only one case where a Q-test was used to reject one tar data point in a neat soil run. A Q-test consists of calculating the ratio of the difference between questionable value and its nearest agreeable value vs. the range of data values obtained. If the Q-value is greater than the critical value, the data point can be rejected. The tar point of 1.14 mg for neat soil at 518°C at 4491°C/s was rejected on this basis (90% confidence limit).

From an Experimental Physical Chemistry text I used as an undergraduate ("Experiments in Physical Chemistry", 4th ed. by Shoemaker, Garland, Steinfeld, and Nibler), it was mentioned that data is increasingly being cited with 95% confidence limits vs. the older practice of using standard deviations. However due to the small number of
data points, the 95% confidence limits in some cases makes the contaminated soil data indistinguishable from the neat soil data. Hence, it was believed that the standard deviation is the more fair measure of our data reproducibility. The standard deviation of the mean was used as an estimate of the standard deviation for each point in the group.

In summary, standard deviations of data points were calculated based on data grouped by heating rate while considering the effects of average group peak temperatures, and any discordant data was rejected using the Q-test.

Temperature Study

Contaminated soil had four target temperatures of 400°C, 550°C, 750°C, and 950°C, while neat EPA soil data points were gathered with more target temperatures in mind. Since the experimental strategies differ slightly, the statistical analysis for contaminated soil and neat soil will be discussed separately.

As not all data points were duplicated, error bars were assigned to points only where sufficient data was available.

Contaminated Soil

As the target temperatures were approximately 400, 550, 750, and 950°C, the data points for contaminated soil in the temperature study were grouped according to the target temperature, except for two points at 877 and 981°C. The point with a peak temperature of 877°C can be removed from the ~950°C peak temperature grouping (2 points) since the variac settings were set incorrectly low. The 981°C peak temperature run was set apart from the grouping since it appeared that the data point was distinctly different from the other two runs, giving the maximum values observed for contaminated soil in all volatiles collected except C2H6. Also, the 800 - 1000°C temperature zone appears to be a zone of rapid change calling for tighter grouping of data points by temperature.

Like in the heating rate study due to the small number of duplicates, the error bars represent standard deviations calculated for the mean values for each data point group.

Neat EPA Soil

Under Dr. Peter’s recommendation, my data was combined with Verónica’s data to provide enough data to compute error bars. There were many target temperatures for these neat soil runs using various variac/timer settings. But sometimes peak temperatures obtained did not increase with increasing variac/timer settings, making grouping of data more difficult. Thus using variac setting information, peak temperature, and gradients in the data from a freehand drawn trendline through the data, the majority of the data (13 of 19 points) were placed into four groupings of with peak temperatures of approximately 550°C, 750°C, 930°C, and 970°C, where standard deviations (reproducibility) were computed for each group. The reasoning used to determine each grouping is as follows:

- 550°C peak temperature runs: my four data points (542, 555, 582, 626°C) were combined with Veronica’s data point at 518°C since my data points all used the same variac/timer settings and Verónica’s data point had a similar temperature for a slightly lower timer setting. It was believed that since the volatiles is not increasing greatly in the 500-600°C temperature range, the grouping is not leading to unfairly large standard deviations.
- 750°C peak temperature runs: this is a case where my run at 718°C used higher variac/timer settings than Verónica's points at 747 and 750°C. Hence, temperature was the criteria for the grouping of these three points. The data point at 805°C was not included since it appeared in the light gases plots that that point is distinct (again, we are entering a high gradient zone where volatiles are increasing rapidly with a small increase in temperature - based on a freehand drawn trendline).

- 930°C peak temperature runs: The variac settings are the same for these runs as for the 970°C peak temperature runs. However peak temperatures can deviate for reasons such as inconsistent foil cutting (I did a calculation using power inputs where I found that a 5% error in cutting a foil could potentially lead to the deviations in temperature observed) and changing variac settings over time (Verónica and I both observed we had to increase variac settings over the last year to obtain the same temperature for reasons unknown). As the volatiles amounts appear to be rapidly increasing with temperature in the >--800°C temperature range, smaller temperature ranges were thought to be necessary to not be too harsh on the data. Primarily based on the CO2 data, it was felt that the 930 and 970°C runs had to be separated since they appeared distinct from each other.

- 970°C peak temperature runs: These points were the remaining high temperature runs not included in the 930°C grouping (except the 1019°C run which yields the maximum values of volatiles for neat soil runs).

I think it is also important to note that there were other potential groupings which could have been made based on temperature but were not, such as the 805 and 848°C neat EPA soil runs. The variac/timer settings used to obtain these points are different. These were not grouped since a freehand drawn trend line indicates the points are in a region of rapid change over a small temperature range (again where the data points are likely to be distinct from each other, hence being unfairly harsh on the data to group them for a statistical analysis).

As it was difficult to make these groupings based on objective criteria, any input comments and suggestions you have concerning this analysis would be greatly appreciated.

Hiroshi Saito
7/28/94
Appendix C

Debugged Computer Codes
for Soil Modeling

Contents of this section

1. DATAPREP2.F  Converts the data file stored by the Bascom-Turner stripchart recorder (rows of ten in groups of fifty points) into a two-column input file for the MIPR (multiple independent parallel reaction) and CRM (competitive reaction model) models. The two columns contain time (sec) and temperature (Celsius) data respectively.

2. Sample input file for the models converted from the Bascom-Turner file. The file was printed to show 4 pages of input file per page to save space.

3. STORE.F  FORTRAN program used to input MIPR (multiple independent parallel reaction) model parameters such as upper and lower bounds for variables, variable names, names of files containing the temperature-time profiles, and values to be fit by the model.

4. MIPR.F  The FORTRAN code for the multiple independent parallel reaction model written by Dr. Verónica Bucalá. The fitting technique is not a Levenberg-Marquardt method as mentioned in the code, but a bounded gradient-based technique. After reading the input files created by running STORE.F, the program will ask for initial guesses before searching for the optimum parameter values.

5. HSTORE2.F  H. Saito’s modification of STORE.F to be used for the CRM (competitive reaction models).

6. CRM2.F  The FORTRAN code for the CRM-P model (competitive reaction model - parallel reactions) written by H. Saito. The fitting technique is a true Levenberg-Marquardt method. After reading the input files created by HSTORE2.F, the program will ask for initial guesses before searching for optimum parameter values. To solve the differential equations for the reaction set, subroutine LSODE (Livermore Solver for ODE’s) was employed (also included in the program listing).

7. CRM3.F  A partial listing of the FORTRAN code for the CRM-S model (competitive reaction model - series reactions) written by H. Saito. Since the program is virtually identical to CRM2.F with the exception of two subroutines (related to solving the differential equations), only the two subroutines affected are listed here.
PROGRAM DATAPREP

This program converts the Bascom-Turner Strip Chart Recorder data file to a two column data file containing time (sec) and temperature (Celsius).

This program converts the numbers (0 - 10, full scale being 50 mV) from the data file to Celsius, requesting input from the user as to which point to start at 0 sec, and proceeds to insert increasing time counts before each temperature reading.

Defining variables
- L,J,K = counter variables (1 for datafile rows, J for counting points, K for reading/printing pts)
- IMAX = number of rows the Bascom-Turner data file has
- JUNK = real number variable to read spaces in Bascom-Turner data file
- NUMPT = number of data points in the file
- BTNUM = numbers read in from Bascom-Turner data file
- TEMP = temperatures (Celsius) converted from BTNUM
- TIME = time (sec) stored in array
- UV = microvolts calculated from BTNUM
- TINT = time interval (msec) between pts. from Bascom-Turner
- TSTART = data point number to start counting up from 0 sec
- FILEIN = input file name (transferred Bascom-Turner data)
- FILEOUT = output file name (containing time and temp. info)

PROGRAM PREPDATA
INTEGER I, J, IMAX, NUMPT, TSTART
REAL BTNUM(2050), TEMP(2050), TIME(2050), UV, TINT
CHARACTER*12 FILEIN, FILEOUT
CHARACTER*1 JUNK

Initializing variables

L = 1
IMAX = 240
NUMPT = 2000
TINT = 0.0
TSTART = 0.0
UV = 0.0
JUNK = ' '  
do 100 I = 1, 2050, 1
   BTNUM(I) = 0.0
   TEMP(I) = 0.0
   TIME(I) = 0.0
100 CONTINUE

File, which data point from which to start counting time, time interval between each data point, and input and output file names

PRINT *, 'Input file name? (containing Bascom-Turner data)
PRINT *, 'd. #___ (in single quotes, up to 12 characters)
READ *, FILEIN
PRINT *, 'Input file name? (to contain time & temp columns)
PRINT *, '__________ (in single quotes, up to 12 characters)
READ *, FILEOUT
PRINT *, 'Number of data points in the file? [Integer]
READ *, NUMPT

Setting IMAX to account for the spaces between every group of 50 data points in the Bascom-Turner data file.

IMAX = NUMPT/10 + INT(NUMPT/50)

PRINT *, 'Start counting time from point number ___? [Integer]
READ *, TSTART
PRINT *, 'Bascom-Turner Sampling Time Interval (ms)? [Real]
READ *, TINT

Opening data file and reading in the values by rows of 10 pts.

PRINT *, ' ', PRINT *, ' ', PRINT *, ' ', PRINT *, ' ', PRINT *, ' ', PRINT *, ' ', OPEN(1, FILE=FILEIN, STATUS='OLD')
I = 0
J = 1
DO 110 J = 1, IMAX
   IF (J LE NUMPT) THEN
      READ (1, *) (BTNUM(J+K), K = 0, 9)
      PRINT *, (BTNUM(J+K), K = 0, 9)
      J = J+10
   ENDIF
110 CONTINUE
CLOSE(1)

Convert Bascom-Turner numbers (BTNUM) to temperature (TEMP) in the units of Celsius for all data points in the file

microvolts = (BTNUM/10)*50 mV * 1000 uV/mV
temperature (C) obtained from quantic equations
in 'Thermocouple Reference Tables based on the IPTS-68', National Bureau of Standards
Monograph 125, March 1974 (Library of Congress
Catalog Number 73-600214)

T(C) = A0 + A1*UV + A2*UV^2 + A3*UV^3 + A4*UV^4

A0 = 6.2300671
A1  2.438324E-2  2.4955374E-2
A2  9.7830251E-9  -7.8783333E-8
A3  3.6276965E-12  1.3269743E-12
A4  -2.5756438E-16  1.5580541E-18
ERROR RANGE (C)  -0.3 TO 0.6  -0.3 TO 0.3

PRINT *,
PRINT *,
PRINT *, 'Converting Bascom-Turner Data to microvolts'
PRINT *, 'and then to temperature (Celsius)'
PRINT *,
PRINT *,
DO 120 J = 1, NUMPT
 UV = (BNUM/JV10)**50*1000
* Checks if microvoltag greater than that corresponding to 400 C,
* the cutoff between the two regressions.
IF (UV < 8314) THEN
 TEMP(I) = (2.4383248E-2*UV) + (9.7830251E-9*(UV**2))
   + (3.6276965E-12*(UV**3)) - (2.5756438E-16*(UV**4))
 ELSE
 TEMP(I) = 6.2300671 + (2.4955374E-8*UV)
   - (7.8783333E-8*(UV**2) + (1.3269743E-12*(UV**3))
   + (1.5580541E-18*(UV**4))
 ENDIF
120 CONTINUE

* Starts timer at point number TSTART to track the reaction time
* PRINT *,
PRINT *,
PRINT *, 'Locating and inserting time counter'
PRINT *,
PRINT *
TIME(TSTART-1) = -(TINT/1000.0)
DO 130, I = TSTART, NUMPT
 TIME(I) = TIME(I-1) + (TINT/1000.0)
 PRINT *, TIME(I)
130 CONTINUE

STORE data as time and temperature columns in the output file
PRINT *,
PRINT *,
PRINT *, 'Storing Time-Temperature data in file', FILEOUT
PRINT *
PRINT *
OPEN(I, FILE=FILEOUT, STATUS='NEW')
DO 140, J = TSTART, NUMPT
 FORMAT (1X, F6.2, 3X, F7.1)
 WRITE (I, 135) TIME(I), TEMP(I)
140 CONTINUE
CLOSE(I)
STOP
END
Sample input file for the MI-FR and CRM programs containing the temperature-time profiles.
| 6.48 | 183.4 | 7.02 | 168.9 | 7.56 | 155.7 | 8.10 | 143.8 |
| 6.49 | 183.1 | 7.03 | 168.7 | 7.57 | 155.5 | 8.11 | 143.6 |
| 6.50 | 182.8 | 7.04 | 168.3 | 7.58 | 155.5 | 8.12 | 143.6 |
| 6.51 | 182.5 | 7.05 | 168.1 | 7.59 | 155.1 | 8.13 | 143.2 |
| 6.52 | 182.1 | 7.06 | 167.7 | 7.60 | 154.9 | 8.14 | 143.2 |
| 6.53 | 181.9 | 7.07 | 167.7 | 7.61 | 154.5 | 8.15 | 142.9 |
| 6.54 | 181.5 | 7.08 | 167.4 | 7.62 | 154.5 | 8.16 | 142.5 |
| 6.55 | 181.3 | 7.09 | 167.1 | 7.63 | 154.2 | 8.17 | 142.3 |
| 6.56 | 180.9 | 7.10 | 166.8 | 7.64 | 153.9 | 8.18 | 142.3 |
| 6.57 | 180.6 | 7.11 | 166.4 | 7.65 | 153.6 | 8.19 | 141.9 |
| 6.58 | 180.5 | 7.12 | 166.4 | 7.66 | 153.6 | 8.20 | 141.7 |
| 6.59 | 180.3 | 7.13 | 166.2 | 7.67 | 153.2 | 8.21 | 141.7 |
| 6.60 | 180.0 | 7.14 | 165.8 | 7.68 | 153.0 | 8.22 | 141.3 |
| 6.61 | 179.6 | 7.15 | 165.5 | 7.69 | 153.0 | 8.23 | 141.3 |
| 6.62 | 179.4 | 7.16 | 165.5 | 7.70 | 152.6 | 8.24 | 141.0 |
| 6.63 | 179.0 | 7.17 | 164.9 | 7.71 | 152.3 | 8.25 | 140.7 |
| 6.64 | 178.7 | 7.18 | 164.9 | 7.72 | 152.3 | 8.26 | 140.7 |
| 6.65 | 178.4 | 7.19 | 164.5 | 7.73 | 152.0 | 8.27 | 140.4 |
| 6.66 | 178.4 | 7.20 | 164.3 | 7.74 | 151.7 | 8.28 | 140.0 |
| 6.67 | 177.7 | 7.21 | 164.3 | 7.75 | 151.3 | 8.29 | 140.0 |
| 6.68 | 177.7 | 7.22 | 163.9 | 7.76 | 151.3 | 8.30 | 139.8 |
| 6.69 | 177.5 | 7.23 | 163.7 | 7.77 | 151.1 | 8.31 | 139.4 |
| 6.70 | 177.1 | 7.24 | 163.1 | 7.78 | 151.1 | 8.32 | 139.4 |
| 6.71 | 176.9 | 7.25 | 163.1 | 7.79 | 150.7 | 8.33 | 139.2 |
| 6.72 | 176.9 | 7.26 | 162.7 | 7.80 | 150.4 | 8.34 | 138.8 |
| 6.73 | 176.5 | 7.27 | 162.7 | 7.81 | 150.1 | 8.35 | 138.5 |
| 6.74 | 176.2 | 7.28 | 162.4 | 7.82 | 149.8 | 8.36 | 138.5 |
| 6.75 | 175.9 | 7.29 | 162.0 | 7.83 | 149.8 | 8.37 | 138.2 |
| 6.76 | 175.6 | 7.30 | 161.8 | 7.84 | 149.5 | 8.38 | 138.2 |
| 6.77 | 175.2 | 7.31 | 161.8 | 7.85 | 149.5 | 8.39 | 137.9 |
| 6.78 | 175.0 | 7.32 | 161.4 | 7.86 | 149.2 | 8.40 | 137.5 |
| 6.79 | 174.6 | 7.33 | 161.1 | 7.87 | 148.8 | 8.41 | 137.5 |
| 6.80 | 174.6 | 7.34 | 161.1 | 7.88 | 148.6 | 8.42 | 137.3 |
| 6.81 | 174.3 | 7.35 | 160.8 | 7.89 | 148.6 | 8.3 | 136.9 |
| 6.82 | 174.0 | 7.36 | 160.5 | 7.90 | 148.2 | 8.44 | 136.9 |
| 6.83 | 173.7 | 7.37 | 160.1 | 7.91 | 147.9 | 8.45 | 136.6 |
| 6.84 | 173.7 | 7.38 | 159.9 | 7.92 | 147.6 | 8.46 | 136.3 |
| 6.85 | 173.3 | 7.39 | 159.9 | 7.93 | 147.6 | 8.47 | 136.3 |
| 6.86 | 173.1 | 7.40 | 159.5 | 7.94 | 147.3 | 8.48 | 136.0 |
| 6.87 | 172.7 | 7.41 | 159.1 | 7.95 | 146.9 | 8.49 | 136.0 |
| 6.88 | 172.5 | 7.42 | 159.3 | 7.96 | 146.9 | 8.50 | 135.6 |
| 6.89 | 172.1 | 7.43 | 158.9 | 7.97 | 146.7 | 8.51 | 135.4 |
| 6.90 | 171.8 | 7.44 | 158.6 | 7.98 | 146.7 | 8.52 | 135.4 |
| 6.91 | 171.8 | 7.45 | 158.3 | 7.99 | 146.3 | 8.53 | 135.0 |
| 6.92 | 171.5 | 7.46 | 158.3 | 8.00 | 146.1 | 8.54 | 134.8 |
| 6.93 | 171.2 | 7.47 | 157.6 | 8.01 | 146.1 | 8.55 | 134.8 |
| 6.94 | 170.8 | 7.48 | 157.6 | 8.02 | 145.7 | 8.56 | 134.4 |
| 6.95 | 170.6 | 7.49 | 157.6 | 8.03 | 145.4 | 8.57 | 134.4 |
| 6.96 | 170.6 | 7.50 | 157.0 | 8.04 | 145.1 | 8.58 | 134.1 |
| 6.97 | 169.9 | 7.51 | 157.0 | 8.05 | 145.1 | 8.59 | 133.8 |
| 6.98 | 169.9 | 7.52 | 156.7 | 8.06 | 144.8 | 8.60 | 133.8 |
| 6.99 | 169.6 | 7.53 | 156.7 | 8.07 | 144.4 | 8.61 | 133.5 |
| 7.00 | 169.5 | 7.54 | 156.4 | 8.08 | 144.2 | 8.62 | 133.1 |
| 7.01 | 168.9 | 7.55 | 156.1 | 8.09 | 144.2 | 8.63 | 133.1 |
STORE.F
PROGRAM TO STORE THE DATA TO BE ADJUSTED BY LEVENBERG-MARQUARDT METHOD

IMPLICIT REAL*(A-H,O-Z)
DOUBLE PRECISION NOMBRP(12)
DIMENSION YE(100),BMN(12),BMAX(12)
CHARACTER*10 C(30)

WRITE(5,1000)
1000 FORMAT(5X,**INPUT OF DATA TO BE ADJUSTED BY SINGLE REACTION MODEL**)
     **J**/**5X,**J**/*10**)
WRITE(5,10002)
1002 FORMAT(5X,**Number of experiments**)
READ(5,1003)ND
WRITE(5,1009)
1009 FORMAT(5X,**Number of experiments**)
READ(5,1000)NEXP
WRITE(5,1005)
1005 FORMAT(5X,**Number of independent variables: in our case=1**)
     **J**/**5X,**J**/**10**)
READ(5,1003)NV
WRITE(5,1005)
1005 FORMAT(5X,**Number of parameters to fit: **)
READ(5,1003)NI
OPEN(UNIT=1,FILE='NOLIN1.DAT',STATUS='NEW',FORM='FORMATTED')
WRITE(1,2000)NEXP,ND,NI,NV
2000 FORMAT(315)
CLOSE(UNIT=1)

WRITE(5,10006)
1006 FORMAT(5X,**File names containing the T-1 profile**)
     **J**/**5X,**J**/**10**)
DO 1010=1,NEXP
     WRITE(5,1007)
1007 FORMAT(5X,**File name containing the T-1 profile**)
     **J**/**5X,**J**/**10**)
READ(5,1000)C(I)
1008 FORMAT(A10)
10 CONTINUE

OPEN(UNIT=1,FILE='NOLIN2.DAT',STATUS='NEW',FORM='FORMATTED')
DO 15=1,NEXP
     WRITE(1,1008)C(I)
1011 FORMAT(D15.6)
15 CONTINUE

CLOSE(UNIT=1)
WRITE(5,1013)
1013 FORMAT(5X,**Values of dependent variables**)
     **J**/**5X,**J**/**10**)
OPEN(UNIT=1,FILE='NOLIN3.DAT',STATUS='NEW',FORM='FORMATTED')
DO 30=1,ND
     WRITE(5,1014)
1014 FORMAT(5X,**Ye(1,4)**)
READ(5,1011)YE(I)
WRITE(1,1011)YE(I)
30 CONTINUE
CLOSE(UNIT=1)
WRITE(5,1015)
DO 40=1,NI
     WRITE(5,1016)
1015 FORMAT(5X,**Names of the parameters to be fitted**)
     **J**/**5X,**J**/**10**)
READ(5,2000)NOMBRP(I)
40 CONTINUE
OPEN(UNIT=1,FILE='NOLIN4.DAT',STATUS='NEW',FORM='FORMATTED')
DO 42=1,NI
     WRITE(1,2200)NOMBRP(I)
42 CONTINUE
CLOSE(UNIT=1)
WRITE(5,1017)
DO 50=1,NI
     WRITE(5,1018)
1017 FORMAT(5X,**CONSTRAINTS**)
     **J**/**5X,**J**/**10**)
OPEN(UNIT=1,FILE='NOLIN5.DAT',STATUS='NEW',FORM='FORMATTED')
DO 50=1,NI
     WRITE(1,1011)BMN(I)
50 CONTINUE
CLOSE(UNIT=1)
PROGRAMA PRINCIPAL - REGRESION NO LINEAL -

IMPLICIT REAL*8(A-H,O-Z)
DOUBLE PRECISION NOMBRP,NOMBRX
INTEGER ACTIV(5)
DIMENSION REST(18),BMIN(18),BMAX(18),B(18),X(5),G(5),F(5).
G(5,5),BOUND(2,10),W(100),IACT(10)
EXTERNAL OBJN,OBJGT,NI,FYT,N,NCGT
CHARACTER*16 C(30)

COMMON/DATOS/ND,NL,NIV,YC(200),NOMBRX(20),
NOMBRP(18),YEXP(200),NEXP
COMMON/PROFILES/
COMMON/ADIMEN/FACT1,FACT2,FACT3,FACT4

C ---- LECTURA DE LOS DATOS DE ARCHIVO -----

OPEN(UNIT=1,FILE='NOLIN1.DAT',STATUS='OLD',FORM='FORMATTED')
READ(1,1000)XEXP,ND,NL,NIV
1000 FORMAT(155)
CLOSE(UNIT=1)

1001 FORMAT(156)

OPEN(UNIT=1,FILE='NOLIN5.DAT',
STATUS='OLD',FORM='FORMATTED')
DO 10 I=1,NI
   READ(1,1001)BMIN(I)
   READ(1,1001)BMAX(I)
10 CONTINUE
CLOSE(UNIT=1)

OPEN(UNIT=1,FILE='NOLIN4.DAT',
STATUS='OLD',FORM='FORMATTED')
DO 105 I=1,NI
   READ(1,1003)NOMBRP(I)
105 CONTINUE
CLOSE(UNIT=1)

C ------- LEER VALORES DE CADA PARÁMETRO ------

WRITE(6,1500)
1500 FORMAT(5X,'INITIAL GUESS FOR PARAMETERS: ',5X,
      364,'.')
DO 30 I=1,NI
   WRITE(6,1501)NOMBRP(I)
30 CONTINUE
CLOSE(UNIT=1)

FORMAT(4X,AR," =")
INITIAL GUESS FOR PARAMETERS
DO 2000 I=1,NI
  X(I)=R(I)
2000 CONTINUE

INITIAL GUESS FOR THE OBJECTIVE FUNCTION VALUE
FORJ=0.1

LOWER AND UPPER BOUNDS FOR PARAMETERS
DO 2001 J=1,NI
  BOUND(J,I)=BMIN(I)
  BOUND(J,I)=BMAX(I)
2001 CONTINUE

WEIGHTING FACTOR FOR PENALTY FUNCTION
PFWT=0.0

TERMINATION ERROR
ERROB=1.0D-07

NON-LINEAR CONSTRAINT VIOLATION ALLOWANCE. IT MUST BE > ERROB
ERROC=0.5E-07

MAXIMUM ALLOWED NUMBER OF EVALUATIONS OF FORJ
MXFORBJ=500

PRINT-OUT INDICATOR
IPRINT=1! VALUES PRINTED AFTER EVERY ITERATION

CALLING VMP
CALL VMP(NL,NC,NB,NX,FOBJ,GOBJ,FG,BOUND,ACTIVE,WPWT,ERROR,ERROC,MXFORBJ,IPRINT,FACTOBJ,OBJCN,OBJNI,OBJTNI,OBJTNC)

OPEN(UNIT=1,FILE='TERR DAT',STATUS='OLD',FORM='FORMATTED')

WRITE(1,*) 'ITERATION',','EVALUATIONS OF FORJ','ERRORS','MXFORBJ','PRINT-OUT','VALUES','PRINTED'
WRITE(1,1000) N, NL, K, NB, NX, FORJ, GOBJ, FG, BOUND, ACTIVE, WPWT, ERROR, ERROC, MXFORBJ, IPRINT

OPEN(UNIT=11,FILE='OBJCN DAT',STATUS='OLD',FORM='FORMATTED')

WRITE(1,2000) FORJ, GOBJ, FG, BOUND, ACTIVE

PRINTING RESULTS

OPEN(UNIT=1,FILE='RESULT DAT',STATUS='OLD',FORM='FORMATTED')

WRITE(1,3000)

WRITE(1,20000) N, NL, K, NB, NX, FORJ, GOBJ, FG, BOUND,-active, WPWT, ERROR, ERROC, MXFORBJ, IPRINT

CLOSE(UNIT=1)
CLOSE(UNIT=11)

WRITE(1,*) 'CALCULO DE LA COEFICIENTE DE VARIANZA'
WRITE(1,10001) VAR, DESV, DESV/SQRT(VAR), DESV/SQRT(VAR)
SUBROUTINE VMP

 PURPOSE

 THIS SET OF THREE SUBROUTINES (VMP,FENFUN,ACTSET) MINIMIZES
 A GENERAL FUNCTION SUBJECT TO LINEAR OR NONLINEAR CONSTRAINTS
 USING GRADIENT AND FUNCTION EVALUATIONS FOR BOTH OBJECTIVE
 FUNCTION AND CONSTRAINTS THE METHOD USED IS A VARIANT OF THE
 VARIABLE METRIC PROJECTION METHOD OF MURTAGH, B.A. AND R.W.H
 SARGENT- PROJECTION METHODS FOR NONLINEAR PROGRAMMING- PAPER
 PRESENTED AT THE INTERNATIONAL SYMPOSIUM ON MATHEMATICAL
 CAN BE OBTAINED FROM THE PROGRAM ADVISORY CHEM ENG ANY QUESTIONS
 PLEASE REFER TO:
 R. GANI

 THE PROBLEM IS FORMULATED AS FOLLOWS:

 MINIMIZE - FOB(X), X(n), J=1,2,.....NV
 SUBJECT TO - BOUND(I,J) LE X(I) LE BOUND(2,J), J=1,2,.....NB.
 WHERE - F(J), J=1,NL ARE NONLINEAR FUNCTIONS OF X
 F(J), J=NL+1,NC ARE LINEAR FUNCTIONS OF X
 F(J), J=NC+1,NB, J=1,NV (I.E. NB=NC+NV)

 THE OBJECTIVE FUNCTION ACTUALLY MINIMIZED IS:

 FOB(X)+PFWT PF(X)

 WHERE PF(X) IS THE SUM OF SQUARES OF THE CONSTRAINT VIOLATIONS
 AT X, BUT PF(X) IS NOT COMPUTED IF PFWT=0.

 NOTES:
 FOR EQUALITY CONSTRAINTS BOTH BOUNDS SHOULD BE SET TO
 THE DESIRED VALUE.
 FOR NONLINEAR CONSTRAINTS THE PROGRAM ALLOWS VIOLATIONS
 IF ERROC

 ARGUMENT LIST

 CALL VMP(NV,NL,NC,NB,NW,X,FOBJ,GBJ,BOUND,ACTIVE,WP,FWT,
            IF,ERROC,MXFORJ,ISTOP,FRMT,XACT,OBJFN,OBJG,NLCPH,NLCGT)

 NOTE: ALL REAL VARIABLES ARE IN DOUBLE PRECISION.

 X - ARRAY (DIMENSION NV) OF VARIABLES INITIAL ESTIMATE ON
 ENTRY SOLUTION ON EXIT (THE INITIAL POINT NEED NOT BE
 FEASIBLE.)

 FOB - VALUE OF OBJECTIVE FUNCTION NEED NOT BE SET ON ENTRY.
 BUT IF SET TO A GOOD ESTIMATE OF THE VALUE AT THE
 EXPECTED SOLUTION CONVERGENCE MAY BE IMPROVED.

 GBJ - ARRAY (DIMENSION NV) FOR OBJECTIVE FUNCTION GRADIENT.
 NOT SET ON ENTRY.

 F - ARRAY (DIMENSION NC) FOR CONSTRAINT FUNCTION VALUES.
 THE VALUES OF LINEAR CONSTRAINTS - F(J), J=NL+1,NC MUST
 BE SET ON ENTRY TO VALUES AT THE INITIAL POINT X OTHER
 VALUES NEED NOT BE SET.

 G - ARRAY (DIMENSION NV,NC) FOR CONSTRAINT GRADIENTS THE
 VALUES AT THE INITIAL POINT X OTHER VALUES NEED NOT BE SET.

 BOUND - ARRAY (DIMENSION 2, NB) FOR LOWER AND UPPER BOUNDS ON
 CONSTRAINT FUNCTIONS AND VARIABLES SET ON ENTRY

 ACTIVE - INTEGER ARRAY (DIMENSION NV) FOR ACTIVE CONSTRAINT
 NUMBERS NOT SET ON ENTRY ON EXIT ACTIVE(K), K=1,NACT
 INDICATE ACTIVE CONSTRAINTS AT SOLUTION WHERE
 ACTIVE(K) = +1 INDICATES UPPER BOUND OF J
 = -1 INDICATES LOWER BOUND OF J.

 IACT - ARRAY (DIMENSION NB) NOT SET ON ENTRY

 W - ARRAY (DIMENSION NW) OF WORKING SPACE.
 NW (GE NV+NL+NL+1+NB+NC)

 PFWT - WEIGHTING FACTOR FOR PENALTY FUNCTION SET ON ENTRY
ERROR - TERMINATION ERROR. SOLUTION IS INDICATED IF X IS
FEASIBLE AND
MAX(NORM2(C),NORM2(S),NORM2(G)) LE. ERROR.
ERROC - NON-LINEAR CONSTRAINT VIOLATION ALLOWANCE
SET TO A VALUE LT. ERROR.
MXFOBJ - MAXIMUM ALLOWED NUMBER OF EVALUATIONS OF FOBJ
ISTOP - INDICATES REASON FOR EXIT FROM SUBROUTINE VMP -
ISTOP = 1 : NORMAL ITERATION PROCEEDING
2: SOLUTION FOUND
3: NO FEASIBLE SOLUTION
4: MXFOBJ HAS BEEN REACHED
5: FAILURE TO FIND DESCENT STEP.BUT
TERMINATION CRITERION FOR SOLUTION NOT
SATISFIED (DUE TO ROUNDING ERRORS, OR
FAULTS IN USER-SUPPLIED DATA )
IPRINT - PRINT-OUT INDICATOR. SET ON ENTRY -
IPRINT = 1 - PRINTS VALUES ON EXIT FROM VMP AND
DIAGNOSTIC MESSAGES ONLY.
0 - NO PRINT-OUT AT ALL.
N - (N IS ANY POSITIVE INTEGER) VALUES
PRINTED AFTER EVERY N ITERATIONS,
TOGETHER WITH DIAGNOSTIC MESSAGES.
NOTE - THE LAST FOUR ARGUMENTS ARE THE NAMES OF USER-SUPPLIED
SUBROUTINES THE USER MAY UTILISE ANY SUBROUTINE NAME HE
WISHES (APART FROM VMP, ACTSET AND PENFUN) BUT HE MUST
DECLARE THEM IN THE CALL TO VMP, AND IN AN EXTERNAL
STATEMENT IN THE CALLING PROGRAM.
OBJFN - MUST EVALUATE FOBJ, GIVEN X(I), I=1,NV
ARGUMENTS - OBJFN(X,V,FOBJ)
OBJGT - MUST EVALUATE GOBJ(I), I=1,NV - GIVEN X(I), I=1,NV
ARGUMENTS - OBJGT(X,V,GOBJ)
NLCFN - MUST EVALUATE THE NONLINEAR CONSTRAINT FUNCTIONS,
F(J), J=1,NL - GIVEN X(I), I=1,NV
ARGUMENTS - NLCFN(X,NL,F)
NLCGT - MUST EVALUATE THE NONLINEAR CONSTRAINT GRADIENTS,
(G(J)), J=1,NL - GIVEN X(I), I=1,NV
ARGUMENTS - NLCGT(X,NL,G)
SUBROUTINE VMP (NV,NL,NC,NB,NX,FOBJ,GOBJ,F,G,BOUND.ACTIVE,W,PFWT
1,ERROB,ERROC,MXFOBJ,ISTOP,IPRINT,ACTOBJ,OBJFN,OBJGT,NL,CTN,NL,CTG)
EXTERNAL OBJFN,OBJGT,NLCFN,NLCGT
INTEGER ACTIVE(NV),LNPRESP
DIMENSION IACT(NB)
DOUBLE PRECISION FORJ,FOBJ,ALPHA,ALPHA1,GC,PFTP,GNORM,PNORM
DOUBLE PRECISION DETS,TR,STZ,TTQC,ZTGC,Z,PFWT,ERROR,ERROC
DOUBLE PRECISION REFD,STAB,SRN,SMA0,SD0,1.0D-30.1,1.0D-6.1,1.0D-30.1,1.0D+30
C.... SETS UP INITIAL CONDITIONS
1,P=0.
C WRITE(6,*)En VMP FOBJ,GOBJ,F,G,MXFOBJ,FOBJ,GOBJ(1),F(1)
C * (G(1),MXFOBJ
C write(p,*), nv = .nv, nl = .nl
C write(p,*), nc = .nc, nh = .nh, nw = .nw
C write(p,*), x = .x(1),=1,nv
C write(p,*), fo = .fo(1),=1,nb
C write(p,*), gb = .g(1)=1,nc
C write(p,*), f = .f(1),=1,nc
C write(p,*), g = .g(1),=1,nv, j=1,nc
C write(p,*), detecting bounds.
C write(p,*), x = .x(1),=1,2,j=1,nb
C write(p,*), w = .w(1),=1,nc
C write(p,*), erro prob = erro prob, erro = .erro
C write(p,*), f = .f(1),=1,nc
C write(p,*), istop = istop, iprint = iprint
C write(p,*), Desenze que la salida sea por pantalla ? (Si=1)
C read(5,*)NRESP
NRESP=1
HRESPN=1.1,P=8
MXCOR=10
ITI=0
ITI=0
ITI=0
ITI=0
ITI=0
ITI=0
ITI=0
ITI=0
ITI=0
ITI=0
ITI=0
ITI=0
NFOBJ=0
NGOBI=0
NFNL=0
NGN=0
NCOA=0
NSAC=0
NCTO=0
ISTOP=1
ISTOP=1
N1=NV+1
N2=NV+1
N3=N2A
N4=N2A
N1=N2A
N2=N2A
N4=N3+NV
N5=N4+N*L+NV
N6=N5+NV
N7=N6+NB
N8=N7+NC
N9=N8+NV
N=NV+NV
IF (N L.E. NW) GO TO 1
WRITE (LP,50) N
STOP
1 I=IFEZ
CALL PENTOSU(NV,NL,NC,NB,X,F0BJ,G0BJ,F,G,BOUND,PFWT,ERROB,ORJN,OB
JGT,NLCFN,NLCGT)
IF (IFEZ.EQ.1) GO TO 3
2 ID=2
GO TO 1
3 IF (PRINT.EQ.0) GO TO 6
IF (NC.NE.0) GO TO 4
C WRITE (LP,52) F0BJ
GO TO 5
4 WRITE (LP,52) F0BJF
C... SETS S=1
5 IF (PRINT.NE.0) WRITE (LP,57) X
6 DO T=1,1,N1
7 W(1)=0
8 DO 8 I=1,NV
9 I=I+1
10 W(I)=G0BJ(I)
IF (NC.EQ.0) GO TO 13
11 DO 10 J=1,NC
12 J=J+1
13 W(I)=F(I)
14 DO 15 J=1,NB
15 J=J+1
16 IF (J.GT.NC) J=J+N2+1
17 IF (W(J).GT.W(J)) GO TO 18
18 C... PROJECTS INTO ACTIVE CONSTRAINTS.
CALL ACTSET(NV,NL,NC,NB,X,G0BJ,F,G,BOUND,ACTIVE,WERROB,ALPHA,
HACT,NLCFN)
N0C=NCOR+1
C... TESTS FOR FEASIBILITY.
IF (ID.EQ.3) IT2=IT2+1
IF (PRINT.EQ.0.AND.ID.TE.5.AND.ID.EQ.3) WRITE (LP,53) ITNO
GO TO (16,17), IFEZ
19 IFEZ=2
WRITE (LP,55) ITNO,F0BJ,F0BJF,NF0BJ,NG0BJ,NGNLC,NCOR,NCAS
IF0BJ=0
GO TO 2
C... EVALUATES OBJECTIVE FUNCTION.
17 CALL OBFIN(NV,X,F0BJ)
C... WRITE(6,*)'SALIO DE OBJFN #1'
C IF (PRINT.NE.0.AND.IF0BJ.EQ.0) WRITE (LP,51) TIMEH,TIMEM,TIMES
IF0BJ=IF0BJ+1
IF0BJ=IF0BJ+1
C... COMPUTES GC,GNORM,PNORM,GT.
18 GTP=0
GNORM=0
PNORM=0
DO 18 I=1,NV
11 N=NV+I
12 F=W(I)*ALPHA
13 N=N+1
14 D2=N2+1
15 G2=N2+1
16 GC=G2+1
C... WRITE2(*)'GRADIENTE DE OBJ.',GC
W(I)=GC
GTP=GTP+W(I)*P
GNORM=GNORM+GC*GC
16 PNORM=PNORM+P*P
GTP=ABS(GTP)
PNORM=DSQRT(PNORM)
GNORM=DSQRT(GNORM)
IF (ID.EQ.3.OR.LD.EQ.2) GO TO 21
C... TERMINATION TEST.
IF (DFMAX(NORM,PNORM,GTP,GT)TERROB) GO TO 19
ISTOP=2
GO TO 25
19 IF (INFOBJ.LT.MXXF0BJ) GO TO 20
ISTOP=4
GO TO 25
C... STABILITY TEST.
20 IF (INFOBJ.FF.STAB*GTP*ALPHA) GO TO 24
IF (IF0BJ.EQ.1) GO TO 24
C... STEP SIZE REDUCTION
ALPHA = RED * ALPHA
IF (ALPHA > L2 * CORR) GO TO 31
C. RESYN S = 1 IF STEP TOO SMALL
IF (FPNORM / ALPHA GT ERROR) GO TO 14
IF (ISTOP EQ 5) GO TO 25
ISTOP = 5
IT = IT + 1
IF (IPRINT NE 0) ND IT = 1
IT = IT + 1
WRITE (LP, 54) ITNO
DO 22 = 1, NC
22 F(I) = W(N+I)
DO 23 = 1, NV
23 X(I) = W(NZ+I)
FOBJ = FOBJ + 1
GO TO 6
C. PRINTS RESULTS
24 IF (IPRINT LE 0) GO TO 33
IF (MOD(ITNO, IPRINT) NE 0) GO TO 33
25 IF (IPRINT EQ 0) GO TO 23
WRITE (LP, 55) ITNO, NFBOBJ, NGNLC, NGNOBJ, NGNLC, NGC, NASC
WRITE (LP, 56) GNORM, PNORM, GTP, FOBJ, F
WRITE (LP, 57) X
251 CONTINUE
GO TO (33, 26, 29, 30, ISTOP)
26 GO TO (28, 27), IFEZ
27 IF (IPRINT NE 0) WRITE (LP, 58)
IF (FAIL = 2)
GO TO 32
28 ISTOP = 3
C. WRITE(2,*) ISTOP = EXIT
IF (IPRINT NE 0) WRITE (LP, 59)
GO TO 32
29 IF (IPRINT NE 0) WRITE (LP, 60) MXFOBJ
GO TO 32
30 GO TO (28, 31), IFEZ
31 IF (IPRINT NE 0) WRITE (LP, 61)
32 IF (FAIL = 1)
RETURN
C. COMPUTES NEW GRADIENTS
33 DO = CALL PENFUN (NV, NL, NC, NB, X, FOBJ, GOBJ, FG, BOUND, FWFT, ERRO, OBINF, OB
1 JGT, NL, GCLT, NLGCLT)
IF (NL EQ 0 OR FWFT NE 0) GO TO 34
CALL NLGCLT (NV, NL, X, G)
NGNLC = NGNLC + 1
C. UPDATES MATRIX S
C. COMPUTES P AND QC
34 DO 35 = 1, NV
35 W(I) = GOBJ(I) * W(I)
IF (NL EQ 0 OR NACT EQ 0) GO TO 38
DO 37 K = 1, NACT
J = IABS(ACTIVE(K))
IF (J GT NL) GO TO 37
J = N+1
K = N+1
DO 36 W(I) = W(I) * W(J) * W(K)
37 CONTINUE
C. COMPUTES Z, ZT, ZTQC AND ZTGC
38 ZTQC = 0
ZTQC = 0
DO 40 = 1, NV
40 I = N+1
Z = W(I)
IF (I GT 1) I = I + 1
GO TO 40
Z = Z * W(I2) * W(I1)
W(I2) = Z
ZTQC = ZTQC * ZT * W(I1)
ZTQC = ZTQC * ZT * W(I2)
GO TO 42
C. DENOMINATOR AND POSITIVE-DEFINITNESS TESTS
Z = PARS(ZTQC * TRS)
IF (ZTQC, LE CORR) GO TO 9
C = ZTQC
IF (ZTQC, LE Z/CO) GO TO 41
C = C/IT + 1
C = ZTQC * TRS
GO TO 42
C = ZTQC * DETS GT 0 DO GO TO 74
C = C * ZTQC * ZTGC * ALPHAS * DETS * GT * (SMIN) DO GO TO 42
C = C * IT + 1
C = C * IT + 1
C = C * IT + 1
C = ZTQC * TRS
C. UPDATES S, DETS AND TRS
42 C = 0
C = 0
DETS = DETS + 1
DETS = DETS + 1
DETS = DETS + 1
IF (C GT ZTQC) ZTQC = ZTQC + 1
DO 43 = 1, NV
43 I = N+1
W(I) = W(I) + 1
Z = W(I)
DO 43 = 1, NV
J = 1
J = 1
I = N+1
FORM=D10.7X,29OBJECIVE FUNCTION TOLERANCE=D10.7X OBJECT
2IVE FUNCTION VALUE =D18.7X0H CONSTRAINT FUNCTION VALUES,2018
38.0X,2D18.8H)
57 FORMAT (22H VARIABLE VALUES(X),8X,2D18.8H)
58 FORMAT (15H SOLUTION FOUND)
59 FORMAT (21H NON FEASIBLE SOLUTION)
60 FORMAT (22H NON CONVERGENCE WITHIN 14.31H OBJECTIVE FUNCTION EVALUATIONS)
61 FORMAT (39H ALGORITHM FAILS TO OBTAIN DESCENT STEP)
62 FORMAT (9H0F PRECIPES/4H STEP TOO SMALL (S=1))
63 FORMAT (1 Correction Convergence Failure, 15.2H1H) DENOMINATOR TOO SMALL
64 FORMAT (15H1H, POSITIVE DEFINITENESS FAILURE, 15.2H1H) MATRIX
65 FORMAT (3S SCALING DOW NA4H L1 MATRIX S) SCALING UP
66 FORMAT (4S4H MATRIX S.HL, CONDITIONED (S=1),15)
67 FORMAT (39H0VFMP EXECUTION TIME=,I2,'H:,X12,'Min':X12,'Sec':I2)
68 FORMAT (6X,17H*** ITERATION NO.,33,3H,2H UPDATE DENOMINATOR TOO SM
69 SMALL)
70 FORMAT (6X,17H*** ITERATION NO.,33,3H,2H POSITIVE DEFINITENESS FAIL
71 URE)
72 FORMAT (6X,17H*** ITERATION NO.,33,3H,2H MATRIX S SCALING DOWN,6X,1)
73 FORMAT (17H6X,17H6X,1H H6X)6X,17H6X,1H6X,1H H6X)6X,17H6X,1H H6X)6X,17H6X,1H6X,1H
74 FORMAT (6X,17H*** ITERATION NO.,33,3H,45H MATRIX S ILL CONDITIONED,
75 1X,1H UNIT MATRIX,4X,17H 6X,17H 6X,17H 6X,17H
76 END TO UNIT MATRIX,4X,17H 6X,17H 6X,17H 6X,17H
77 SUBROUTINE ASCTET (NV,1L,NC,IB,NNW,GOBI,FG,BOUND,ACTIVE,W.ERROR.
78 ALPHA,JACTIN,CNT)
79 DOUBLE PRECISION X(NV),GOB(NV),F(S),G(NV),BOUND,2NB,W(NW)
80 DOUBLE PRECISION SG.FN,UG,RO,RO.R.BETA,1AMBD,1AMB.D.1DEL}
81 F Também,6X
82 FORMAT (6X,17H*** ITERATION NO.,33,3H,45H MATRIX S ILL CONDITIONED,
83 1X,1H UNIT MATRIX,4X,17H 6X,17H 6X,17H 6X,17H
84 END TO UNIT MATRIX,4X,17H 6X,17H 6X,17H 6X,17H
85 SUBROUTINE ASCTET (NV,1L,NC,IB,NNW,GOBI,FG,BOUND,ACTIVE,W.ERROR.
86 ALPHA,JACTIN,CNT)
87 DOUBLE PRECISION X(NV),GOB(NV),F(S),G(NV),BOUND,2NB,W(NW)
88 DOUBLE PRECISION SG.FN,UG,RO,RO.R.BETA,1AMBD,1AMB.D.1DEL}
89 F Também,6X
90 IF (PRINT.NE.0.0 AND ITS.LE.5) WRITE (LP,60) ITNO,EMIN,EMAX
91 GO TO 47
92 IF (EMIN.ES.GE.MIN) GO TO 9
93 C WRITE(5,*)EMIN,EMAX
94 SCALE=EMIN GT MAX
95 GO TO 49
96 IF (PRINT.NE.0.0 AND ITS.LE.5) WRITE (LP,67) ITNO,EMIN,EMAX
97 GO TO 47
98 WTH(=SCALE*W(I))
99 GO TO 8
100 IF (PRINT.NE.0. AND ITS.LE.5) WRITE (LP,68) ITNO,EMIN,EMAX
101 GO TO 6
102 C FORMAT (32H PERFECT VMP - NW SHOULD BE AT LEAST,14)
103 C FORMAT (41H HEOOSIBLE POINT FOUND EXECUTION TIME=,I2,'H:',X12.
104 C ' Min':X12,'Sec':I2)
105 C ' Min':X12,'Sec':I2)
106 C ' Min':X12,'Sec':I2)
107 C ' Min':X12,'Sec':I2)
108 C ' Min':X12,'Sec':I2)
C** SETS UP ACTIVE SET
K=1
DO 13 K=K1,NB
11 I=IACT(K)
GO TO (7,13,13), 11

DO 16 I=NK,K,-1
13 NL=NL+1
DO 16

J=1
DO 10 J=J1,JN+1,1
10 DELR=0.5D0*(BOUND(2,K)+BOUND(1,K))
DO 10
IF (ERROR GT DELR) GO TO 11
IF (K LE JEQ) GO TO 13
11 RN=RN(J)+DELK
13 CONTINUE
GO TO 14

DO 17 L=1,NV
17 RN=L+1
IF (I GT L) H=H*L+1
17 CONTINUE
RN=RN+1

DO 18 I=1,NV
18 RN=RN+1
IF (I EQ L) H=H*L+1
18 CONTINUE
RN=RN+1

DO 19 I=1,NV
19 RN=RN+1
IF (I EQ L) H=H*L+1
19 CONTINUE
RN=RN+1

C** COMPUTES R U AND U*U R U
NACT=T1+1
RN=RN+1
R=BOUND(1,K)
DO 20 I=I+1,1
1 RN=RN+1
DO 20
20 CONTINUE

C** TESTS FOR LINEAR DEPENDENCE
DO 22 I=1,NV
22 RN=RN+1
IF (RN EQ I) GO TO 23
23 CONTINUE

C** UPDATES R
RN=RN+1
DO 24 I=I+1,1
24 RN=RN+1
IF (RN EQ I) GO TO 25
25 RN=RN+1
IF (RN EQ I) GO TO 26
26 CONTINUE

C** ADDS CONSTRAINTS TO ACTIVE SET
K=K1
DO 27 I=I+1,1
27 RN=RN+1
IF (I EQ K) GO TO 28
28 CONTINUE

C** UPDATES W
DO 29 I=1,NV
29 CONTINUE

C** SETS UP ACTIVE SET
K=1
DO 32 K=K1,NB
I=IACT(K)
GO TO (7,13,13), 11

DO 34 I=NK,K,-1
32 NL=NL+1
33 CONTINUE
J=1
DO 34 J=J1,JN+1,1
34 RN=RN(J)+DELK
35 CONTINUE
J=1
DO 35 I=1,NV
36 RN=RN+1
IF (I EQ L) H=H*L+1
36 CONTINUE
RN=RN+1

DO 37 I=1,NV
38 RN=RN+1
IF (I EQ L) H=H*L+1
39 CONTINUE
RN=RN+1

DO 40 I=1,NV
40 RN=RN+1
IF (I EQ L) H=H*L+1
41 CONTINUE
RN=RN+1

C** COMPUTES R U AND U*U R U
NACT=T1+1
RN=RN+1
R=BOUND(1,K)
DO 42 I=I+1,1
4 RN=RN+1
DO 42
42 CONTINUE

C** TESTS FOR LINEAR DEPENDENCE
DO 44 I=1,NV
4 RN=RN+1
IF (RN EQ I) GO TO 45
45 CONTINUE

C** UPDATES R
RN=RN+1
DO 46 I=I+1,1
4 RN=RN+1
IF (RN EQ I) GO TO 47
47 CONTINUE

C** UPDATES W
DO 48 I=1,NV
49 CONTINUE
K1=NROW+L
26  W(I)=W(I)+RU*W(K1)
27  NACT=NACT+1
28  NASC=NASC+1
29  ACTIVE=NACT=1*IBND
30  K=NACT
31  C******** FINDS GT.S GO
32  K1=N5+K
33  W(I1)=0.000
34  NROW=K*(K-1)/2+N1
35  SG=0.000
36  J1=ABS(ACTIVE(I))
37  DO 30 I1=1,NV
38        IROW=I1*(I1)/2
39        FN=0.000
40        DO 28 L=1,NV
41              II=IROW+L
42              IF (L.GT.I1) II=I1*(L-1)/2+1
43              IF (L.EQ.I1) II=I1*(I1)/2+1
44        28  CONTINUE
45  30  CONTINUE
46  C******** FINDS MAXIMUM BETA
47  K=0
48  IF (NACTEQ 1) GO TO 4
49  BETA=0.000
50  DO 41 I=1,NACT
51        IROW=I1*(I1)/2+N1
52        J1=ABS(ACTIVE(I))
53        I2=N5+1
54        I4=IROW+1
55        IBND=ISGN(1.ACTIVE(I))
56        IF (0.5*BOUND(1,J1)-BOUND(1,J1)) LE ERROR) IBND=0
57        LAMBDA=W(I2)
58        GAMMA=W(I3)
59        DLAMB=DABS(LAMBDA)
60        IF (IBND) 77,41,38
61  40  CONTINUE
62  41  CONTINUE
63  IF (LAMBDA.LT.DLAMB) GO TO 41
64  GO TO 39
65  IF (LAMBDA.GE.DLAMB) GO TO 41
66  FN=0.000
67  DO 40 L=1,NACT
68        I4=IROW+L
69        IF (L.GE.J1) I4=I1*(L-1)/2+1+N1
70        J1=ABS(ACTIVE(L))
71        IBND=ISGN(1.ACTIVE(L))
72        IF (0.5*BOUND(1,J1)-BOUND(1,J1)) LE ERROR) IBND=0
73        J2=L+J1
74        R=W(I1)-BOUND(J2,J1)
75  40  CONTINUE
76  IF (FN+R*W(I4)<0) GO TO 41
77  IF (FN+R*W(I4)) GAMMA
78  IF (FN+R*W(I4)) 41,77,39
79  IF (FN+R*W(I4)) BETA=F
80  CONTINUE
81  K=1
82  41  CONTINUE
83  K1=K+1
84  C******** INTERCHANGES LAST ACTIVE CONSTRAINT WITH THE REJECTED
85  K
86  IF (BETA.LE.ERROR) GO TO 4
87  NACT=NACT-1
88  NROW=NACT*NACT+2*N1
89  IF (1.EQ.NACT) GO TO 43
90  IF (1.NE.NACT) GO TO 43
91  IROW=I1*(I1)/2+N1
92  J1=ACTIVE(I)
93  ACTIVE(I)=ACTIVE(NACT)
94  ACTIVE(NACT)=J1
95  DO 42 K=1,NACT
96        I1=IROW+K
97        IF (K.GT.K1) II=K*(K-1)/2+N1+1
98        I1=K-NROW+K
99  42  CONTINUE
100
R=W(I)
W(I)=W(K1)

42 W(K1)=R
I=1:ROW+1
K1=NROW*1
R=W(I)
W(K1)=R
I=N5+1
K1=N5+NACT
R=W(I)
W(I)=W(K1)
W(K1)=R

C******** R, LAMBDA AND GC
43 K1=NROW+NACT
R0=1.0D0/W(K1)
K1=N5+NACT
LAMBD=1/W(K1)
DO 48 K1=1,NACT
K=KK
JIA=ABS(ACTIVE(K))
K1=NROW+K
RU=RD*W(K1)
DLAMB=LAMDA*RU
IF (K.EQ.NACT) GO TO 45
K1=N5+K
W(K1)=W(K1)+DLAMB
IROW(K-K-1)*Z2+NI
DO 44 L=1,K
I=IROW+L
K1=NROW+L

44 W(I)=W(I)+RU*W(K1)
45 IF (J1.GT.NC) GO TO 47
DO 46 J=1,NC
I=N8+1
46 W(I)=W(I)+G(J)*DLAMB
GO TO 48
47 J=1:NC
D=N8+1:NC
W(I)=W(I)+DLAMB
48 CONTINUE
NACT=NACT1
NASC=NASC+1
GO TO 36

C******** FINDS NEW POINT
49 DF=0.000
DO 50 J=1,NC
I=N6+NC+1
J=N8+1
GAMMA=ALPHA*W(I2)
R=W(I1)+GAMMA.X(I)
X(I)=W(I1)+GAMMA
50 IF (DF+R*R) GO TO 51

CALL NLCFN (NVNL, X, F)
NFCN.C=NFCN.C+1
51 IF (NC.FEQ.NL) GO TO 54
I=NL+1
DO 53 J=1,NC
SG=0.000
DO 52 K=1,NC
K=NK+1
52 SG=SG+G*(K,J)*W(K1)
53 F(I)=W(N6+1)*SG*ALPHA
54 DO 57 J=1,NC
I=N6+1
IF (J.GT.NC) GO TO 56
SG=0.000
F(N)=F(I)
DO 55 K=1,NC
K=NK+1
55 SG=SG+D*W(K1)*G(K,J)
GO TO 57
56 K=N4+1:NC
F(N)=X(I:NC)
SG=W(K1)
57 W(I)=F*N-SG*ALPHA
ALPHA=ALPHA
IF (DFLT.EROR OR NL.FEQ.0) RETURN
ITCOR=ITCOR+1
IF (ITCOR.LT.MXCOR) GO TO 1
ID=1
RETURN
FDN
SUBROUTINE PENFUN (NVNL, NC, NL, XI, XOBJ, GOBJ, G, LBND, PFWT, ERROR, OB
1JEN, OBGET, NLCTN, NLCT)
C . COMPUTES PENALTY FUNCTION AND ITS GRADIENT
C . ID=1 COMPUTES F AND G GIVEN NO VALUES
C . =2 COMPUTES F AND G GIVEN CONSTRAINT FUNCTION AND GRADIENT VAL
C . =3 COMPUTES F GIVEN CONSTRAINT FUNCTION VALUES
C . =4 COMPUTES G GIVEN ALL FUNCTION VALUES
EXTERNAL OBJFUN, OBJGET, NLCTN, NLCT

DOUBLE PRECISION X(INV), GOR(INV), F(S), GOR, LBND(2, NB)
DOUBLE PRECISION PFWT, PE, PFWE.PFWT, ERROR, ALPHA
COMMON XVMY, NFORB.IOBJ, NOBJ, NLNLC, NLNC, LAUG, NACT, TNO, ITCOR, MXCOR, L
ID, N1, N2, N3, N4, N5, N6, N7, N8, N9, NACT, FJ, JF, I, I1, ALPHA
C . OBTAINS REQUIRED FUNCTION AND GRADIENT VALUES
PE=0.000
GO TO 57, 1, IFEZ
1 PE=PFWT
GO TO 2, 2, 2, 3, ID
2 CALL OBJFN (NVX, XI, GOR)
C WRITE(6, 'YSAILOF ORFJN#')
NFORB.I=NFORB(1)
GO TO 3, 3, 4, 3, ID
3 CALL OBJGT (NVX, GOR)
NOBJ=NOBJ+1
4 IF (PFWT.LT.P) RETURN
GO TO 9
5 PFW=1.000
GO TO (6,6,6,7), ID
6 FDBL=0
GO TO (7,7,9,7), ID
7 DO I=1,11,1
8 GOBI(I)=0.0
9 IF(=2)
IF (NB EQ 0) RETURN
IF (NL EQ 0) GO TO 13
GO TO (10,11,1,11), ID
10 CALL NLCFN(NV,NLX,F)
NPNLC=NPNL+1
11 GO TO (12,12,13,12), ID
12 CALL NLCGT(NV,NLX,G)
NGLCN=NGLCN+1
C... COMPUTES CONSTRAINT VIOLATIONS
13 DO 22 I=1,16
IF(J GT NC) GO TO 14
FN=F(J)
GO TO 15
14 I=J-NC
FN=X(I)
15 IF (FN GE BOUND(J,JERROR) GO TO 16
FN=BOUND(J)
IFZ=1
GO TO 18
16 IF (FN LE BOUND(J,JERROR) GO TO 17
FN=BOUND(J)
IFZ=1
GO TO 18
17 FN=0.0
18 PFS=PFS+FN*FN
GO TO (19,19,19,21), ID
19 IF (J GT NC) GO TO 21
DO 20 I=1,11,1
20 GOBI(I)=GOBI(I)+PFW*FN*G(J)
GO TO 22
21 I=J-NC
GOBI(I)=GOBI(I)+PFW*FN
22 CONTINUE
GO TO (23,23,23,24), ID
23 FDBL=FDBL+1.500*PFW*PF
24 RETURN
END

SUBROUTINE FUNC(B,YCAL,FDBL)

C IMPLICIT REAL*(A-H,O-Z)
DOUBLE PRECISION NOMBRE,NOMBRP
DIMENSION YCAL(200),B(18), TEMP(500),TEMPO(5000)
CHARACTER*10(C)
COMMON/NATOS/ND,NI,NV,YE(200),NOMBRX(20),
NOMBRP(18), YEX(1,200)
COMMON/PARAM/BD(5),EA,NUMDARIO
COMMON/PROFILE/C
COMMON/PROFIL/NDATOS,TEMP
COMMON/TP/TEMPO,TEMAX,ZMAX
COMMON/ADIMN/FACT1,FACT2,FACT3,FACT4
DO 2 I=1,5
B(I)=B(I)
2 CONTINUE
BR(1)=BR(1)*FACT1
BR(2)=BR(2)*FACT2
BR(3)=BR(3)*FACT3
BR(4)=BR(4)*FACT4
C C C
C C
C C
C C
C C
C C
C C
C OPEN(UNIT=1,FILE=CON, STATUS='OLD',FORM='FORMATTED')
C NN=0
C TEMAX=0
C DO 1000 INT=1,1000
C READ(*,*) ND,TEMAX,TMAX,TEMAPINT,TEMPINT,
IF(TEMPINT GE TEMAX)
C TMAX=TEMPINT
C ZMAX=TEMPINT
C ELSE
C ENDIF
C NN=NN+1
1000 CONTINUE
C C 2000 CONTINUE
C C C C C C
C WRITE(*,*) EN FUNC ZMAX, TEMAX, ZMAX,TEMAPINT,TEMPINT,
C ZT=Z-2*SIGMA
C ZB=ZT+2*SIGMA
C ZF=ZT-2*SIGMA
C I=1 NUMER OF POINTS TO BE EVALUATED FROM Z TO ZF
C IT=50
C CALL SIMPSON(20,50,IT,TEMPINT)
C YCAL=BR(1)*AINT
C WRITE(*,*) YCAL, YCAL**, YCAL**, YCAL**, YCAL**
C CONTINUE
C 1.0150(UNIT=1,)
C
C CALCULO DE LA DESVIAICION ESTANDAR Y VARIANZA
C
VAR=DESV/(ND-N1)
IF(VAR.LT.0) VAR=VAR
DESV=SQRT(VAR)
FIN=DESV
C WRITE(6,*) FOBI,FOBJ
RETURN
END

C SUBROUTINE FUN : DETAILS OF THE EQUATION 1 TO BE INTEGRATED
SUBROUTINE FUN(Z,F)
IMPLICIT REAL*(A-H,O-Z)
CHARACTER*10 CD(10)
DIMENSION TEMP(5000),TIEMPO(5000)
COMMON/DATOS,N1,N2,ND,NI,YE(20),NOMBRX(20),
HOBRE(18),XEP(200)
COMMON/PARAM(5),EA,NUMDATO
COMMON/PROFILES,C
COMMON/T PROFILE,N/DATOS,TEMP
COMMON/TIEMPO,TEM,ŽMAX
COMMON/ADIMEN/FAC1,FAC2,FAC3,FAC4
C INFORMATION FOR THE SECOND INTEGRAL
C WRITE(6,*)'EN FUN B(2)=SIGMA',B(2)
EA=Z
C WRITE(6,*)'EN FUN EA****', Z
Z=0
Z=Z+ZMAX
C WRITE(6,*)'EN FUN ZMAX, TEMAX,ŽMAX,TEMAX
NIT=(ZMAX-0.0001)
CALL SIMPSON4(2,Z2,EF,NTT,NTT,AIN=2)
C WRITE(6,*)'EN FUN AIN=2,AIN=2
Z=TIEMPO(NDATOS)
NIT=(Z2-ZMAX)/0.2
CALL SIMPSON4(2,Z,EN,NTT,AIN=3)
C WRITE(6,*)'EN FUN AIN=3,AIN=3

C AIN=AIN+AIN+AIN
C
C THE FOLLOWING EQ. REPRESENTS THE SOIL WEIGHT LOSS
C
P=1.14152654
CTE=1/(2.*PI**0.5)
IF(AINT2+1.3<100) THEN
XX=0
ELSE
XX=DEXP(-AINT2)
ENDIF
AINT=1.2(B1**2+B2**2++)
IF(AINT LE 1.1) THEN
XX1=0
ELSE
XX1=DEXP(-AINT1)
ENDIF
F=CTE*(B1**2+1)**XX1
RETURN
END

C SUBROUTINE FUN1 : DETAILS OF THE EQUATION 2 TO BE INTEGRATED
SUBROUTINE FUN1(Z,F)
IMPLICIT REAL*(A-H,O-Z)
DIMENSION TEM(5000),TIEMPO(5000)
COMMON/DATOS,N1,N2,ND,NI,YE(200),NOMBRX(20),
HOBRE(18),XEP(200)
COMMON/PARAM(5),EA,NUMDATO
COMMON/PROFILES,C
COMMON/T PROFILE,N/DATOS,TEMP
COMMON/TIEMPO,TEM,ŽMAX
COMMON/ADIMEN/FAC1,FAC2,FAC3,FAC4
C
C WRITE(6,*)'EN FUN ZMAX, TEMAX,ŽMAX,TEMAX
DO 3 J=1,NN
IF(Z.EQ.TIEMPO(J)) THEN
T=TEMP(J)+273.
ELSE
IF(Z.GT.TIEMPO(J) AND Z.LT.TIEMPO(J+1)) THEN
WRITE(6,*)'****, EXPLOTARA'
T=TEMP(J+1)+273.-(TIEMPO(J+1)-Z)**
ELSE
WRITE(6,*)'EFUICOVADO'
ENDIF
ENDIF
CONTINUE
RETURN
END

C SUBROUTINE SIMPSON(70,ZEN,AINT)
IMPLICIT REAL*(A-H,O-Z)
CALL FUNC(X,Y,C,FOBJ)
FOBJ=(FOBJREF-FOBJ)
C
X1REF=X(1)
X2REF=X(2)
X3REF=X(3)
C
FIRST PERTURBATION
C
X(1)=1.01*X(1) "Nuevo valor de la Variable X(1)
X(1)=1.001*X(1) "Nuevo valor de la Variable X(1)
PERT1=X(1)
C
CALL FUNC(X,Y,C,FOBJ)
FOBJ1=FOBJ
X(1)=X1REF
C
SECOND PERTURBATION
C
X(2)=1.01*X(2)
X(2)=1.001*X(2)
PERT2=X(2)
C
CALL FUNC(X,Y,C,FOBJ)
FOBJ2=FOBJ
X(2)=X2REF
C
THIRD PERTURBATION
C
X(3)=1.01*X(3)
X(3)=1.001*X(3)
PERT3=X(3)
C
CALL FUNC(X,Y,C,FOBJ)
FOBJ3=FOBJ
X(3)=X3REF
C
gradient vector (GOBJ)
C
GOBJ(1)=(FOBJ1-FOBJREF)(PERT1-X(1))
GOBJ(2)=(FOBJ2-FOBJREF)(PERT2-X(2))
GOBJ(3)=(FOBJ3-FOBJREF)(PERT3-X(3))
C
RETURN
END
C
SUBROUTINE OBJFN(NV,X,FOBJ)
C
EVALUATES THE GRADIENT OF OBJECTIVE FUNCTION
C
SUBROUTINE OBJFN(NV,X,FOBJ)
C
IMPLICIT REAL*(A-H,O-Z)
DIMENSION X(NV),GOBJ(NV),YC(10)
C
CALL FUNC(X,Y,C,FOBJ)
RETURN
END
SUBROUTINE NLCP(NV, NL, X, F)
RETURN
END

SUBROUTINE NLCGT(NV, NL, X, G)
RETURN
END
HSTORE2.F
PROGRAM HSTORE2
C
C     PROGRAM TO STORE THE DATA TO BE ADJUSTED BY LEVENBERG-MARQUARDT METHOD
C
C     DOUBLE PRECISION Y(30), SIGMA
C     INTEGER NDATA, MA, I, J
C     CHARACTER*10 X(30)
C
C     WRITE(5,1000)
1000   FORMAT(/5X,**INPUT OF DATA TO BE ADJUSTED BY THE MODEL.**
* /5X,5F13.4/) WRITE(5,1002)
1002   FORMAT(5X,'Number of experiments/data points:') READ(5,1003) NDATA
1003   FORMAT(15) WRITE(5,1005)
1005   FORMAT(5X,'Estimated std. dev. for each data point:')
C
C     OPEN(UNIT=1,FILE='CRM DATA1.DAT',STATUS='NEW',FORM='FORMATTED')
C     WRITE(1,1011) SIGMA
C
C     WRITE(5,1006)
1006   FORMAT(/5X,'***File names containing the T-t profile***',
* /5X,47(':'))
C     DO 10 I=1,NDATA WRITE(5,1007)
1007   FORMAT(5X,'File name containing the T-t profile for',
* ' experiment #',J3,':') READ(5,1008) X(I)
1008   FORMAT(A10)
10   CONTINUE
C
C     Opening data files......
C
C     OPEN(UNIT=1,FILE='CRM DATA2.DAT',STATUS='NEW',FORM='FORMATTED')
C
C     DO 15 J=1,NDATA WRITE(1,1009) X(J)
1011   FORMAT(15.6)
15   CONTINUE
C
C     CLOSE(UNIT=1)
C
C     WRITE(5,1013)
1013   FORMAT(5X,'*** Values of dependent variables ***',
* /5X,42(':'))
C     OPEN(UNIT=1,FILE='CRM DATA3.DAT',STATUS='NEW',FORM='FORMATTED')
C PROGRAM CRM2
C*****************************************************************************
C CRM2 = Competitive Reaction Model No. 2
C*****************************************************************************
C This program uses the Levenberg-Marquardt Method to solve for the
C kinetic parameters of a set of competitive reactions by fitting
C to experimental data for the amount of product P formed.
C For each experimental data point to be fitted, LSODE.F (the
C Livermore Solver for ODEs) is used to predict the amount of
C product P formed over that data point’s corresponding
C experimental temperature-time profile for the parameter values
C guessed. For details on the reaction set used in this model,
C refer to subroutine PLSODE.
C
C The user will have already input into text files using HSTORE.F the
C following information:
C File name Information held
C
C CRMDAT1.DAT the no. of data points, total no. of adjustable
C parameters, and estimated std. dev. for each
C of the data points
C CRMDAT2.DAT names of files containing the temperature
C (Celsius)-time (sec) profiles for each
C data point/experiment
C CRMDAT3.DAT the values we want to fit for, i.e. the
C dependent variable values, in the same
C order as in CRMDAT2.DAT, to match their
C values with the corresponding T-t profiles
C
C This program is a combination of two programs: MRQMIN obtained from
C MIT's FORTRAN Recipes locker (mit/recipes/src/recipes_f/precipes)
C and PLSODE.F written by Hiroshi Saito. The multiple independent
C reaction model evaluations were removed from MRQMIN and passed
C in its place to PLSODE.F to numerically calculate the result of
C the system of ODE’s describing the reaction set (where a
C time-dependent temperature profile is imposed).
C*****************************************************************************
C Main Program - Non-linear regression
C
C The purpose of this level is to read in the variables needed to
C properly size the arrays passed to subroutine MRQMIN.
C
C NMAX = absolute max. no. of data points to be fit by model
C PARAMETER NMAX = 30
C INTEGR NDATA, I
C DOUBLE PRECISION SIGMA,Y(NMAX),SIG(NMAX),YPRED(NMAX)
C CHARACTER*10 X(NMAX)
C
C Initializing the required arrays
C
C*****************************************************************************
C
C DO 100 I=1,NMAX
C Y(I) = 0.0
C SIG(I) = 0.0
C YPRED(I) = 0.0
C X(I) = ' '  
C CONTINUE
C
C* READING IN: NDATA = number of data points input
C* SIGMA = estimated standard deviation for each data
C* point, estimated from exp. scatter
C* OPEN(UNIT=1,FILe=‘CRMDAT1.DAT’,STATUS=’OLD’,FORM=’FORMATTED’)
C* READ(1,1001) NDATA
C 1000 FORMAT(3H5)
C READ(1,1001) SIGMA
C 1001 FORMAT(D15.8)
C CLOSE(UNIT=1)
C CALL MAINSUB(NDATA,SIGMA,X,Y,YPRED,SIGMA)
C STOP
C END
C*****************************************************************************
C SUBROUTINE MAINSUB(NDATA,SIGMA,X,Y,YPRED,SIGMA)
C MA = total number of parameters PLSODE depends on
C NCA = Only requirement is NCA > no. of parameters to be adjusted
C INTEGER NDATA, I,J,M,P,E,DELT,A,B
C DOUBLE PRECISION X(NCA), Y(NDATA), YPRED(NDATA), SIGMA, ALAMDA.
C CHISQ, PCHISQ, YPRED(NDATA), DIF, ERR, DESV
C VAR
C CHARACTER*10 XNDATA)
C PARAMETER (MA = 5, NCA = 10)
C INTEGER IA(MA)
C DOUBLE PRECISION A(MA), B(MA), D(MA), COVAR(NCA, NCA),
C COVAR(NCA, NCA)
C CHARACTER*10 PNAME(MA)
C EXTERNAL PFUNC
C
C Assigning parameter names: PNAME(I) = name of parameter I
C
C PNAME(1) = 'Ep'
C PNAME(2) = 'Eh'
C PNAME(3) = 'A0'
C PNAME(4) = 'kgo'
C PNAME(5) = 'kbo'
C
C*****************************************************************************
C Prompting user for input of initial guesses for parameters to be fit
C*****************************************************************************
C*****************************************************************************
C*****************************************************************************
DO 29 I=1,NDATA
   YPRED(I) = 0.
29 CONTINUE
WRITE(6,1500)
1500 FORMAT(5X,'INITIAL GUESS FOR PARAMETERS',/5X,36('=',1X))
   DO 30 I=1,MA
      WRITE(6,1501)(PNAME(I))
30 CONTINUE
   WRITE(6,1502)(PNAME(I))
1501 FORMAT(4X,AR,'=')
   READ(3,1502)(BEST(I))
1502 FORMAT(25.12)
   A(I)=BEST(I)
30 CONTINUE
PRINT *,...
PRINT *, 'Of the 5 parameters input, the first N are to be'
PRINT *, 'adjusted by the program. N = ?'
1000 FORMAT(35)
   READ(3,1000)(MFFT)
   DO 31 J=1,MA
      IF (1.0 .LE. MFFT) THEN
         I(A(I)) = 1
      ELSE
         I(A(I)) = 0
      END IF
31 CONTINUE
C C
C Reading in: X(I) = name of data file containing the Temp./time
C profile for experiment/data pt. no. 1
C OPEN(UNIT=1,FILE='CRMDAT2.DAT',
   STATUS='OLD',FORM='FORMATTED')
   DO 36 J=1,NDATA
      READ(1,35) X(J)
36 CONTINUE
C C
C PRINT *, X(I)
C 35 FORMAT (A10)
   PRINT *, 'Read in File names'
   CLOSE(UNIT=1)
C C
C Reading in: Y(I) = experimental data values for experiment/data
C pt. no. 1 to be fit by the parameters
C OPEN(UNIT=1,FILE='CRMDAT2.DAT',
   STATUS='OLD',FORM='FORMATTED')
   DO 40 I=1,NDATA
1001 FORMAT (D15.6)
   READ(1,1001)Y(I)
40 CONTINUE
C C
C Assigning the std. deviation input by the user to all data points
C SIG(I) = array containing individual standard dev.
C for point 1
C DO 10 I = 1, NDATA, 1
   SIG(I) = SIGMA
10 CONTINUE
C C
C Initializing arrays COVAR and ALPHA
C DO 110 I = 1, NCA, 1
   DO 110 J=1, NCA, 1
      IF (1.0 .EQ. J) THEN
         IF (1.0 .LE. MFFT) THEN
            COVAR(I,J) = 1.
         ELSE
            COVAR(I,J) = 0
         END IF
      ELSE
         COVAR(I,J) = 0
      END IF
   110 CONTINUE
C C
C 1002 FORMAT (3X,'Covar-postinit', D15.6, D15.6)
C PRINT 1002, COVAR(1,1), COVAR(1,2)
C PRINT 1002, COVAR(2,1), COVAR(2,2)
C C
C 1005 FORMAT(3X,'post-init A(F)', D15.6, D15.6, D15.6)
C PRINT 1005, A(1), A(2), A(3)
C PRINT 1006, A(4), A(5)
C C
C Initializing CHISQ, PCHISQ, and ALAMDA for first pass into MRQMIN
C C
C CHISQ = 0
C PCHISQ = 0
C ALAMDA = -1.
C C
C Iterating using MRQMIN to fit parameters to the data until CHISQ
C decreases by less than 0.1 absolutely or 10^-3 fractionally,
C while never stopping if CHISQ has just increased. Repeat
C until coding structure used.
C
C ITER = 0
C PRINT *, 'OPTIMIZING'
C 17 FORMAT(3X,A10,3X,A10,3X,A10,3X,A10,3X,A10)
C WRITE(6,7) (PNAME(I),I=1,MA)
200 PCHISQ = CHISQ
   ITER = ITER + 1
   CALL mrqmin(x,y,signdata,a,ma,covar,alpha,nc,chiq,
      plur,alamda)
   CALL mrqmin(x,y,signdata,a,ma,covar,alpha,nc,chiq,
      plur,alamda)
C 1004 FORMAT (3X,'Covar-iterating', D15.6, D15.6)
C PRINT 1004, COVAR(1,1), COVAR(1,2)
C PRINT 1004, COVAR(2,1), COVAR(2,2)
C 1007 FORMAT(3X,'iterating A(F)', D15.6, D15.6, D15.6)
1008 FORMAT('X','iterating A(i) :',D15.6,D15.6)
PRINT 1007, A(1), A(2), A(3)
PRINT 1008, A(4), A(5)
IF(CHISQ.GT.PCHISQ) OR.
* (ABS(CHISQ-PCHISQ)/CHISQ.GT.0.0001) GO TO 200
C C Setting ALAMDA = 0 to call back the COVAR and ALPHA matrices
C from MROMIN
C PRINT *, 'EXTRACTING THE 2ND DERIV. and COVAR MATRICES'
C ALAMDA = 0,
C CALL murofin, ysigndata, ylf, ma, covar, alpha, nca, chisq,
* plfunc, alamda)
C DO 210, I=1,MA
C WRITE(6,37) (ALPHA(I,J)=1,MA)
C 210 CONTINUE
C DO 220, I=1,MA
C WRITE(6,37) (COVAR(I,J)=1,MA)
C 220 CONTINUE
C C Placing the estimated parameters and objective function information
C into an iterations log file
C OPEN(UNIT=1,FILE='ITRLOG.DAT',
* STATUS='OLD',FORM='FORMATTED')
C WRITE(1,137) (A(i),J=1,MA)
999 FORMAT (C10.5)
CLOSE(UNIT=1)
C C Calling PFUNC to calculate the estimated values based on the now
C optimized' parameter values stored in array A.
C PRINT *, 'CALCULATING THE OPTIMIZED MODEL PREDICTIONS'
C DO 300, I = 1, NDATA, I
C CALL PFUNC(X(I),A,YPRED(I),DYDA,MA)
C DO 300 J=1,MA
C PRINT *, 'Mainstep DYDA = ', DYDA
C 301 CONTINUE
C C sample call funcA(x(i),ymod,dyda,ma)
C 300 CONTINUE
C C PRINTING RESULTS of the fit into text files
C PRINT *, 'PLACING RESULTS IN OUTPUT FILES'
C OPEN(UNIT=1,FILE='RESULTS.DAT',
* STATUS='OLD',FORM='FORMATTED')
C OPEN(UNIT=12,FILE='GRAPHDAT',
* STATUS='OLD',FORM='FORMATTED')
C WRITE(1,20300)
20300 FORMAT(20X,'RESULTS' /, 19X,8('..',/)
WRITE(1,20400)
20400 FORMAT(5X,'POINT',5X,'Y EXPERIMENTAL',5X,'Y CALCULATED',5X,
* 'DIFFERENCE',5X,'RELATIVE ERR.')
C C Calculating the variance and std. deviation of the final fit
C DESV=0.
C DO 60 I=1,NDATA
C IF(Y(I)EQ.YPRED(I))
C IF(Y(I)-YPRED(I)).GT.1.D-16
C ERF=(DIF(Y(I))*100.)
121 IF(DABS(ERF).GT.9999) ERR=9999
C WRITE(1,20500) (Y(I),YPRED(I),DIF,ERR)
C WRITE(12,12) (Y(I),YPRED(I))
C 2050 FORMAT(7X,13,4X,D15.6,1X,D15.6,D15.6,8X,9.3,,'%')
C DESV=DESV+DIF*2
60 CONTINUE
C C (Calculation of the standard deviation and variance)
C C C ACOLE DE LA DESVIACION ESTANDAR Y VARIANZA
C C VAR=DESV/(NDATA-MA)
C IF(VAR.LT.0.0) VAR=VAR
C DESV=SQRT(VAR)
C WRITE(1,10001) DESV
C C WRITE(1,10001) VAR
10000 FORMAT(5X,'STD. ERROR OF EST. , VAR=% ,D15.6)
10001 FORMAT(5X,'VARIANCE = ',D15.6)
C C WRITE(1,20600)
C C 20600 FORMAT(20X,'FITTED PARAMETERS',/19X,19('..',/)
C DO 70 I=1,MA
C WRITE(1,20700) PNAME(I), A(I)
2070 FORMAT(5X,AK, ' = ',D25.12)
70 CONTINUE
C C WRITE(1,15000)
C C 5000 FORMAT(20X,'INITIAL GUESS',/19X,15('..',/)
C DO 5001 I=1,MA
C WRITE(1,20800) PNAME(I), BEST(I)
5001 CONTINUE
C WRITE(1,2081)
2081 FORMAT(5X,1X,8('..',/)
C C CLOSING UNIT=13
C CLOSING UNIT=13)
C C RETURN
END
C C C C
SUBROUTINE mrqmin(x,y,signdata,ia,ma,covar,alpha,nc,chiqs,func,alama)
  C
  C Levenberg-Marquardt method, attempting to reduce the value chi-
  C squared of a fit between a set of NDATA points X(I),Y(I) and a nonlinear
  C function dependent on MA coefficients A(1:MA). The array IA(1:MA)
  C indicates by non-zero entries those components of A that should be
  C fitted for, and by zero entries those components that should be held
  C fixed at their input values. The program returns current best-fit
  C values for the parameters A(1:MA), and chiqs=q=CHISQ. The
  C arrays COVER(1:NCA,1:NCA), ALPHA(1:NCA,1:NCA) with physical dimension
  C NCA => the number of fitted parameters are used as working space
  C during most iterations. ALAMDA initially starts at 0.001. If a step
  C succeeds, CHISQ becomes smaller, and ALAMDA decreases by a factor of
  C 10. If a step fails, ALAMDA grows by a factor of 10. This subroutine
  C is to be called repeatedly until convergence is achieved. Then, one
  C final call is made with ALAMDA=0, so that COVAR(1:MA,1:MA) returns
  C the covariance matrix, and ALPHA(1:MA,1:MA) the curvature matrix.
  C (Parameters held fixed will return zero covariances).
  C
  C HIROSHI'S NOTE: Since this work doesn't have for each value of Y(I)
  C a corresponding X(I) but rather a temperature-time profile. X(I)
  C will be used to hold the name of the file containing the
  C temperature-time profile to be used by LSODE.F.
  C
  C INTEGER ma,nc,ndata,ia(i),A(MAX)
  C DOUBLE PRECISION alama,chiqs,func,alama(alpha,nc,ala)
  C * covar(nc,nc),signdata,y(nc,nc)
  C CHARACTER*10 x(nc,nc)
  C PARAMETER (MAXX=20)
  C USES covar,gauss,mrqcof
  C INTEGER j,k,k1,mfit
  C DOUBLE PRECISION ochiqs,atry(MMAX),beta(MMAX),da(MMAX)
  C SAVE ochiqs,atry,beta,da,mfit
  C PRINT *, 'CALL MRQMIN'
  C 1009 FORMAT(3X,'mrqmin-top A(I)',D15.6,D15.6,D15.6)
  C 1010 FORMAT(3X,'mrqmin-top A(I)',D15.6,D15.6)
  C PRINT 1009, A(1), A(2), A(3)
  C PRINT 1010, A(4), A(5)
  C 1001 FORMAT (3X,'Covar-MRQMIN',D15.6,D15.6)
  C PRINT 1001, COVAR(1,1), COVAR(1,2)
  C PRINT 1001, COVAR(2,1), COVAR(2,2)
  C IF (alamda.lt.0) THEN
  C mfit=0
  C DO 11 j=1,ma
  C IF (ia(j).ne.0) mfit= mfit+1
  C 11 CONTINUE
  C alama=0.001
  C CALL MRQCOF(x,y,signdata,ia,ma,alpha,nc,chiqs,func)
  C 1003 FORMAT(3X,'mrqmin-postmrqcof A(I)',D15.6,D15.6,D15.6)
  C 1004 FORMAT(3X,'mrqmin-postmrqcof A(I)',D15.6,D15.6)
  C PRINT 1003, A(1), A(2), A(3)
  C PRINT 1004, A(4), A(5)
  C PRINT 1004, A(2), A(3)
  C 13 CONTINUE
  C ochiqs=chiqs
  C DO 12 j=1,ma
  C atry(j)=atry(j)
  C 12 CONTINUE
  C END

C END
*funcx *)
C C Used by MRQMIN to evaluate the linearized fitting matrix ALFA.
C and the vector BETA, and calculate CHISQ.
C
INTEGER ma, nalp, ndata, ist(la), MMAX
DOUBLE PRECISION chiq, a(ma), alpha(nalp, nalp), beta(ma),
* sig(ndata), y(ndata)
CHARACTER*10 x(ndata)
EXTERNAL funcx
PARAMETER (MMAX=20)
INTEGER mfi, k1, m
DOUBLE PRECISION dy, sig2l, wt, ymod, dyda(MMAX)
C PRINT *, 'CALL MRQCOF'
C 1001 FORMAT (3X,'Covar-MRQCOF', D15.6, D15.6)
C PRINT 1001, alpha(1,1), alpha(1,2)
C PRINT 1001, alpha(2,1), alpha(2,2)
C 1003 FORMAT (3X, 'mrqcof-tor A(1)', D15.6, D15.6, D15.6)
C 1004 FORMAT (3X, 'mrqcof-tor A(1)', D15.6, D15.6)
C PRINT 1003, A(1), A(2), A(3)
C PRINT 1004, A(4), A(5)

mfi=0
do 11 j=1, ma
  if (ist(j) ne 0) mfi=mfi+j+1
11 continue
  do 12 j=1, mfi
    do 13 k=1, j
      alpha(j,k)=0.
    12 continue
    beta(j)=0.

13 continue
chiq=0.
  do 16 i=1, ndata
    call funcx(x(i), a, ymod, dyda, ma)
  16 continue
  C DO 301 k=1, MA
  C PRINT *, j, 'mrqcof DYDA(J)= ', DYDA(J)
C 301 CONTINUE
    sig2l=1./sig(i)*sig(i))
    dy=dy(i)-ymod
    if (dy.eq.0) then
      j=j+1
      wt=dyda(i)*sig2l
      10 k=1
      do 14 m=1, j
        if (ist(m) ne 0) then
          k=k+1
          alpha(j,k)=alpha(j,k)+wt*dyda(m)
        endif
      14 continue
      beta(j)=beta(j)+dy*wt
      15 continue
    endif
    else if (dpivot(j) gt 1.) then
      pause 'singular matrix in gauss - 1'
      endif
  10 continue
  endif
11 continue
  do 22 i=1, n
    ifid=0.
    do 13 k=1, n
      if (ipivot(k) ne 1) then
        do 12 k=1, n
          if (ipivot(k) eq 0) then
            if (abs(a(k,k)) ge big) then
              big=abs(a(k,k))
              icol=k
              endif
            else if (ipivot(k) eq 1) then
              pause 'singular matrix in gauss - 1'
              endif
          endif
        12 continue
        endif
      endif
    13 continue
    ipivot(icol)=ipivot(icol)+1
    if (row ne icol) then
      do 14 l=1, n
        dum=dum+row(l)
      14 continue
      endif
    endif
  22 continue
C ***********************************************************************
SUBROUTINE gauss(x, a, n, np, b, m, mp)
C C Subroutine called by MRQMIN
C linear equation solution by Gauss-Jordan elimination, A(1,N,1,N)
C is an input matrix stored in an array of physical dimensions NPNP
C b(1,n,1) is an input matrix containing the M right-hand side
C vectors, stored in an array of physical dimensions NP by MP. On
C output, A(1,N,1,N) is replaced by its matrix inverse, and
C b(1,N,1) is replaced by the corresponding set of solution
C vectors. NMAX is the largest anticipated value of n.
C
INTEGER m, mp, n, np, NMAX
DOUBLE PRECISION a(np, np), b(mp, mp)
PARAMETER (NMAX=50)
INTEGER icol, irow, j, k, l, index(NMAX), jindex(NMAX), piv(NMAX)
DOUBLE PRECISION bg, dumm, pivinv
C PRINT *, 'CALL GAUSS'
C 1001 FORMAT (3X,'Covar-gauss', D15.6, D15.6)
C PRINT 1001, a(1,1), a(1,2)
C PRINT 1001, a(2,1), a(2,2)
  do 11 j=1, n
    ipiv(j)=0.
11 continue
  do 22 i=1, n
    big=0.
    do 13 k=1, n
      if (ipiv(k) ne 1) then
        do 12 k=1, n
          if (ipivot(k) eq 0) then
            if (abs(a(k,k)) ge big) then
              big=abs(a(k,k))
              icol=k
              endif
            else if (ipivot(k) eq 1) then
              pause 'singular matrix in gauss - 1'
              endif
          endif
        12 continue
        endif
      endif
    13 continue
    ipivot(icol)=ipivot(icol)+1
    if (row ne icol) then
      do 14 l=1, n
        dum=dum+row(l)
      14 continue
      endif
    endif
  22 continue
C
C
a(row,j)=a(col,i)
a(col,i)=dum
continue
do 15 i=1,n
dum=b(row,j)
b(row,j)=b(col,i)
b(col,i)=dum
15 continue
endif
index(i)=row
index(i)=icol
C PRINT *,icol if (a(col,icol).eq.0.) pause 'singular matrix in gauss j - 2'
pivinv=1./a(col,icol)
a(col,icol)=1.
do 16 i=1,n
a(col,i)=a(col,icol)*pivinv
continue
do 17 i=1,n
b(col,i)=b(col,icol)*pivinv
17 continue
do 21 l=1,n
if(i.ne.icol) then
dum=a(l,icol)
a(l,icol)=0.
do 18 l=1,n
a(l,i)=a(l,icol)-a(col,i)*dum
continue
do 19 l=1,n
(b(l,i))=(b(l,icol))-b(col,i)*dum
19 continue
endif
21 continue
do 24 k=1,n
if(index(l).ne.index(l)) then
do 23 k=1,n
dum=a(k,index(l))
a(k,index(l))=a(k,index(l))+a(index(l),index(l))
a(k,index(l))=dum
23 continue
24 continue
return
END

DOUBLE PRECISION covar(np,mp)
INTEGER i,j,k
DOUBLE PRECISION swap
C PRINT *, 'CALL COVSR'
C 1001 FORMAT (3X,'Covar(covar)', D15.6, D15.6)
C PRINT 1001, COVAR(1,1), COVAR(1,2)
C PRINT 1001, COVAR(2,1), COVAR(2,2)
do 12 i=1,ma
do 11 j=1,i
covar(i,j)=covar(j,i)
11 continue
12 continue
k=min(k, i)
do 15 j=ma,1,-1
if(i.e.j) then
do 13 i=1,ma
swap=covar(k)
covar(k)=covar(i,j)
covar(i,j)=swap
13 continue
do 14 i=1,ma
swap=covar(k)
covar(k)=covar(j,i)
covar(j,i)=swap
14 continue
k=k-1
endif
15 continue
return
END

SUBROUTINE PFUNC (FILEIN,A,YMODEL,dyda,ma)
EXTERNAL PLSODE
INTEGER MA, LJ
DOUBLE PRECISION A(MA), APRIME(10), DYDA(MA), YMODEL
CHARACTER*10 FILEIN
C PRINT *, 'CALL PFUNC'
C 1001 FORMAT (X, 'func A(l)', D15.6, D15.6, D15.6)
C 1004 FORMAT (X, 'func A(l)', D15.6, D15.6)
C PRINT 1004, A(1), A(2), A(3)
C PRINT 1004, A(4), A(5)
C C Evaluating the model prediction
YMODEL = PLSODEFILE(IN,MA)
C C Evaluating the derivatives for each variable numerically
DO 100 J=1,MA
100 CONTINUE
101 CONTINUE
IF (I .NE. 1) THEN
   APRIME(I) = A(I)
ELSE
   APRIME(I) = 1.0001*A(I)
END IF
110 CONTINUE
DYDA(I) = (PLSODE(FILEIN,APRIME,MA)-YMODEL)/(0.0001*A(I))
100 CONTINUE
DO 301, I=1,MA
C PRINT *, 'plane DYDA(',I,')=',DYDA(I)
C 301 CONTINUE
RETURN
END

DOUBLE PRECISION FUNCTION PLSODE(FILEIN,MA)
C
C This program uses LSODE.F (the Livermore Solver for ODE's) to solve
C for P(i) over a given temperature-time profile for the reaction
C set:
C
C  \ A -------> B
C  \ \ kp
C  \ -----> C
C
C The user will input a file name containing the temperature (Celsius)-
C time (sec) profile (used for MIPR calculations) to calculate the
C quantity P produced, and compare this number to that obtained by
C the analytical expression calculated by MAPLE V, coded in program
C PANA.L.F over the same temperature-time profile. Values of kpo,
C kbo, Ep, Eb, and Ao will be supplied by the user.
C
C The order in which variables must be input for fitting are:
C  Ao, kbo, Eb, kpo, Ep
C  in this precise order.
C
C The ODE's will be solved on the interval from t = 0.0 sec to the time
C at the end of the temperature-time profile (or t = TINT*IMAX).
C with the initial conditions A(0)=Ao, B(0)=0, and P(0)=0. The
C problems is assumed to be non-stiff (a guess since I cannot
C supply the jacobian).
C
C*****************************************************************************
C
C Defining variables
C KBO, KPO = Arrhenius preexponential factors for reaction rate
C coefficients kp and kbo; units = calories
C AO = initial concentration of reactant A
C P = amount of product P formed
C TIME = array of time values to be stored from the
C temperature-time profiles
C TEMP = array of temperature values to be stored from the
C in the temperature-time profiles
C I,J = counters to be used in integration steps
C IMAX = number of points in the data file
C TINT = time interval between data points in input file,
C set to 0.01 seconds (10 msec)
C FILEIN = variable to hold file name containing the temperature-
C time profile, with time intervals of 10 msec

C Declaring variables being passed in
INTEGER MA
DOUBLE PRECISION A(MA)

C Declaring as external (as required by LSODE.F) subroutines FUNC
C (contains system of ODE's) and JDUMMY (dummy subroutine name)
C for the jacobian I'm not using)
* Ao, TINT
INTEGER I, J, IMAX
CHARACTER*10 FILEIN

C Declaring variables for LSODE subroutine
EXTERNAL FUNC
DOUBLE PRECISION ATOL, RTOL, RWORK, T, Tout, Y
INTEGER ITOL, ITASK, ISTATE, ILOPT, LW, LIW, MF
DIMENSION Y(1), ATOL(1), RWORK(68), IW(20)

C Declaring common parameters I input which will be passed from
C this main program to the subroutines
COMMON /PARAMTR/ TEMP, IMAX, KBO, KPO, EP, EB, AO

C Assigning the parameter values passed by array X, in the order
C Ep [=] cal/gmole
C Eb [=] cal/gmole
C Ao [=] arbitrary
C kpo [=] 1/sec
C kbo [=] 1/sec

C PRINT *, 'CALL PLSODE'
C PRINT *, FILEIN
EP = A(1)
EB = A(2)
AO = A(3)
KPO = A(4)
KBO = A(5)
C Initializing variables
C
I = 1
IMAX = 2000
J = 1
TINT = 0.01
C Initializing the array to hold temp and time data
DO 100 J = 1, 2000, 1
   TIME(J) = 0.0
   TEMP(J) = 0.0
100 CONTINUE

C LOADING THE TEMP. TIME PROFILE from the specified data file
C and reading in Time (sec) & Temperature (Celsius) respectively
C
IMAX = 0
OPEN(UNIT=1, FILE='FILEIN, STATUS='OLD')
DO 1010 J = 1, 2000
   READ(*, END=2000) TIME(J), TEMP(J)
   TEMP(J) = TEMP(J) + 273.15
   IMAX = IMAX + 1
1010 CONTINUE

C Preparing to call LSODE to solve system of ODE's for Pt
C
C no. of ODE's = 3
NFO=3
C use initial conditions for initial values of A, P, and B.
Y(1) = A0
Y(2) = 0.0
Y(3) = 0.0
C integrate from time = 0 sec
T = 0.0
C initially integrate to first interval, TINT
TOUT = TINT
C tolerances stored in an array
ITOL = 2
C relative tolerance parameter
RTOL = 1.0
C absolute tolerance parameters for A, P, and B
ATOL(1) = 1.0
ATOL(2) = 1.0
ATOL(3) = 1.0
C computes for trend of profile
ITASK = 1
C required by LSODE
ISTATE = 1
C no optional inputs used
IIJNT = 0
C required for mf = 10
LRW = 68
C required for mf = 10
LWR = 20

C for nonstiff (adams) method, no jacobian used
MF = 10
C
C Using LSODE to solve for Pt
C
DO 1000 J = 2, IMAX
   CALL LSODE (FUNC, NEQ, Y, TOUT, ITOL, RTOL, ATOL, ITASK,
       * ISTATE, J, IJNT, TWORK, LRW, IWORK, LIW, J, IDUMMY, MF)
   99 FORMAT (1X, F5.2, 4X, FR 4.2, 3X, FR 4.2, 3X, FR 4.2)
   C WRITE(*, 99) TOUT, Y(1), Y(2), Y(3)
C
C Stops program if error in LSODE calculation detected.
C
IF (ISTATE .LE. 0) THEN
   WRITE(*, 90) ISTATE
   90 FORMAT (1X, 'ERROR HAT,, ISTA = ', 1X, I3)
   STOP
END IF
C
1000 CONTINUE

C Assigning the LSODE calculated value to PLSDODE
C
PLSDODE = Y(2)
C
RETURN
C
END
c standard, with minor modifications.
c
reference.
c alan c. hindmarsh, odepack, a systematized collection of ode
c solvers, in scientific computing, t. s. steplenman et al., eds.,
c north-holland, amsterdam, 1983, pp. 55-64.
c
c author and contact. alan c. hindmarsh,
c computing and mathematics research div., 1-316
c lawrence livermore national laboratory

livermore, ca 94550.
c
summary of usage.
c
c communication between the user and the tcode package, for normal
c situations, is summarized here. this summary describes only a subset
c of the full set of options available. see the full description for
c details, including optional communication, nonstandard options,
c and instructions for special situations. see also the example
c problem (with program and output) following this summary.
c
c a. first provide a subroutine of the form,
c subroutine f (neq, t, y, ydot)
c dimension y(neq), ydot(neq)
c which supplies the vector function f by loading ydot(i) with f(i)
c
b. next determine (or guess) whether or not the problem is stiff.
c stiffness occurs when the jacobi matrix df/dy has an eigenvector
whose real part is negative and large in magnitude, compared to the
c reciprocal of the span of interest. if the problem is stiff,
c use a method flag mf = 10. if it is stiff, there are four standard
choices for mf, and tcode requires the jacobi matrix in some form.
c this matrix is regarded either as full (mf = 21 or 22),
c or banded (mf = 24 or 25). in the banded case, tcode requires two
half-bandwidth parameters ml and mu. these are, respectively, the
widths of the lower and upper parts of the band, excluding the main
c diagonal, thus the band consists of the locations (i,j) with
1+ml le j le j+mu, and the full bandwidth is ml+mu+1.
c
c c. if the problem is stiff, you are encouraged to supply the jacobi

c directly (mf = 21 or 24), but if this is not feasible, tcode will
c compute it internally by difference quotients (mf = 22 or 25).
c if you are supplying the jacobi, provide a subroutine of the form,
c subroutine jac (neq, t, y, ml, mu, pd, rowpd)
c dimension y(neq), pd(neq), rowpd(neq)
c which supplies pd(i) by loading pd(i) as follows.
c
c d. write a main program which calls subroutine tcode once for
each point at which answers are desired. this should also provide
c for possible use of logical unit 6 for output of error messages
c by tcode. on the first call to tcode, supply arguments as follows...
c f = name of subroutine for right-hand side vector f.
c this name must be declared external in calling program.
c neq = number of first order ode's.
c y = array of initial values, of length neq.
c t = the initial value of the independent variable.
c tout = first point where output is desired (neq = 1).
c retol = 1 or 2 according as retol (below) is a scalar or array.
c retol = relative tolerance parameter (scalar).
c atol = absolute tolerance parameter (scalar or array).
c teh = the estimated local error in y(i) will be controlled so as
to be roughly on (in magnitude) than
c
ew(i) = retol*abs(y(i)) + atol if retol = 1, or
c
c new(i) = retol*abs(y(i)) + atol(i) if retol = 2.
c
c c. thus the local error test passes if, in each component,
c e.
the absolute error is less than atol (or atol(i)),
c or the relative error is less than retol.
c use retol = 0.0 for pure absolute error control, and
c use atol = 0.0 (or atol(i) = 0.0) for pure relative error
c control. caution: actual (global) errors may exceed these
local tolerances, so choose them conservatively.
c
set = 1 for normal computation of output values of y at t = tout.
c state = integer flag (input and output). set state = 1.
c inpt = 0 to indicate no optional inputs used.
c rwork = real work array of length at least,
c
20 + 16*neq for mf = 10,
c 22 + 9*neq + neq*2 for mf = 21 or 22,
c 22 + 10*neq + (2*ml + mu)*neq for mf = 24 or 25.
c
lwr = declared length of work (input parameter).
c iwork = integer work array of length at least,
c
20 for mf = 10,
c 20 + 9*neq for mf = 21, 22, 24, or 25.
c
if mf = 24 or 25, input in iwork(1),iwork(2) the lower
and upper half-bandwidths ml,mu.
c
lwr = declared length of work in user-s dimension.
c
jac = name of subroutine for jacobi matrix (mf = 21 or 24).
c
if used, this name must be declared external in calling
program. if not used, pass a dummy name.
c
mf = method flag, standard values are...
c
10 for nonstiff (adams) method, no jacobi used,
c 21 for stiff (bd) method, user-supplied full jacobi,
c 22 for stiff method, internally generated full jacobi,
c 24 for stiff method, user-supplied banded jacobi,
c 25 for stiff method, internally generated banded jacobi.
c note that the main program must declare arrays y, rwork, iwork,
c and possibly atol.
c
c e. the output from the first call (or any call) is,
c
c y = array of computed values of y(i) vector.
c t = corresponding value of independent variable (normally tout).
c state = 2 if tcode was successful, negative otherwise.
-1 means excess work done on this call (perhaps wrong mf).
-2 means excess accuracy requested (tolerances too small).
-3 means illegal input detected (see printed message).
-4 means repeated error test failures (check all inputs).
-5 means repeated convergence failures (perhaps bad jacobian supplied or wrong choice of mf or tolerances).
-6 means error weight became zero during problem. (solution component i vanished, and atol or atol(i) = 0.)

c f. to continue the integration after a successful return, simply
reset tout and call hode again. no other parameters need be reset.
c
------
c
example problem.
c
the following is a simple example problem, with the coding
needed for its solution by hode. See problem is from chemical
kinetics, and consists of the following three rate equations.
c
dy1dt = 0.4*y1 + 1.4*y2*y3
dy2dt = 0.4*y1 - 1.4*y2*y3 - 3.3*y2**2
dy3dt = 3.3*y2**2
c
on the interval from t = 0.0 to t = 4.0, with initial conditions
y1 = 10.0, y2 = y3 = 0. the problem is stiff.
c
the following coding solves this problem with hode, using mf = 21
c and printing results at t = 4.0, 8.0, ..., 40.0. It uses
citol = 2 and atol much smaller for y2 than for y1 or y3 because
c y2 has much smaller values.
c at the end of the run, statistical quantities of interest are
c printed (see optional outputs in the full description below).
c
e external fty, jex

e double precision atol, rtol, rwork, t, tout, y
e dimension y(3), atol(3), rwork(58), iwork(23)
e
neq = 3
ey(1) = 1.00
ney(2) = 0.00
ney(3) = 0.00
t = 0.00
tout = 4.00
titol = 2
rtol = 1e-4
atol(1) = 1e-6
atol(2) = 1e-10
atol(3) = 1e-6
itask = 1
istate = 1
iprt = 0
lrw = 58
liw = 23
mf = 21
c do 40 iomt = 1,12
c call hode(fty, neq, y, tout, itol, rtol, atol, itask, istate,
c 1 iprt, rwork, iwork, liw, jex, mf)
c
write(6,20)t,y(1),y(2),y(3)
c 20 format(7ह w = 1.d12, 6 unfortunately
if (istate 0) go to 80
c 40 tout = tout+1.0d0
c write(6,600)tout(11),iwork(12),iwork(13)
c 60 format(12h no. steps = i4,11h no. f s = i4,11h no. j s = i4)
c stop
c 80 write(6,900)istate
c 90 format(11h error halt. istate = i3)
c end

c subroutine fty (neq, t, y, ydot)
c double precision t, y, ydot
c dimension y(3), ydot(3)
c ydot(1) = 0.4*y1*y(1) + 0.8*y2*y(3)
c ydot(2) = 3.07*y(2)**2
ydot(3) = -ydot(2) - ydot(3)
c return
c end

c subroutine jex (neq, t, y, ml, mu, pd, npd)
c double precision ml, mu
c dimension y(3), pd(npd,3)
c pd(1,1) = 0.040
pd(1,2) = 0.040
pd(1,3) = 0.040
pd(2,1) = 0.040
pd(2,2) = 0.040
pd(2,3) = 0.040
pd(3,2) = 0.040
pd(3,3) = 0.040
return
c end

c the output of this program (on a cdc-7600 in single precision)
c is as follows.
c
at = 4.0000e-01 y = 9.851726e-01 3.386406e-05 1.4793576e-02
c at = 4.0000e-00 y = 9.851425e-01 2.240143e-05 9.462543e-02
c at = 4.0000e+00 y = 7.58055e-01 9.184641e-06 2.841585e-01
at = 4.0000e+02 y = 4.504846e-01 3.222434e-06 5.495122e-01
at = 4.0000e+03 y = 1.937701e-01 8.940759e-07 8.16292e-01
at = 4.0000e+04 y = 3.89971e-02 1.62119e-07 9.61027e-01
at = 4.0000e+05 y = 9.3521e-03 1.98377e-08 9.95089e-01
at = 4.0000e+06 y = 5.15296e-04 2.064735e-09 9.99481e-01
at = 4.0000e+07 y = 3.0141e-05 2.122673e-10 9.99946e-01
at = 4.0000e+08 y = 5.493529e-06 2.19782e-11 9.99994e-01
at = 4.0000e+09 y = 5.129458e-07 2.031784e-12 9.99999e-01
at = 4.0000e+10 y = 7.170588e-08 2.868234e-13 1.0000000e00
c
no steps = 330 no f s = 405 no j s = 69

------
c full description of user interface to hode.
c
the user interface to node consists of the following parts.

c i. the call sequence to subroutine node, which is a driver
   routine for the solver. this includes descriptions of both
   the call sequence arguments and of user-supplied routines.
   following these descriptions is a description of
   optional inputs available through the call sequence, and then
   a description of optional outputs (in the work arrays).

c ii. descriptions of other routines in the node package that may be
   (optionally) called by the user. these provide the ability to
   alter error message handling, save and restore the internal
   common, and obtain specified derivatives of the solution y(i).

c iii. descriptions of common blocks to be declared in overlay
   or similar environments, or to be saved when doing an interrupt
   of the problem and continued solution later.

c iv. description of two routines in the node package, either of
   which the user may replace with his own version, if desired
   these relate to the measurement of errors.

c part i. call sequence.

c the call sequence parameters used for input only are
   c f, neq, tout, tol, rtol, atol, h, hmax, leps, hms, c
   c and those used for both input and output are
   c y, l, state.

c the work arrays rwork and iwork are also used for conditional and
   optional inputs and optional outputs. (the term output here refers
   to the return from subroutine node to the user's calling program.)

c the legality of input parameters will be thoroughly checked on the
   initial call for the problem, but not checked thereafter unless a
   change in input parameters is flagged by state = 3 on input.

c the descriptions of the call arguments are as follows.

c f = the name of the user-supplied subroutine defining the
   c ode system. the system must be put in the first-order
   c form dy/dt = f(t,y), where f is a vector-valued function
   c of the scalar t and the vector y. subroutine f is to
   c compute the function f. it is to have the form
   c subroutine f(neq, t, y, ydot)
   c dimension y(1), ydot(1)
   c where neq, t, and y are input, and the array ydot = f(t,y)
   c is output. y and ydot are arrays of length neq.
   c (in the dimension statement above, 1 is a dummy
   c dimension. it can be replaced by any value.)
   c subroutine f should not alter y(1),...,y(neq)
   c f must be declared external in the calling program
   c subroutine f may access user-defined quantities in

neq(2),... and/or in y(neq(1)+1),... if neq is an array
(c dimensioned in f) and/or y has length exceeding neq(1).
see the descriptions of neq and y below

if quantities computed in the f routine are needed
externally to node, an extra call to f should be made
for this purpose, for consistent and accurate results.
if only the derivative dy/dt is needed, use intdy instead.

c neq = the size of the ode system (number of first order
ordinary differential equations) used only for input.
neq may be decreased, but not increased, during the problem.
if neq is decreased (with istate = 3 on input), the
remaining components of y should be left undisturbed. if
these are to be accessed in f and/or jac.

normally, neq is a scalar, and it is generally referred to
as a scalar in this user interface description. however,
neq may be an array, with neq(1) set to the system size.
(the node package accesses only neq(1)) in either case,
this parameter is passed as the neq argument in all calls
to f and jac (hence, if it is an array, locations
eq(2),... may be used to store other integer data and pass
it to f and/or jac. subtorences f and/or jac must include
neq in a dimension statement in that case.

y = a real array for the vector of dependent variables, of
length neq or more; used for both input and output on the
first call (istate = 1), and only for output on other calls.
on the first call, y must contain the vector of initial
values. on output, y contains the computed solution vector,
evaluated at t. if desired, the y array may be used
for other purposes between calls to the solver.

c this array is passed as the y argument in all calls to
f and jac, hence its length may exceed neq, and locations
y(neq+1),... may be used to store other real data and
pass it to f and/or jac. (the node package accesses only
y(1),...,y(neq).

c t = the independent variable; input, t is used only on the
first call, at the initial point of the integration.

on output, after each call, t is the value at which a
computed solution y is to begin (usually the same as tout). on
an error return, t is the furthest point reached.

c tout = the next value of t at which a computed solution is desired
used only for input.

when starting the problem (istate = 1), tout may be equal
to t for one call, then should not t for the next call
for the initial t, an input value of tout not t is used
in order to determine the direction of the integration
(c the absolute sign of the step size) and the rough
scale of the problem. Integration in either direction
towards or backward in t) is permitted.

if itask = 2 or 5 (one-step modes), tout is ignored after
the first call (i.e. the first call with tout < 0).
otherwise, tout is required on every call.

if itask = 1, 3, or 4, the values of tout need not be
monotone, but a value of tout which falls up is limited
to the current internal t interval, whose endpoints are
tcr - hu and tcr + hu (see optional outputs, below, for
tcr and hu).

int = an indicator for the type of error control. see
description below under atol. used only for input.

rtol = a relative error tolerance parameter, either a scalar or
an array of length neq. see description below under atol.
input only.

atol = an absolute error tolerance parameter, either a scalar or
an array of length neq. input only.

the input parameters int, rtol, and atol determine
the error control performed by the solver. the solver will
control the vector \( e = (e(t)) \) of estimated local errors
in \( y \), according to an inequality of the form

\[
\text{rms-norm of } (e(t)) \leq \text{int},
\]

where \( \text{rtol} = \text{int}(\gamma)^{\text{int}} + \text{atol}(\gamma) \),
and the rms-norm (root-mean-square norm) here is

\[
\text{rms-norm}(y) = \sqrt{\sum y(i)^{2} / \text{neg}}, \quad \text{here } E = \text{rtol}(\gamma)
\]
is a vector of weights which must always be positive, and
the values of rtol and atol should be non-negative.

the following table gives the types (scalar/array)
of rtol and atol, and the corresponding form of rtol(i).

<table>
<thead>
<tr>
<th>rtol</th>
<th>atol</th>
<th>rtol(i)</th>
<th>atol(i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>scalar</td>
<td>scalar</td>
<td>rtol(i)^2 + atol(i)</td>
<td>rtol(i)^2 + atol(i)</td>
</tr>
<tr>
<td>array</td>
<td>array</td>
<td>rtol(i)^2 + atol(i)</td>
<td>rtol(i)^2 + atol(i)</td>
</tr>
</tbody>
</table>

when either of these parameters is a scalar, it need not
be dimensioned in the user's calling program.

if none of the above choices (with int, rtol, and atol
fixed throughout the problem) is suitable, some general
error controls can be obtained by substituting
user-supplied routines for the setting of rtol and/or for
the norm calculation. see part iv below.

if global errors are to be estimated by making a repeated
treatment of the same problem with smaller tolerances, then all
components of rtol and atol (i.e. of rtol) should be scaled
down uniformly.

itask = an index specifying the task to be performed.
input only. itask has the following values and meanings:
1 means normal computation of output values of \( y(t) \) at
\( t = \text{tout} \) (by overshooting and interpolating).
2 means take one step only and return.
3 means stop at the first internal : : : t point at or
beyond \( \text{t} = \text{tout} \) and return.
4 means normal computation of output values of \( y(t) \) at
\( t = \text{tout} \) but without overshooting \( t = \text{tcrit} \).
5 means the solution must be wronog(1) = 1 (exactly) to
be returned. if int = 4 or 5 and the solver reaches \( \text{tcrit} \)
(within roundoff), it will return \( t = \text{tcrit} \) (exactly) to
indicate this (unless int = 4 and tout comes before \( \text{tcrit} \),
in which case answers at \( t = \text{tout} \) are returned first).

istate = an index used for input and output to specify the
state of the calculation.

on input, the values of istate are as follows:
1 means this is the first call for the problem
(initializations will be done). see note below.
2 means this is not the first call, and the calculation
is to continue normally, with no change in any input
parameters except possibly tout and itask.
3 means this is not the first call, and the
calculation is to continue normally, but with
a change in input parameters other than
tout and itask. changes are allowed in
neg, itol, rtol, atol, inpt, lwp, iw, ml, mn,
and any of the optional inputs except h0
(see iwork description for ml and mn).

note: a preliminary call with tout = 1 is not counted
as a first call here, as no initialization or checking of
input is done. such a call is sometimes useful for the
purpose of outputting the initial conditions.
thus the first call for which tout neq 1 requires
istate = 1 on input.
on output, istate has the following values and meanings:
1 means nothing was done, as tout was equal to 1 with
istate = 1 on input. (however, an internal counter was
set to detect and prevent repeated calls of this type)
2 means the integration was performed successfully.
-1 means an excessive amount of work (more than maxstep
steps) was done on this call, before completing the
request task, but the integration was otherwise
successful as far as t. (maxstep is an optional input
and is normally 500) to continue, the user may
simply reset istate to a value gr. 1 and call again
(the excess work step counter will be reset to 0).
In addition, the user may increase maxstep to avoid
this error return (see below on optional inputs).
-2 means too much accuracy was requested for the precision
of the machine being used. this was detected before
completing the requested task, but the integration
was successful as far as t. to continue, the tolerance
parameters must be reset, and istate must be set
to 3. the optional output told may be used for this
purpose. (note, if this condition is detected before
taking any steps, then an illegal input return
(istate = -3) occurs instead.)
-3 means illegal input was detected, before taking any
integration steps. see written message for details.
(note, if the solver detects an infinite loop of calls
to the solver with illegal input, it will cause
the run to stop.
-4 means there were repeated error test failures on
one attempted step, before completing the requested
task, but the integration was successful as far as t.
the problem may have a singularity, or the input
may be inappropriate.
-5 means there were repeated convergence test failures on
one attempted step, before completing the requested
task, but the integration was successful as far as t.
this may be caused by an inaccurate jacobian matrix,
if one is being used.
-6 means ewt(t) became zero for some i during the
integration, pure relative error control (asol(1)=0)
was requested on a variable which has now vanished.
the integration was successful as far as t.

Note, since the normal output value of istate is 2,
it does not need to be reset for normal continuation.
also, since a negative input value of istate will be
regarded as illegal, a negative output value requires the
user to change it, and possibly other inputs, before
calling the solver again.

c iopt = an integer flag to specify whether or not any optional
c inputs are being used on this call. input only.
c the optional inputs are listed separately below.
c iopt = 0 means no optional inputs are being used.
c default values will be used in all cases.
c iopt = 1 means one or more optional inputs are being used.
c
c rwork = a real working array (double precision)
c the length of rwork must be at least
20 + nhb*(maxord + 1) + 3*neq + lwm
where
nhb = the initial value of neq.
maxord = 12 (if meth = 1) or 5 (if meth = 2) (unless a
smaller value is given as an optional input),
lwm = 0 if miter = 0,
lwm = neq**2 + 2 if miter is 1 or 2,
lwm = neq + 2 if miter = 3, and
lwm = (2*nhb+mu+1)*neq + 2 if miter is 4 or 5,
see the mf description for miter and miter.
thus if maxord has its default value and neq is constant,
this length is
20 + 16*neq for mf = 10,
22 + 16*neq + neq**2 for mf = 11 or 12,
22 + 17*neq for mf = 13,
22 + 17*neq + (2*nhb+mu)*neq for mf = 14 or 15,
20 + 1*neq for mf = 20,
22 + 9*neq + neq**2 for mf = 21 or 22,
22 + 10*neq for mf = 23,
22 + 10*neq + (2*nhb+mu)*neq for mf = 24 or 25.
the first 20 words of rwork are reserved for conditional
and optional inputs and optional outputs.
c
the following word in rwork is a conditional input
rwork(1) = crit = critical value of f which the solver
is not to overshoot required if imsk is
4 or 5, and ignored otherwise. (see imsk.)
c
lwm = the length of the array rwork, as declared by the user.
c (this will be checked by the solver.)
c
iwork = an integer work array, the length of iwork must be at least
20 if miter = 0 or 3 (mf = 10, 13, 20, 23), or
20 + neq otherwise (mf = 11, 12, 14, 15, 21, 22, 24, 25).
the first few words of iwork are used for conditional and
optional inputs and optional outputs.
c
the following 2 words in iwork are conditional inputs.
iwork(1) = ml these are the lower and upper
iwork(2) = mu half-bandwidths, respectively of the
handed jacobian, excluding the main diagonal.
c the hand is defined by the matrix locations
t(i,j) with i,mu le j le i+mu+1, ml and mu
c must satisfy 0 le ml, mu le neq-1.
c these are required if miter is 4 or 5, and
ignored otherwise. ml and mu may in fact be
the hand parameters for a matrix to which
c diddy is only approximately equal.
c
c lwm = the length of the array iwork, as declared by the user.
c (this will be checked by the solver.)
c
Note, the work arrays must not be altered between calls to ode
for the same problem, except possibly for the conditional and
optional inputs, and except for the last *neq words of rwork.
c the latter space is used for internal scratch space, and so is
c available for use by the user outside node between calls, if
c desired (but not for use by f or jac).
c
ec jác = the name of the user-supplied routine (miter = 1 or 4) to
c compute the jacobian matrix, df/dy, as a function of
c the scalar t and the vector y. it is to have the form
ec subroutine jac (neq, t, y, ml, mu, pd, mrowpd)
c dimension y(1), pd(irowpd, 1)
ec where neq, t, y, ml, mu, and mrowpd are input and the array
ec pd is to be loaded with partial derivatives (elements of
ec the jacobian matrix) on output. pd must be given a first
c dimension of mrowpd. if and y have the same meaning as in
ec subroutine f (in the dimension statement above, 1 is a
dummy dimension... it can be replaced by any value.)
c in the full matrix case (miter = 1), ml and mu are
ec ignored, and the jacobian is to be loaded into pd in
coloumnwise manner, with df(ij)/dy(j) loaded into pd(i,j).
ec in the band matrix case (miter = 4), the elements
ec within the band are to be loaded into pd in columnwise
ec manner, with diagonal lines of df/dy loaded into the rows
ec of pd. thus df(ij)/dy(j) is to be loaded into pd(i:j+mu+1, j)
c ml and mu are the half-bandwidth parameters (see iwork)
c the locations in pd to the two triangular areas which
ec correspond to nonexistent matrix elements can be ignored
ec or loaded arbitrarily, as they are overwritten by isode.
c jác need not provide df/dy exactly. a crude
ec approximation (possibly with a smaller bandwidth) will do.
c in either case, pd is preset to zero by the solver,
c so that only the nonzero elements need be loaded by jác.
c each call to jác is preceded by a call to f with the same
ec arguments neq, t, and y, thus to gain some efficiency.
c intermediate quantities shared by both calculations may be
ec saved in a user common block by f and not recomputed by jác.
c if desired, also, jác may alter the y array, if desired.
c jác must be declared external in the calling program.
ec subroutine jac may access user-defined quantities in
ec neq(2) and/or in y(neq(1)+1),... if neq is an array
ec (dimensioned in jac) and/or has length exceeding neq(1).
ec see the descriptions of neq and y above.
c
ec mnf = the method flag. used only for input. the legal values of
c mnf are 10, 11, 12, 13, 14, 15, 20, 21, 22, 23, 24, and 25
ec mnf has decimal digits meth and miter. mnf = 10*meth + miter.
ec meth indicates the basic linear multistep method.
ec meth = 1 means the implicit Adams method.
ec meth = 2 means the method based on backward
ec differentiation formulas (bhdf).
ec miter indicates the correction iteration method.
ec miter = 0 means functional iteration (no jacobian matrix
ec is involved).
ec miter = 1 means chord iteration with a user-supplied
ec full (neq by neq) jacobian.
ec miter = 2 means chord iteration with an internally
ec generated (difference quotient) full jacobian
ec (using neq extra calls to f per df/dy value).
ec miter = 3 means chord iteration with an internally
ec generated diagonal jacobian approximation.
ec (using 1 extra call to f per df/dy evaluation).
ec miter = 4 means chord iteration with a user-supplied
ec banded jacobian.
ec miter = 5 means chord iteration with an internally
ec generated banded jacobian (using m1*mu+1 extra
ec calls to f per df/dy evaluation).
ec if miter = 1 or 4, the user must supply a subroutine jác
ec (the name is arbitrary) as described above under jac.
ec for other values of miter, a dummy argument can be used.
c
ec optional inputs.
c
ec the following is a list of the optional inputs provided for in the
c call sequence (see also part ii) for each such input variable.
c this table lists its name as used in this documentation, its
c location in the call sequence, its meaning, and the default value.
c the use of any of these inputs requires input = 1, and in that
c case all of these inputs are examined a value of zero for any
ec of these optional inputs will cause the default value to be used.
c thus to use a subset of the optional inputs, simply preload
ec locations 5 to 10 in rwork and iwork to 0 and 0 respectively, and
ec then set those of interest to nonzero values.
c
ec name location meaning and default value
ec
ec h0 rwork(5) the step size to be attempted on the first step.
ec the default value is determined by the solver.
ec
ec hmax rwork(6) the maximum absolute step size allowed.
ec the default value is infinite.
ec
ec hmin rwork(7) the minimum absolute step size allowed.
ec the default value is 0. (this lower bound is not
ec enforced on the final step before reaching exit)
ec when nst = 4 or 5.)
ec
ec mmax rwork(5) the maximum order to be allowed. the default
ec value is 12 if meth = 1, and 5 if meth = 2.
ec if mmax exceeds the default value, it will
ec be reduced to the default value.
ec if mmax is changed during the problem, it may
ec cause the current order to be reduced.
ec
ec maxstep rwork(6) maximum number of (internally defined) steps
ec allowed during one call to the solver.
ec the default value is 500.
ec
ec msteps iwork(7) maximum number of messages printed (per problem)
ec warning that t+h = c on a step (h = step size).
ec this must be positive to result in a non-default
value. The default value is 10.

**Optional output.**

As an optional additional output from IMODE, the variables listed below are quantities related to the performance of ISOLVE, which are available to the user; these are communicated by way of the work arrays, but also have internal mnemonic names as shown.

**Except where stated otherwise, all of these outputs are defined on any successful return from ISOLVE, and on any return with istatus = -1, -2, -4, -5, or -6, on an illegal input return.**

*if any,* except possibly for tolstf, lenw, and lenw, on any error return, outputs relevant to the error will be defined, as noted below.

**Name**

**Location**

**Meaning**

**hu**

rwork(11) the step size in x last used (successfully).

**hcur**

rwork(12) the step size to be attempted on the next step.

**tcur**

rwork(13) the current value of the independent variable, which the solver has actually reached, i.e., the current internal mesh point in x on output; tcur will always be at least as far as the argument t, but may be further (if interpolation was done).

**tolstf**

rwork(14) a tolerance scale factor, greater than 1.0, computed when a request for too much accuracy was detected (istatus = -3) if detected at the start of the problem, if status = -2 otherwise. If tolstf is left unaltered but toler and atol are uniformly scaled up by a factor of 10 for the next call, then the solver is deemed likely to succeed. (The user may also ignore tolstf and alter the tolerance parameters in any other way appropriate.)

**nst**

iwork(11) the number of steps taken for the problem so far.

**nfe**

iwork(12) the number of f evaluations for the problem so far.

**nje**

iwork(13) the number of Jacobian evaluations (and of associated lu decompositions) for the problem so far.

**nru**

iwork(14) the method order last used (successfully).

**nrer**

iwork(15) the order to be attempted on the next step.

**ninse**

iwork(16) the index of the component of largest magnitude in the weighted local error vector (e(i)/hjw(i)).

**lenw**

iwork(17) the length of rwork actually required.

**Lenw**

iwork(18) the length of iwork actually required.

**This is defined on normal returns and on an illegal input return for insufficient storage.**

**lenw**

iwork(18) the length of iwork actually required.

**This is defined on normal returns and on an illegal input return for insufficient storage.**

**The following two arrays are segments of the rwork array which may also be of interest to the user as optional outputs.**

**For each array, the table below gives its internal name, its base address in rwork, and its description.**

<table>
<thead>
<tr>
<th><strong>Name</strong></th>
<th><strong>Base Address</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>yh</td>
<td>21</td>
<td>The nodestck history array, of size nyh by (neq + 1), where nyh is the initial value of neq for j = 0, 1, ..., neq, column j = 1 of y contains hcur***/actorinal(j) times the j-th derivative of the interpolating polynomial currently representing the solution, evaluated at t = tcur</td>
</tr>
<tr>
<td>acor</td>
<td>lenw-neq+1</td>
<td>Array of size neq used for the accumulated corrections on each step, scaled on output to represent the estimated local error in y on the last step. This is the vector e in the description of the error control. It is defined only on a successful return from ISOLVE</td>
</tr>
</tbody>
</table>

**Part II: Other routines callable**

**The following are optional calls which the user may make to gain additional capabilities in conjunction with ISOLVE (the routines xerem and xsetf are designed to conform to the static error handling package.)**

**Form of call**

**Function**

**call setun(lun)** set the logical unit number, lun, for output of messages from ISOLVE, if the default is not desired, the default value of lun is 6.

**call setfl(mflag)** set a flag to control the printing of messages by ISOLVE

| mflag = 0 | Do not print. (Danger.) |
| mflag = 1 | This risks losing valuable information. |
| mflag = 2 | Always print (the default). |
| mflag = 10 | Either of the above calls may be made at any time and will take effect immediately. |

**call stcom(cre, iavex, job)** saves and restores the contents of the internal common blocks used by
c c c c c c c c c c c c
isnode (see part iii below).
	raw must be a real array of length 218

or more, and raw must be an integer

array of length 41 or more.

job=1 means save common into raw/sav.

job=2 means restore common from raw/sav.

srcm is useful if one is

interrupting a run and restarting

later, or alternating between two or

more problems solved with isode.

c c c c c c c c c c c c
 call indy(%...)

provide derivatives of y, of various

orders, at a specified point t, if

desired. it may be called only after

a successful return from isode.

c c c c c c c c c c c c
 the detailed instructions for using indy are as follows.

c c c c c c c c c c c c
 the form of the call is...

c c c c c c c c c c c c
call indy (t, k, rwork(21), nyh, dky, iflag)

c c c c c c c c c c c c
 c the input parameters are...

c c c c c c c c c c c c
c t = value of independent variable where answers are desired

(normally the same as the t last returned by isode).

c c c c c c c c c c c c
c for valid results, t must lie between the first and the last of

c c c c c c c c c c c c
c (see optional outputs for first and last of

c c c c c c c c c c c c

c k = integer order of the derivative desired. k must satisfy

0 ≤ k ≤ 2, where nqcur is the current order

(see optional outputs) the capability corresponding

to k = 0, i.e., computing y(t), is already provided

by isode directly. since nqcur ge 2, the first

derivative dy/dt is always available with indy.

c c c c c c c c c c c c

c rwork(21) = the base address of the history array yh.

c c c c c c c c c c c c

nyh = column length of yh, equal to the initial value of neq.

c c c c c c c c c c c c
 c the output parameters are...

c c c c c c c c c c c c
c dky = a real array of length neq containing the computed value

of the k th derivative of y(t).

c iflag = integer flag, returned on 0 if k and t were legal.

1 if k was illegal, and -2 if t was illegal

on an error return, a message is also written.

c c c c c c c c c c c c

c part iii. common blocks.

c c c c c c c c c c c c
 c if isode is to be used in an overlay situation, the user

c must declare, in the primary overlay, the variables in

(1) the call sequence to isode,

(2) the two internal common blocks

/iso001/ of length 257 (218 double precision words)

followed by 39 integer words),

/iso0001/ of length 2 (integer words).

c c c c c c c c c c c c
 c if isode is used on a system in which the contents of internal

c common blocks are not preserved between calls, the user should

c declare the above two common blocks in his main program to insure

c that their contents are preserved.

c c c c c c c c c c c c
 c if the solution of a given problem by isode is to be interrupted

c and then later continued, such as when restarting an interrupted run

c or alternating between two or more problems, the user should save,

following the return from the last isode call prior to the

interruption, the contents of the call sequence variables and the

c internal common blocks, and later restore these values before the

c next isode call for that problem. to save and restore the common

c blocks, use subroutine srcm (see part ii above).

c c c c c c c c c c c c
 c c c c c c c c c c c

c part iv. optionally replaceable solver routines.

c c c c c c c c c c c c
 c below are descriptions of two routines in the isode package which

c relate to the measurement of errors; either routine can be

replaced by a user-supplied version, if desired. however, since such

a replacement may have a major impact on performance, it should be

done only when absolutely necessary, and only with great caution.

(c note: the means by which the package version of a routine is

c superseded by the user's version may be system-dependent.)

c c c c c c c c c c c c
 c (a) ewset.

the following subroutine is called just before each internal

c integration step, and sets the array of error weights, ewt, as
described under /io(t), tao/ above.

subroutine ewset (neq, tol, tao, yr, ewt)

c where neq, tol, tao, and yr are as in the isode call sequence.

c yr contains the current dependent variable vector, and

c ewt is the array of weights set by ewset.

c c c c c c c c c c c c
 c if the user supplies this subroutine, it must return in ewt(i)

(i = 1, ...neq) a positive quantity suitable for comparing errors

c in y(i) to the ewt array returned by ewset is passed to the

c norm routine (see below), and also used by isode in the computation

c of the optional output imx, the diagonal jacobian approximation,

c and the increments for difference quotient jacobians.

c c in the user-supplied version of ewset, it may be desirable to use

c the current values of derivatives of y derivatives up to order

c neq are available from the history array yh, described above under

c optional outputs, in ewset, yh is identical to the yr array.

c extended to no + n i columns with a column length of nyh and scale

c factors of y*y (where y(i) is on the first call for the problem.

c given by y(i) = 0, neq is 1 and it is temporarily set to 1.0.

c the quantities yh, nyh, and not can be obtained by including

c in ewset the statements...

c c c c c c c c c c c c

double precision h, it

common f,io(t), tao,...

c c c c c c c c c c c c
 nyh = it(14)
c
n = 1 (*36*)

c
h = rts(212)

c thus, for example, the current value of dyth can be obtained as
c yr(hy=h*h) y(=1,...,n) (and the division by h is
unnecessary when n = 0).

c
(c) vnorm:
c the following is a real function routine which computes the weighted
root-mean-square norm of a vector v.

c d = vnorm (n, v, w)

c where:

c n = the length of the vector,

c v = real array of length n containing the vector,

c w = real array of length n containing weights,

c d = sqrt (n/2) * sum(v(i)^2*w(i))

vnorm is called with n = n and with w(i) = 1.0, i.e., w(i) = 1.0.

c ewt is as set by subroutine ewset.

c
if the user supplies this function, it should return a non-negative
value of vnorm suitable for use in the error control in isode.
none of the arguments should be altered by vnorm.
for example, a user-supplied vnorm routine might...

- substitute a max-norm of (v(i)^2*w(i)) for the rms-norm, or
- ignore some components of v in the norm, with the effect of
supressing the error control on those components of y

-************************************************************************
- other routines in the isode package.
-************************************************************************

c
(c) in addition to subroutine isode, the isode package includes the
following subroutines and function routines:
c indy computes an interpolated value of the y vector at t = tout.
c isode is the core integrator, which does one step of the
c integration and the associated error control.
c cfode sets all method coefficients and test constants.
c prep computes and preprocesses the Jacobian matrix J = df/dy.
c and the newton iteration matrix p = I - a^T J.
c solty manages solution of linear system in chord iteration.
c ewset sets the error weight vector ewt before each step.
c vnorm computes the weighted rms norm of a vector.
c exvm is a user-callable routine to save and restore
the contents of the internal common blocks.
c dgefs and dgepl are routines from linpack for solving full
systems of linear algebraic equations.
c dgefa and dgepl are routines from linpack for solving banded
linear systems.
c daxpy, dscs, idaxsax, and ddot are basic linear algebra modules
(cf. bk3) used by the above linpack routines.
c dlmach computes the unit roundoff in a machine-independent manner.
c xerrw, xsetu, and xsetf handle the printing of all error
messages and warnings. xerrw is machine-dependent.
c note: vnorm, idaxsax, ddot, and dlmach are function routines
all the others are subroutines.

c
(c the intrinsic and external routines used by isode are:
c dabs, dmax1, dmin1, dfloat, max0, min0, mod, sqrt, and write.
c
(c) a block data subprogram is also included with the package,
c for loading some of the variables in internal common.
c
-************************************************************************
- the following card is for optimized compilation on itl compilers.
c c
-************************************************************************

-************************************************************************
- external preproc solvy
-************************************************************************

integer illin, init, lyh, lowe, ltrav, ltrav, ltw, liwm, liwn,

1 maxstep, maxs, minr, mrepl, mrepl, nlyh, ilw, ilw,

1 integer icf, ielp, ierp, ierp, icf, jsat, fflag, iflag, iflag, iflag,

1 maxord, maxorder, mpsv, maxf, n, nq, m, nq, iflag, iflag, iflag,

1 n, nt, iflag, iflag, nq, icf, icf, icf, iflag, iflag, iflag,

1 intw, intw, intw, intw, intw, intw, intw, intw, intw, intw, intw,

1 double precision row, row, row, row, row, row, row, row, row,

1 ecm, ecm, ecm, ecm, ecm, ecm, ecm, ecm, ecm, ecm, ecm,

1 err, err, err, err, err, err, err, err, err, err, err,

1 logical intw

-************************************************************************
- the following internal common block contains
-************************************************************************

c (a) variables which are local to any subroutine but whose values must
\n be preserved between calls to the routine (own variables),
c (b) variables which are communicated between subroutines:
c the structure of the block is as follows...

-************************************************************************
- common /f0001/ row(209).

1 ecm, ecm, ecm, ecm, ecm, ecm, ecm, ecm, ecm, ecm, ecm,

1 err, err, err, err, err, err, err, err, err, err, err,

1 logical intw

-************************************************************************
- block a
-************************************************************************

c this code block is executed on every call.
c it tests state andsplash for legality and branches appropriately.
c if state > 1 but the flag init shows that initialization has
not yet been done, an error return occurs.
c if state = 1 and tout = 1, jump to block g and return immediately.
if (istate .lt. 1 .or. istate .gt. 3) go to 601
if (istate .lt. 1 .or. istate .gt. 5) go to 602
if (istate .ne. 1) go to 10
if (init .eq. 0) go to 603
if (istate .eq. 2) go to 200
  go to 20
10  init = 0
  if (tent .eq. 1) go to 430
20  nstep = 0

c---------------------------------------------------
  c block b
  c the next code block is executed for the initial call (istate = 1).
  c or for a continuation call with parameter changes (istate = 3)
  c it contains checking of all inputs and various initializations
  c
  c first check legality of the non-optional inputs neq, init, lept,
  c mf, ml, and mu.
  c---------------------------------------------------

  if (neq(1) .le. 0) go to 604
  if (istate .eq. 1) go to 25
  if (neq(1) .gt. n) go to 605
25  n = neq(1)
  if (nitol .lt. 1 .or. nitol .gt. 4) go to 606
  if (ntopt .le. 0 .or. nipt .gt. 1) go to 607
  meth = mf/10
  niter = mf - 10*meth
  if (meth .lt. 1 .or. meth .gt. 2) go to 608
  if (niter .le. 0 .or. niter .gt. 5) go to 608
  if (niter .le. 3) go to 30
20  ml = iwork(1)
  mu = iwork(2)
  if (ml .lt. 0 .or. ml .ge. n) go to 609
  if (mu .lt. 0 .or. mu .ge. n) go to 610
30  continue
  c next process and check the optional inputs.  -------------------
  if (npt .eq. 1) go to 40
  maxord = mord(meth)
  maxstep = mstep0
  mshnll = mshn0
  if (istate .eq. 1) h0 = 0.000
  hmax = 0.000
  hmin = 0.000
  go to 60
40  maxord = iwork(5)
  if (maxord .lt. 0) go to 611
  if (maxord .gt. 0) maxord = 100
  maxord = min(maxorder, maxord, mord(meth))
  maxstep = iwork(6)
  if (maxstep .lt. 0) go to 612
  if (maxstep .gt. 0) maxstep = mstep0
  mshnll = iwork(7)
  if (mshnll .lt. 0) go to 613
  if (mshnll .ge. 0) mshnll = mshn0
  if (istate .ne. 1) go to 50

h0 = rwork(5)
if (ttype .eq. 1) h0 = 0.000 go to 614
50  hmax = rwork(6)
  if (hmax .lt. 0.000) go to 615
  hmax = 0.000
  if (hmax .gt. 0.000) hmax = 1.000/hmax
  hmin = rwork(7)
  if (hmin .lt. 0.000) go to 616
  c---------------------------------------------------
  c set work array pointers and check lengths lw and lw.
  c pointers to segments of rwork and iwark are named by prefixing 1 to
  c the name of the segment.  e.g., the segment yh starts at rwork(1,yh).
  c segments of rwork (in order) are denoted yh, wmn, ewt, savf, scon.
  c---------------------------------------------------

60  yh = 21
  if (istate .eq. 1) yh = n
  iwm = yh + (maxord + 1)*nyh
  if (niter .eq. 0) lenw = 0
  if (niter .eq. 1 .or. niter .eq. 2) lenw = n+n+2
  if (niter .eq. 3) lenw = n+n+2
  if (niter .eq. 4) lenw = (2*m + mu + 1)*n+2
  lenw = iwm + lenw
  lavef = lave + n
  lace = lavef + n
  lenw = lace + n + 1
  iwork(17) = lenw
  iwork(18) = lenw
  lw = 20 + n
  if (niter .eq. 0 .or. niter .eq. 3) lenw = 20
  iwork(19) = lenw
  if (lenw .gt. lw) go to 617
  if (lenw .gt. lw) go to 618
  c check ritol and atol for legality
  ritol = ritol(1)
  atol = atol(1)
  do 70 = 1, n
  if (itol .eq. 3) ritol = ritol(i)
  if (itol eq 2 .or. itol eq 4) atol = atol(i)
  if (itol .eq. 0) atol = 0.000
  if (itol .lt. 0 .or. atol .gt. 0.000) go to 620
  70  continue
  if (istate .eq. 1) go to 100
  c if istate = 3, set flag to signal parameter changes to stode. -------
  jstart = 1
  if (ng .le. maxord) go to 90
  c maxord was reduced below ng. copy yh(*,maxord+2) into savf -------
  do 80 i = 1, n
  80  rwork(i) = savf(i)
  c reload wmn(i) = rwork(i,lenw), since lenwm may have changed. -------
  if (nle .eq. njh) go to 299
  c if maxim was reduced, zero part of yh to avoid undefined references. -------
  i = yh + 1
  j2 = yh + (maxord + 1)*nyh - 1
if (l1 > l2) go to 200
  do 95 i = l1, l2
  95 rwork(i) = 0.0d0
  go to 200

C-------------------------------------------------------------------------------
  c block c.
  c the next block is for the initial call only (istate = 1).
  c it contains all remaining initializations, the initial call to f,
  c and the calculation of the initial step size.
  c the error weights in ewt are inverted after being loaded.
  c-------------------------------------------------------------------------------
100  around = d1mach(4)
    in = i
    if (itask ne 4 .and. itask ne 5) go to 110
    tcrit = rwork(1)
    if ((itryt - tout)**(tout - t).lt. 0.0d0) go to 625
    if (bnm ne 0.0d0 .and. (n + h0 - tcrit)*h0 .ge. 0.0d0)
      1 h0 = tcrit - t
    110  jstart = 0
    if (intit .gt. 0) rwork(1wm) = dsqrt(around)
    nsho = 0
    nst = 0
    nje = 0
    nlast = 0
    hu = 0.0d0
    nve = 0
    cmax = 0.0d0
    maxcor = 3
    msho = 20
    maxfs = 10

  c initial call to f. (fn0 points to yh(*,2)). -------------------------------
  fn0 = lbh + nyh
  call f (nqeq, l, y, rwork(fn0))
  nfe = 1
  c load the initial value vector in yh. --------------------------------------
  do 115 i = 1, n
    115 rwork(i+yh-1) = yh(i)
  c load and invert the ewt array. (t is temporarily set to 1.0) ------
    nq = 1
    h = 1.0d0
    call ewt(n, itol, ritol, atol, rwork(lyh), rwork(lew))
    do 120 i = 1, n
      if (rwork(i+lew-1).le. 0.0d0) go to 621
      120 rwork(i+lew-1) = 1.0d0/rwork(i+lew-1)
  c-------------------------------------------------------------------------------
  c the coding below computes the step size, h0, to be attempted on the
  c first step, unless the user has supplied a value for this
  c first check that tout - t differs significantly from zero
  c a scalar tolerance quantity tol is computed, as max(titol(i))
  c if this is positive, or max(atol(i)+dtol(titol(i))) otherwise, adjusted
  c so as to be between 100*around and 1.0e-3.
  c then the computed value h0 is given by:
  c-------------------------------------------------------------------------------
    c
    c h0**2 = tol / ( w0**2 + 2 * (1/nveq) * sum ( f(i)*ywty(f(i)) )**2 )
if (tcris - tout)*h. lt. 0.000 go to 625
if (tn - tout)*h. lt. 0.000 go to 245
call indx(tout, 0, rwork(lyh), nyh, y, iflag)
if (iflag. ne. 0) go to 627
k = tout
go to 420
240 tcris = rwork(1)
if (ttn - tcris)*h. gt. 0.000 go to 624
245 hmax = dabs(tn - tcris) + dabs(h)*

ibht = dabs(tn - tcris) + dabs(h)*
le 100.000*around*hmax
if (ibht) go to 400
wnt = tn + h'(1.000 + 4.000*around)
if ((wnt - tctn)*h. le. 0.000 go to 250
h = (tctn - wnt)'(1.000 + 4.000*around)
if (jstate eq. 2) jstart = -2

C-----------------------------------------------
c block e,
c the next block is normally executed for all calls and contains
c the call to the one-step core integrator code.
c
c this is a looping point for the integration steps.
c
c first check for too many steps being taken, update ewt (if not at
c start of problem), check for too much accuracy being requested, and
c check for h below the roundoff limit in t.
c
C-----------------------------------------------
250 continue
if (ttn - nlowest). ge. maxstep go to 500
call ewt(n, itol, rtol, atol, rwork(lyh), rwork(lye))
do 260 i = 1, n
if (rwork(i + lew) - le) 0.000 go to 510
260 rwork(i + lew) = 1.0000*rwork(i + lew - 1)
270 tofsl = around*norm(a, rwork(lyh), rwork(lye))
if (tofsfle 1.0000 go to 280
tofsf = tofsf*2.000
if (nst eq. 0) go to 626
go to 520
go to 520
280 if (tn = nhighestl n) go to 290
if (nshnl gt. mshnr) go to 290
C-----------------------------------------------
c call stode(n, y, yh, yeh, ewt, sav, facr, wmn, iwm, f, jac, pres, solsy)
c call stode(n, y, rwork(lyh), yb, yb0, rwork(lye), rwork(ley)),
1 rwork(lye)), rwork(acy), rwork(lyw), rwork(lyew),
2 f, jac, pre, solsy, temp, imax)
kg = 1 - kflag
go to (300, 530, 540, kg)
c-----------------------------------------------
c block f,
c the following block handles the case of a successful return from the
c core integrator (kflag = 0), test for stop conditions.
c
300 init = 1
go to (310, 400, 330, 340, 350, init)
c itask = 1. if tout has been reached, interpolate.------------------------
310 if (ttou - tout)*h. lt. 0.000 go to 250
call indy(tout, 0, rwork(lyh), nyh, y, iflag)
t = tout
go to 420
320 itask = 3. jump to exit if tout was reached.-----------------------------
330 if (ttou - tout)*h. ge. 0.000 go to 400
go to 250
340 itask = 4. see if tout or tcris was reached. adjust h if necessary.
350 if (ttou - tcris)*h. lt. 0.000 go to 345
call indy(tout, 0, rwork(lyh), nyh, y, iflag)
t = tout
go to 420
345 hmax = dabs(tn - tctn) + dabs(h)*

ibht = dabs(tn - tctn) + dabs(h)*
le 100.000*around*hmax
if (ibht) go to 500
wnt = tn + h'(1.000 + 4.000*around)
if ((wnt - tctn)*h. le. 0.000 go to 250
h = (tctn - wnt)'(1.000 - 4.000*around)
jstart = -2
go to 250
c itask = 5. see if tcris was reached and jump to exit.---------------------
350 hmax = dabs(tn - tctn) + dabs(h)*

ibht = dabs(tn - tctn) + dabs(h)*
le 100.000*around*hmax
C-----------------------------------------------
c block g,
c the following block handles all successful returns from stode.
c if stirn, ne. 1, y is loaded from yh and t is set accordingly
c iestate is set to 2, the illegal input counter is zeroed, and the

c optional outputs are loaded into the work arrays before returning.
c if iestate = 1 and tout = t, there is a return with no action taken.
c except that if this has happened repeatedly, the run is terminated.
c
400 do 410 i = 1, n
410 y(i) = rwork(i + lyh - 1)
t = tn
if (itask ne. 4 and itask ne. 5) go to 420
if (istate eq. tcris)
420 istate = 2
flan = 0
rwork(11) = h
rwork(12) = h
rwork(13) = tn
iwork(11) = nst
iwork(12) = nfe
iwork(13) = nje
iwork(14) = nqu
iwork(15) = nq
return
c
430 step = nstep + 1
if (nstep > 5) return
   call xerrw(1
   big(50ihsode=-- repeated calls with iistate = 1 and tout = t (=1) ,
      1 50, 201, 0, 0, 0, 0, 1, t, 0, 0)
go to 500

500 call xerw(50ihsode=-- at current t (=1), nstep (=1) steps .
      1 50, 201, 0, 0, 0, 0, 0, 0, 0, 0
      call xerw(50ihsode=-- taken on this call before reaching tout .
      1 50, 201, 0, 0, 0, 0, 0, 0, 0
      istate = -1
      go to 580

c ew(1) = 0.0 for some i (not at start of problem .
510- ew(1) = xwork(i+ew(1)-1)
      call xerw(50ihsode=-- at t (=1), ew(1) has become r2. le 0 .
      1 50, 202, 0, 1, 0, 2, tm, ew(1))
istate = -6
      go to 580

c too much accuracy requested for machine precision .
520 call xerw(50ihsode=-- at t (=1), too much accuracy requested .
      1 50, 203, 0, 0, 0, 0, 0, 0, 0, 0
      call xerw(50ihsode=-- for precision of machine .
      see tosdf (=2) ,
      1 50, 203, 0, 0, 0, 0, 2, tm, tosdf)
rwork(14) = tosdf
istate = -2
      go to 580

c kflag = -1, error test failed repeatedly or with abs(h) = hmin .
530 call xerw(50ihsode=-- at t (=1) and step size h (=2) , the error .
      1 50, 204, 0, 0, 0, 0, 0, 0, 0, 0
      call xerw(50ihsode=-- test failed repeatedly or with abs(h) = hmin .
      1 50, 204, 0, 0, 0, 0, 2, tn, h)
istate = 4
      go to 560

c kflag = -1, convergence failed repeatedly or with abs(h) = hmin .
540 call xerw(50ihsode=-- at t (=1) and step size h (=2) ,
      1 50, 205, 0, 0, 0, 0, 0, 0, 0, 0
      call xerw(50ihsode=-- corrector convergence failed repeatedly .
      1 50, 205, 0, 0, 0, 0, 0, 0, 0, 0
      call xerw(50ihsode=-- or with abs(h) = hmin .
      1 50, 205, 0, 0, 0, 0, 2, tm, h)
istate = -5

c compute imset if relevant .
540 imset = 0
      imset = 1
      do 570 i = 1, n
      size = dtah(t)=work(i+1,1)+work(i+1,1)
      if (size .ge. size) go to 570
      big = size
      imset = i
      go to 570

570 continue
iwork(10) = imset
      c set y vector, i, iln, and optional outputs .
580 do 590 i = 1, n
      590 y(i) = xwork(i+1,1)
t = t
      iln = 0
      rwork(11) = hu
      rwork(12) = h
      rwork(13) = tn
      rwork(14) = nje
      rwork(15) = nq
      rwork(16) = nqu
      return

580- c block i
590 - c the following block handles all error returns due to illegal input

600 call xerw(50ihsode=-- iistate (=1) illegal ,
      1 50, 1, 0, 0, 1, istate, 0, 0, 0, 0, 0, 0)
go to 700

601 call xerw(50ihsode=-- iistate (=1) illegal ,
      1 50, 2, 0, 0, 1, istate, 0, 0, 0, 0, 0, 0)
go to 700

602 call xerw(50ihsode=-- iistate (=1) illegal ,
      1 50, 3, 0, 0, 1, istate, 0, 0, 0, 0, 0, 0)
go to 700

603 call xerw(50ihsode=-- iistate (=1) but node not initialized .
      1 50, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
go to 700

604 call xerw(50ihsode=-- neq (=1) l t 1 ,
      1 50, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
go to 700

605 call xerw(50ihsode=-- iistate (=1) and neq increased (11 to 12) .
      1 50, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
go to 700

606 call xerw(50ihsode=-- iistate (=1) and neq increased (11 to 12) .
      1 50, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
go to 700
606 call serrw(30) node --> ilot (=11) Illegal .
  l  30, 6, 0, 1, ilot, 0, 0, 0.000, 0.000
  go to 700
607 call serrw(30) node --> ilot (=11) Illegal .
  l  30, 7, 0, 1, ilot, 0, 0, 0.000, 0.000
  go to 700
608 call serrw(30) node --> mf (=11) Illegal .
  l  30, 8, 0, 1, mf, 0, 0, 0.000, 0.000
  go to 700
699 call serrw(30) node --> ml (=11) Illegal . It 0 or ge neq (=2).
  l  50, 9, 0, 2, ml, neq(1), 0, 0.000, 0.000
  go to 700
610 call serrw(30) node --> ms (=11) Illegal . It 0 or ge neq (=2).
  l  50, 10, 0, 2, ms, neq(1), 0, 0.000, 0.000
  go to 700
611 call serrw(30) node --> maxord (=11) lt. 0 .
  l  30, 11, 0, 1, maxord, 0, 0, 0.000, 0.000
  go to 700
612 call serrw(30) node --> mstep (=11) lt. 0 .
  l  30, 12, 0, 1, mstep, 0, 0, 0.000, 0.000
  go to 700
613 call serrw(30) node --> maxnl (=11) h. 0 .
  l  30, 13, 0, 1, maxnl, 0, 0, 0.000, 0.000
  go to 700
614 call serrw(40) node --> tout (=11) behind (=2) .
  l  40, 14, 0, 0, 0, 0, 0, 2, tout, t
  call serrw(50) node --> integration direction is given by h0 (=11) .
  l  50, 14, 0, 0, 0, 0, 1, h0, 0.000
  go to 700
615 call serrw(30) node --> hmax (=11) lt. 0 .
  l  30, 15, 0, 0, 0, 0, 0, 1, hmax, 0.000
  go to 700
616 call serrw(30) node --> hmin (=11) lt. 0 .
  l  30, 16, 0, 0, 0, 0, 1, hmin, 0.000
  go to 700
617 call serrw
  l  60hiloode-- rwork length needed, lenw (=11), exceeds lw (=12).
  l  60, 17, 0, 2, lenw, lw, 0.000, 0.000
  go to 700
618 call serrw
  l  60hiloode-- rwork length needed, lenw (=11), exceeds lw (=12).
  l  60, 18, 0, 2, lenw, lw, 0.000, 0.000
  go to 700
619 call serrw(40) node --> rtol(1) is rl. lt. 0 .
  l  40, 19, 0, 1, i, 0, 1, rtol, 0.000
  go to 700
620 call serrw(40) node --> atol(1) is rl. lt. 0 .
  l  40, 20, 0, 1, i, 1, 0, atol, 0.000
  go to 700
621 call serrw rwork(neww+1)
  l  40, 21, 0, 1, i, 0, 1, rwork, 0.000
  go to 700
622 call serrw
this routine manages the solution of the linear system arising from
chord iteration. It is called if miter ne 0.
cif miter is 1 or 2, it calls dgesl to accomplish this.
cif miter = 3 it updates the coefficient b^2/0 in the diagonal
matrix, and then computes the solution.
cif miter is 4 or 5, it calls dgbk.
ccommunication with solsy uses the following variables:
c wvm = real work space containing the inverse diagonal matrix if
c miter = 3 and the lu decomposition of the matrix otherwise.
c storage of matrix elements starts at wvm(3).
c wvm also contains the following matrix-related data.
c wvm(1) = sqrt(111111) (not used here).
c wvm(2) = b^2/0, the previous value of b^2/0 if used.
c iwm = integer work space containing pivot information, starting at
iwm(21), if miter is 1, 2, 4, or 5, iwm also contains band
parameters ml = iwm(1) and mu = iwm(2) if miter is 4 or 5.
c x = the right-hand side vector on input, and the solution vector
on output, of length n.
c tem = vector of work space of length n, not used in this version.
c iersl = output flag (in common): iersl = 0 if no trouble occurred.
c iersl = 1 if a singular matrix arose with miter = 3.
c this routine also uses the common variables e0, h, miter, and n.
c
iersl = 0
go to (100, 100, 300, 400, 400, 400), miter
100 call dgesl (wvm(3), n, n, iwm(21), x, 0)
return c

300 phib = wvm(2)
qphib = nphib
wvm(2) = phib
if (phib = qphib) go to 330
r = phib
do 320 i = 1, n
320 di = 1.000 - r*(1.000 - 1.000*wvm(i+2))
if (dphib(di) = qphib) go to 390
390 iersl = 1
return
c
340 s(i) = wvm(i+2)*x(i)
return
390 iersl = 1
return
c
400 ml = iwm(1)
u = iwm(2)
meband = 2*ml + mu + 1
call dgbk (wvm(3), meband, n, ml, mu, iwm(21), x, 0)
return c

end subroutine solsy

---------- end of subroutine solsy ----------
end subroutine prepj (neq, y, yh, yhy, e0, etol, fcmp, savf, wvm, iwm,
  f, jac)
call equilure
e external f, jac
integer neq, yhy, iwm
to rend, iwm,
1 icf, ierrj, iersl, jres, jstart, kflag, l, miter, n,
2 maxc0, maxcol, mshp, nmaxf, nq, nst, nfe, njj, nq
  integer i, j, k, l, r, s, y, y, yj, zj
1 min, mband, ml, mu, ml, mu, np1
double precision y, yh, etol, fcmp, savf, wvm
2 double precision rows
1 cmax, e0, h, lmax, ymax, u, rc, m, tr, un, urow
2 double precision con, di, fac, h0, r, s, df, r, t, y, yj, zj
1 yvmax
dimension neq(1), y(1), yh(yhy, yhy, yhy), (neq(1), fcmp(1), savf(1),
1 wvm(50), iwm(50)
common f(0000), rows(wvm(20)),
2 e0, h, lmax, ymax, u, rc, m, tr, un, urow,
3 iwm(40), iwm(60),
4 icf, ierrj, iersl, jresl, jstart, kflag, l, miter,
5 maxc0, maxcol, mshp, nmaxf, nq, nst, nfe, njj, nq
DOUBLE PRECISION TEMPEPS(2000), KBG, KPG, EP, EB, AO,
INTEGER IMAX

common /PARAMTR/ TEMP, IMAX

RETURN
\texttt{c miter, n, nfe, and nje.}
\texttt{c}
\begin{verbatim}
  nje = nje + 1
  iterp = 0
  jcor = 1
  h0 = h * e0

do to (100, 200, 300, 400, 500), miter
\end{verbatim}

\texttt{c if miter = 1, call jac and multiply by scalar.}
\begin{verbatim}
100  lerp = n * n
     do 110 i = 1, lerp
     110  wmi(i) = 0.0
     call jac (seq, tn, y, 0, 0, wmi(3), n)
     con = h0
     do 120 i = 1, lerp
     120  wmi(i) = wmi(i) * con
     go to 240
\end{verbatim}

\texttt{c if miter = 2, make n calls to f to approximate j.}
\begin{verbatim}
200  fac = 1.0 * float(n) / float(n)
     r0 = 1000.0 * h * h * float(n) * fac
     if (r0 eq 0.060) r0 = 1.060
     snur = wmi(1)
     j1 = 2
     do 230 j = 1, n
          y(j) = y(j)
          r = dmax1 (snur * dabs(y(j)), 0.0, wmi(1))
          y(j) = y(j) + r
          fac = -h0
          call f (seq, tn, y, xem)
     230  enddo
     do 220 i = 1, n
     220  wmi(i) = dsum (i, savef(i)) * fac
          y(j) = y(j)
          j1 = j1 + n
     230  continue
\end{verbatim}

\texttt{c add identity matrix.}
\begin{verbatim}
240  j = 3
     npj = n + 1
     do 250 i = 1, n
     250  wmi(i) = wmi(i) + 1.0
     250  j = j + npj
\end{verbatim}

\texttt{c do lu decomposition on p.}
\begin{verbatim}
call dgefa (wmi(3), n, n, wmi(21), lerp)
if (iterp ne 0) iterp = 1
return
\end{verbatim}

\texttt{c if miter = 3, construct a diagonal approximation to j and p.}
\begin{verbatim}
300  wmi(2) = h0
     r = r0 * 1.0
     do 310 i = 1, n
     310  (yi(i) = yi(i) + r * h * savef(i) - yh(i, 2))
     call f (seq, tn, y, wmi(1))
     nfe = nfe + 1
     do 320 i = 1, n
     320  r0 = h * savef(i) - yh(i, 2)
     di = 0.1 * r0 - h * (wmi(i+2) - savef(i))
     \texttt{wmi(i+2) = 1.000}
     if (dabs(r0) eq 0.000) go to 320
     if (dabs(di) eq 0.000) go to 330
     wmi(i+2) = 0.1 * r0 + di
     320  continue
     return
     330  iterp = 1
     return
\end{verbatim}

\texttt{c if miter = 4, call jac and multiply by scalar.}
\begin{verbatim}
400  ml = wmi(1)
     mu = wmi(2)
     ml3 = ml + 3
     mband = ml + mu + 1
     mehand = mband + ml
     lenp = mehand * n
     do 410 i = 1, lenp
     410  wmi(i+2) = wmi(i+2) * con
     go to 570
\end{verbatim}

\texttt{c if miter = 5, make mband calls to f to approximate j.}
\begin{verbatim}
500  ml = wmi(1)
     ml3 = ml + 3
     mband = ml3 + mu + 1
     mehand = mband + ml
     mehand = mehand - 1
     snur = wmi(1)
     fac = vnorm (n, savef, ewt)
     r0 = 1000.0 * h * float(n) * fac
     if (r0 eq 0.060) r0 = 1.060
     do 500 i = 1, mba
     530  y(i) = yi(i)
     r = dmax1 (snur * dabs(y(i)), 0.0, wmi(1))
     y(i) = y(i) + r
     call f (seq, tn, y, xem)
     do 550 j = 1, n
     550  wmi(i+j) = dsum (i, savef(i)) * fac
     540  continue
     nfe = nfe + mba
     c add identity matrix
\end{verbatim}
do 580 i = 1, n
   wmi(2) = wmi(2) + 1.0/d0
580   it = it + 1
   call dbfba (wmi(3), mb, n, m, mu, iwmi(21), ier)
   if (ier .ne. 0) ierpi = 1
end

-----------------------------------------------------------------------------
end subroutine stode (neq, y, yh, yhl, cwt, savf, acor)

1. wm, iwmi, f, jac, pjac, slvs

cell optimize external f, jac, pjac, slvs
   integer neq, nh, iwmi
   integer iownd, iath, iup, imax, mex, nqyhp, mlp,
                   i, j, k, l, m, n, ncf, newq
   double precision y, yhl, cwt, savf, acor, wmi
   double precision con, crute, e1, e2, e3, e4, e5, e6,
                   ex, ef, f, g, h, hmax, hr, jt, m, ncf, newq
   common /x000/1/ con, crute, e1(13), e2(13), e3(13), e4(13),
                  e5(13), e6(13), f(13), g(13), h(13),
                  i(13), j(13), k(13), l(13), m(13), n(13),
                  ncf(13), newq(13), x00000(13), x00000(13),
                  x00000(13), x00000(13), x00000(13)
   integer /x000/1/ con, crute, e1(13), e2(13), e3(13),
                  e4(13), e5(13), e6(13), f(13), g(13), h(13),
                  i(13), j(13), k(13), l(13), m(13), n(13),
                  ncf(13), newq(13), x00000(13), x00000(13),
                  x00000(13), x00000(13), x00000(13)
   common /x000/1/ con, crute, e1(13), e2(13), e3(13),
                  e4(13), e5(13), e6(13), f(13), g(13), h(13),
                  i(13), j(13), k(13), l(13), m(13), n(13),
                  ncf(13), newq(13), x00000(13), x00000(13),
                  x00000(13), x00000(13), x00000(13)
   common /x000/1/ con, crute, e1(13), e2(13), e3(13),
                  e4(13), e5(13), e6(13), f(13), g(13), h(13),
                  i(13), j(13), k(13), l(13), m(13), n(13),
                  ncf(13), newq(13), x00000(13), x00000(13),
                  x00000(13), x00000(13), x00000(13)
   common /x000/1/ con, crute, e1(13), e2(13), e3(13),
                  e4(13), e5(13), e6(13), f(13), g(13), h(13),
                  i(13), j(13), k(13), l(13), m(13), n(13),
                  ncf(13), newq(13), x00000(13), x00000(13),
                  x00000(13), x00000(13), x00000(13)

c cwt = an array of length n containing multiplicative weights
   for local error measurements. local errors in y(i) are
   compared to 1.0e(wti) in various error tests.
   savf = an array of working storage, of length n.
   also used for input of yhl .maxed &2 when jstart = -1
   acor = a work array of length n, used for the accumulated
         corrections. on a successful return, acor(i) contains
         the estimated one-step local error in y(i).
   wmi, iwmi = real and integer work arrays associated with matrix
         operations in chord iteration (mier .ne. 0).
   p, q = name of routine to evaluate and preprocess jacobian matrix
   p and q = i + b*el+/jac, if a chord method is being used.
   slvs = name of routine to solve linear system in chord iteration.
   cmax = maximum relative change in h*0 before pjac is called.
   h = the step size to be attempted on the next step.
   h is altered by the error control algorithm during the
   problem. h can be either positive or negative, but its
   sign must remain constant throughout the problem.
   hmin = the minimum absolute value of the step size h to be used.
   hmax = inverse of the maximum absolute value of h to be used.
   hmax = 0.0 is allowed and corresponds to an infinite hmax.
   hmin and hmax may be changed at any time, but will not
   take effect until the next change of h is considered
   b = the independent variable is to be updated on each step taken.
   jstart = an integer used for input only, with the following
         values and meanings:
           0 perform the first step
           1 take a new step, multiplying from the last
           -1 take the next step with a new value of h, maxord,
              n, meth, mier, and/or matrix parameters.
           -2 take the next step with a new value of h,
              but with other inputs unchanged
           on return, jstart is set to 1 to facilitate continuation.
   fflag = a completion code with the following meanings:
           0 the step was successful.
           -1 the requested integration could not be achieved.
           -2 incorrect convergence could not be achieved.
           -3 final error in pjac or slvs
           a return with fflag = -1 or -2 means either
           slvs(f) = hmin or 10 consecutive failures occurred
           on a return with fflag negative, the values of to and
           the y array are as of the beginning of the last
           step, and h is the last step size attempted.
   maxord = the maximum order of integration method to be allowed
   maxord = the maximum number of corrector iterations allowed
   mlp = maximum number of steps between pjac calls (mier .gt. 0)
   maxel = maximum number of convergence failures allowed
   meth/mier = the method flags. see description in driver
   n = the number of first-order differential equations
   kflag = 0
   tot = -1
   ncf = 0
iterp = 0
iterl = 0
jeur = 0
tcf = 0
defp = 0.000
if (jstart < 0) go to 200
if (jstart = -1) go to 100
if (jstart = -2) go to 160

c...on the first call, the order is set to 1, and other variables are initialized.
maxO is the maximum ratio by which h can be increased
in a single step. It is initially 1.0 to compensate for the small
initial h, but then is normally equal to 10. If a failure
occurs (in corrector convergence or error test), maxO is set to 2
for the next increase.

1..maxO = maxord + 1
nq = 1
l = 2
ialth = 2
maxO = 100000.0
rc = 0.000
rl = 1.0
br = 0.0
hold = h
meh = meh
mep = 0
ipup = ipam
iret = 3

go to 140

c...the following block handles preliminaries needed when jstart = -1.
ipup is set to ipam to force a matrix update.
if an order increase is about to be considered (ialth = 1),
ialth is reset to 2 to postpone consideration one more step.
if the caller has changed meh, cflode is called to reset
the coefficients of the method.
if the caller has changed maxord to a value less than the current
order nq, nq is reduced to maxord, and a new h chosen accordingly.
if h is to be changed, yh must be rescaled.
if h or meh is being changed, ialth is reset to 1 = nq + 1

c...to prevent further changes in h for too many steps.

140 ipup = ipam
if (ialth eq 1) ialth = 2
if (mech = meh) go to 110
-call cflode (meh, elec, testc)
meh = meh
if (nq gt maxord) go to 120
ialth = 1
iret = 1
go to 150
110 if (nq le maxord) go to 160

120 nq = maxord
l = lmax
do 125 i = 1, l
125 el(i) = elco(i, nq)
nqnyh = nq*nyh
rc = rc*el(i)*h
el(i) = el(i)
conit = 0.5/(0.0000000000000000000)
ddd = vnorm (n, savf, ewt, ascco(1,1)
exdn = 1.0/(0.0000000000000000000)
rdh = 1.0/(1.0000000000000000000)
ry = dmin1(rdh, 1.0000000000000000000)
iredo = 3
if (ireq = hold) go to 170
ry = dmin1(rh, dabs(h/dh))
h = hold
go to 175

c...cflode is called to get all the integration coefficients for the
current meth. Then the el vector and related constants are reset
whenver the order nq is changed, or at the start of the problem.

140 call cflode (meh, elec, testc)
150 do 155 i = 1, l
155 el(i) = elco(i, nq)
nqnyh = nq*nyh
rc = rc*el(i)*h
el(i) = el(i)
conit = 0.5/(0.0000000000000000000)
go to (160, 170, 200, iret

c...c if h is being changed, the h ratio rh is checked against
maxO, hmin, and hmax, and the yh array rescaled. ialth is set to
1 = nq + 1 to prevent a change of h for too many steps, unless
a forced by a convergence or error test failure

160 if (ireq = hold) go to 200
rh = h/dh
h = hold
iredo = 3

go to 175
170 rh = dmax1(rh, hmin/dabs(h))
175 rh = dmin1(rh, rmax)
rh = rh*dmax1(1.0000000000000000000, hmax*rh)
t = 1.0000000000000000000
do 180 j = 2, l
r = rh

dx = 1.0000000000000000000
180 yh(i,j) = yh(i,j)*r
h = h*r
rc = rc*r
ialth = 1
if (iredo eq 0) go to 690

c...
c this section computes the predicted values by effectively
  c multiplying the yh array by the pascal triangle matrix.
  c rc is the ratio of new to old values of the coefficient h*el(1).
  c when rc differs from 1 by more than cmx, ipup is set to miter
  c to force pjac to be called, if a jacobian is involved.
  c in any case, pjac is called at least every mulp steps.

c=======================================================================
  200 if (dabs(rc-1.0d0) .gt. cmx) ipup = miter
      if (nst .ne. nslp+nslp) ipup = miter
      if (nslp .ne. nslp+1) ipup = miter
      tn = tn + h
      if (i+nyh .lt. 1) go to 215
      do 215 jb = 1, nq
        if (i+nyh .lt. 1) go to 215
      ediv $ svedep
      do 210 i = 1, nq
        do 210 i = 1, nq
          y(i) = y(i) + y(i+nyh)
      215 continue

  c=======================================================================
  c up to maxim corrctor iterations are taken. a convergence test is
  c made on the r.m.s. norm of each correction, weighted by the error
  c weighted vector ewt. the sum of the corrections is accumulated in
  c the vector acor(). the yh array is not altered in the corrector loop.

  220 m = 0
  230 do 240 i = 1, n
    y(i) = y(i+1) - h*el(1)*y(i)
  240 call f (nq, tn, y, savf)
  250 nfe = nfe + 1
  260 if (ipup .ne. 0) go to 250
  270 continue

  c=======================================================================
  c if indicated, the matrix p = i - h*el(1)*i is reevaluated
  c and processed before starting the corrector iteration. ipup is set
  c to 0 as an indicator that this has been done.

  c=======================================================================
    c call pjac (nq, y, yh, nyh, ewt, acor, savf, wtm, lim, l, j, temp, imax)
  270 ifpup = 0
  280 rc = 1.0d0
  290 nslp = nst
  300 cmx = rc + 0.7d0
  310 if (srep .ne. 0) go to 340
  320 do 340 i = 1, n
    y(i) = y(i+1) - h*el(1)*y(i)
  340 call f (nq, tn, y, savf)
  350 nfe = nfe + 1
  360 if (ipup .ne. 0) go to 360
  370 do 380 i = 1, n
    y(i) = y(i+1) + el(1)*savf(i)
  380 390 c=======================================================================

  c in the case of the chord method, compute the corrector error,
  c and solve the linear system with Pre = righthand side and
  c p as coefficient matrix.

  350 do 360 i = 1, n
  360 y(i) = h*acor(i) - (y(i) + h*el(1)*y(i+nyh))
  370 call elsv (wnm, iw, y, savf)
  380 if (srep .ne. 0) go to 340
  390 do 390 i = 1, n
    y(i) = y(i) + el(1)*acor(i)
  400 c=======================================================================
  c test for convergence. if mge. an estimate of the convergence
  c rate constant is stored in crat, and this is used in the test
  c=======================================================================
  400 if (m .ge. 0) crate = dmax1 (0.0d0, 1.0d0)
  410 do 420 i = 1, n
    if (nn .ne. 0) go to 410
    y(i) = y(i+1) - h*el(1)*y(i)
  420 call f (nq, tn, y, savf)
  430 nfe = nfe + 1
  440 if (ipup .ne. 0) go to 440
  450 do 450 i = 1, n
    y(i) = y(i+1) + el(1)*savf(i)
  460 470 c=======================================================================

  c the corrector iteration failed to converge.
  c if miter .ne. 0 and the jacobian is not done, pjac is called for
  c the next try. otherwise the yh array is retracted to its values
  c before prediction, and h is reduced, if possible. if h cannot be
  c reduced or miter failures have occurred, exit with iflag = -2.

  430 if (miter .eq. 0 .or. jver .eq. 1) go to 430
  440 if (iflag .ne. 1) go to 220
  450 aflag = 2
  460 if (m .ne. 0) go to 460
  470 if (acor(i) .ne. 0.0d0) go to 470
  480 acor(i) = acor(i) + 1
  490 nn = nst - 1
  500 if (nn .ne. 0) go to 500
  510 if (iflag .eq. 1 .or. iflag .ne. 0) go to 680
  520 if (iflag .ne. 0) go to 670
  530 if (dabs(i) .le. hmin*1.0d0) go to 670
  540 if (dabs(i) .le. hmin*1.0d0) go to 670
  550 if (dabs(i) .le. hmin*1.0d0) go to 670
  560 if (dabs(i) .le. hmin*1.0d0) go to 670
  570 if (dabs(i) .le. hmin*1.0d0) go to 670
  580 if (dabs(i) .le. hmin*1.0d0) go to 670
  590 if (dabs(i) .le. hmin*1.0d0) go to 670
  600 if (dabs(i) .le. hmin*1.0d0) go to 670
  610 if (dabs(i) .le. hmin*1.0d0) go to 670
  620 if (dabs(i) .le. hmin*1.0d0) go to 670
  630 if (dabs(i) .le. hmin*1.0d0) go to 670
  640 if (dabs(i) .le. hmin*1.0d0) go to 670
  650 if (dabs(i) .le. hmin*1.0d0) go to 670
  660 if (dabs(i) .le. hmin*1.0d0) go to 670
  670 if (iflag .ne. 0) go to 680
  680 stop
if (ncf.eq. mxncf) go to 670
  rh = 0.25d0
  isup = miter
  irre = 1
  go to 170

! the correction + - converged. jcnr is set to 0
! to signal that the jacobian involved may need updating later.
! the local error test is made and control passes to statement 500
! if it fails.

540 jcnr = 0
  if (m.eq. 0) dom = d1/escosa(2,nq)
  if (m.gt. 0) dom = vnorn(n, acor, ewt)/escosa(2,nq)
  if (dom.gt. 1.0d0) go to 500

! after a successful step, update the yh array.
! consider changing h if jath is 1; otherwise decrease jath by 1.
! if jath is 1 and nq, it maxed, then acor is saved for
! use in a possible order increase on the next step.
! if a change in h is considered, an increase or decrease in order
! by one is considered also. a change in h is made only if it is by a
! factor of at least 1.1, if not, jath is set to 3 to prevent
! testing for that many steps.

kflag = 0
  irre = 0
  nx = nqts + 1
  hu = h
  nq = nqts
  do 470 j = 1, n
  do 470 i = 1, n
  470 yh(i,j) = yh(i,j) + el(j)*acor(i)
  jath = jath - 1
  if (jath.eq. 0) go to 520
  if (jath.gt. 1) go to 700
  if (1.eq. lmax) go to 700
  do 490 j = 1, n
  490 yh(i,lmax) = acor(i)
  go to 700

! the error test failed. kflag keeps track of multiple failures.
! restore it and the yh array to their previous values, and prepare
! to try the step again. compute the optimum step size for this or
! one lower order. after 2 or more failures, h is forced to decrease
! by a factor of 0.2 or less.

500 kflag = kflag - 1
  tn = tol
  i1 = nqyn히
  do 515 jh = 1, nqyn히
  i1 = i1 - nqyn히
  cdv$ ivdep
do 510 i = 1, nqyn히

510 yh(i) = yh(i1) - yh(i1+nqyn히)
  continue
  nmax = 2.0d0
  if (dahb-h) le. hmin*(1 00001d0) go to 660
  if (kflag.le. -3) go to 640
  irre = 2
  rhup = 0.0d0
  go to 540

! regardless of the success or failure of the step, factors
! rhmin, rhup, and rhup are computed, by which h could be multiplied
! at order nq - 1, order nq, or order nq + 1, respectively.
! in the case of failure, rhup = 0.0 to avoid an order increase.
! the largest of these is determined and the new order chosen
! accordingly. if the order is to be increased, we compute one
! additional scaled derivative

520 rhup = 0.0d0
  if (1.eq. lmax) go to 540
  do 530 i = 1, n
  530 savf(i) = acor(i) - yh(i,lmax)
  dup = vnorn(n, savf, ewt)/escosa(3,nq)
  exup = 1.0d0-
  if (pq.eq. 1) go to 560
  do 500 j = 1, n
  500 yh(j) = yh(j1) + el(j)*acor(j)
  jath = jath - 1
  if (jath.eq. 0) go to 520
  if (jath.gt. 1) go to 700
  if (1.eq. lmax) go to 700
  do 490 j = 1, n
  490 yh(j,lmax) = acor(j)
  go to 700

! the error test failed. kflag keeps track of multiple failures.
! restore it and the yh array to their previous values, and prepare
! to try the step again. compute the optimum step size for this or
! one lower order. after 2 or more failures, h is forced to decrease
! by a factor of 0.2 or less.

500 kflag = kflag - 1
  tn = tol
  i1 = nqyn히
  do 515 jh = 1, nqyn히
  i1 = i1 - nqyn히
  cdv$ ivdep
do 510 i = 1, nqyn히

510 yh(i) = yh(i1) - yh(i1+nqyn히)
  continue
  nmax = 2.0d0
  if (dahb-h) le. hmin*(1 00001d0) go to 660
  if (kflag.le. -3) go to 640
  irre = 2
  rhup = 0.0d0
  go to 540

! regardless of the success or failure of the step, factors
! rhmin, rhup, and rhup are computed, by which h could be multiplied
! at order nq - 1, order nq, or order nq + 1, respectively.
! in the case of failure, rhup = 0.0 to avoid an order increase.
! the largest of these is determined and the new order chosen
! accordingly. if the order is to be increased, we compute one
! additional scaled derivative

520 rhup = 0.0d0
  if (1.eq. lmax) go to 540
  do 530 i = 1, n
  530 savf(i) = acor(i) - yh(i,lmax)
  dup = vnorn(n, savf, ewt)/escosa(3,nq)
  exup = 1.0d0-
  if (pq.eq. 1) go to 560
  do 500 j = 1, n
  500 yh(j) = yh(j1) + el(j)*acor(j)
  jath = jath - 1
  if (jath.eq. 0) go to 520
  if (jath.gt. 1) go to 700
  if (1.eq. lmax) go to 700
  do 490 j = 1, n
  490 yh(j,lmax) = acor(j)
  go to 700
c if there is a change of order, reset eq. 1, and the coefficients.
c in any case h is reset according to rh and the yh array is rescaled.
c then exit from 600 if the step was ok, or redo the step otherwise.
c------------------------------------------------------------
   if (newq .eq. nq) go to 170
630  nq = newq
   l = nq + 1
   ier = 2
   go to 150
   c------------------------------------------------------------
c control reaches this section if 3 or more failures have occurred.
c if 10 failures have occurred, exit with kflag = -1.
c it is assumed that the derivatives that have accumulated in the
   yh array have errors of the wrong order hence the first
c derivative is recomputed, and the order is set to 1. then
   c h is reduced by a factor of 10, and the step is retried.
c until it succeeds or h reaches hmin.
c-----------------------------------------------------------------
640  if (kflag .eq. -10) go to 660
   rh = 0.1d0
   rh = dmax1(hmin/dabs(h),rh)
   h = h*rh
   do 655 i = 1,n
   yh(i,2) = h*svrf(i)
   iipup = iiter
   iah = 5
   if (eq .eq. 1) go to 200
   nq = 1
   i = 2
   ier = 3
   go to 150
   c-----------------------------------------------------------------
c all returns are made through this section. h is saved in hold
c to allow the caller to change h on the next step.
c---------------------------------------------------------------
660  kflag = -1
   go to 720
670  kflag = -2
   go to 720
680  kflag = -3
   go to 720
690  mmax = 10.0d0
700  t = 1.0d0/tesco(2,req)
   do 710 j = 1,n
710  acost(j) = acost(j)**
720  hold = h
   jstart = 1
   return
   c---------------------------------------------------------------
   end of subroutine code ----------------------------------------

---------------------------------------------------------------
double precision function vnorm (n, v, w)
c this function routine computes the weighted root-mean-square norm
   c of the vector of length n contained in the array v, with weights
   c contained in the array w of length n.
c vnorm = sqrt( (1/n)*sum( v(i)*w(i) )**2 )
c-----------------------------------------------------------------
   integer n, i
c double precision v, w, sum
dimension v(n), w(n)
sum = 0.d0
   do 10 i = 1,n
      10  sum = sum + (v(i)**w(i))**2
   vnorm = dsqrt(sum/dfloat(n))
   return
   c---------------------------------------------------------------
   end of function vnorm------------------------------------------
c
   subroutine code (mesh, elco, tesco)
c
   optimize
   integer meth
   integer i, rh, nq, nq1, nq2
c double precision elco, tesco
c double precision agmq, fq, fnq1, pc, pnt, nqg,
c 1 rjqc, jqc, jqc, tspn, spin
dimension elco(13,12), tesco(3,12)
c-----------------------------------------------------------------
c code is called by the integrator routine to set coefficients
   c needed there. the coefficients for the current method, as
   c given by the value of meth, are set for all orders and saved.
c the maximum order assumed here is 12 if meth = 1 and 5 if meth = 2.
c (a smaller value of the maximum order is also allowed )
c code is called once at the beginning of the problem, c
   and is not called again unless and until meth is changed.
c
c the elco array contains the basic method coefficients.
c the coefficients el(i), k = 1, 2, 3, nq-1, for the method of
   c order nq are stored in elco(i,nq). they are given by a generating
c polynomial, i.e.
c i ek(i) = el(i) + el(i+1)*x + ... + el(nq)*x**(nq-1)
c for the implicit Adams methods, f(x) is given by
   c df(x) = f(x) = 1**x**(nq-1)factorial(nq-1), f(x-1) = 0.
c for the bdf methods, f(x) is given by
   c f(x) = (x+1)**x**(nq-1)factorial(nq-1), k = 1, 2, ... 1/nq
   c where
   c factorial(nq) = factorial(nq-1)(1 + 1/2 + ... 1/nq)
c the tesco array contains test constants used for the
   c local error test and the selection of step size and/or order.
c as order nq, tesco(nq) is used for the selection of step
   c size at order nq = 1 if k = 1, at order nq if k = 2, and at order
   c nq = 1 if k = 3.
c dimension pc(12)
c
c the array will contain the coefficients of the polynomial

c \( p(x) = (s+1)^*x*(s+2)*...*(s+\text{ng}-1). \)

c initially, \( p(x) = 1. \)

c form of coefficients of \( p(x)*(s+\text{ng}-1) \)

c \( nq = 0.00 \)
do 110 \( nq = 1.00 \)
do 120 \( x = 1.00 \)
do 130 \( x = 2.00 \)
c store coefficients in elco and tesco.

c \( \text{elco}(\text{ng}) = \text{p}(\text{x}) \cdot \text{elco}(\text{ng}) \)

c \( \text{elco}(\text{ng}) = 1.00 \)
do 140 \( nq = 1.5 \)
c the array will contain the coefficients of the polynomial

c \( p(x) = (s+1)^*x*(s+2)*...*(s+\text{ng}). \)
c where c(i,k) = j*(j-1)*...*(j-k+1), q = ncur, tn = tnow, h = hcur.
c the quantities nz = ncur, l = nz+1, n = nz, tn, and h are
c communicated by common. the above sum is done in reverse order.
c iflag is returned negative if either k or i is out of bounds.
c
1 1, h, lunit, msgf, npcw, nch, nwds
double precision r1, r2
dimension msg(nmes)

subroutines xerrw, xerf, and xetun, as given here, constitute
c a simplified version of the small error handling package
c this version is in double precision.
c
1 c all arguments are input arguments.
c
1 c msg = the message (for errors literal or integer array).
c
1 c nmes = the length of msg (number of characters).
c
1 c nerr = the error number (not used).
c
1 c level = the error level.
c
1 c 0 or 1 means recoverable (control returns to caller).
c
1 c 2 means fatal (run is aborted—see note below).
c
1 c nl = number of integers (0, 1, or 2) to be printed with message.
c
1 c ni, i2 = integers to be printed, depending on ni.
c
1 c nr = number of reals (0, 1, or 2) to be printed with message.
c
1 c r1, r2 = reals to be printed, depending on nr.
c
1 c note: this routine is machine-dependent and specialized for use
1 in limited context, in the following ways.
c
1 1 the number of booleans characters stored per word, denoted
1 by npcw below, is a data-loaded constant.
c
1 1 the value of nmes is assumed to be at most 60.
c
1 c (multi-line messages generated by repeated calls.)
1 3. if level = 2, control passes to the statement stop
1 c to abort the run. this statement may be machine-dependent.
c
1 4. r1 and r2 are assumed to be in double precision and are printed
1 in d21.13 format.
c
1 5. the common block ich00001 below is data-loaded (a machine-
c
1 dependent feature) with default values.
c
1 6. this block is needed for proper retention of parameters used by
1 c this routine which the user may reset by calling xerf or xetun.
c
1 7. the variables in this block are as follows.
c
1 c msgf = print control flag
1 c 1 means print all messages (the default).
c
1 c 0 means no printing.
c
1 lunit = logical unit number for messages.
c
1 c the default is 6 (machine-dependent).
c
1 c the following are instructions for installing this routine
1 in different machine environments.
c
1 c to change the default output unit, change the data statement
1 in the block data subroutine below.
c
1 c for a different number of characters per word, change the
1 c data statement setting npcw below, and format 10. alternatives for
1 c various computers are shown in comment cards.
c
1 c for a different run-abort command, change the statement following
c statement 100 at the end.
c common lch0001/mesfig, lunit
c the following data-loaded value of ncpw is valid for the cdc-6600
and cdc-7600 computers.
c data ncpw/10/
c the following is valid for the cray-1 computer.
c data ncpw/6/
c the following is valid for the burroughs 6700 and 7800 computers.
c data ncpw/5/
c the following is valid for the pdp-10 computer.
c data ncpw/4/
c the following is valid for the vax computer with 4 bytes per integer,
c and for the ibm-360, ibm-370, ibm-303x, and ibm-43xx computers.
c data ncpw/4/
c the following is valid for the pdp-11, or vax with 2-byte integers.
c data ncpw/2/
c if (mesfig .eq. 0) go to 100
get logical unit number.
lun = lunit
c get number of words in message.

   nch = min(60, nmes, 60)
   nwds = nch/ncpw
   if (nch .ne. nwds*ncpw) nwds = nwds + 1

write the message.
write (lun, 10) (msg(i), i=1,nwds)
c write the format statement is to have the form
10 format (1, x, mmnn)
c where nn = ncpw and mm is the smallest integer ge 60/ncpw.
c the following is valid for ncpw = 10.
10 format (1, x, 61a10)
c the following is valid for ncpw = 8.
10 format (1, x, 8a8)
c the following is valid for ncpw = 6.
10 format (1, x, 10a6)
c the following is valid for ncpw = 5.
10 format (1, x, 12a5)
c the following is valid for ncpw = 4.
10 format (1, x, 15a4)
c the following is valid for ncpw = 2.
10 format (1, x, 30x2)
c if (nch .eq. 1) write (lun, 20)
20 format (2, x, 23bin above message, i1 = .i10)
   if (nch .eq. 2) write (lun, 30)
30 format (6, x, 23bin above message, i1 = .i10, x, 3x, 4bi2 = .i10)
   if (nch .eq. 1) write (lun, 40)
40 format (6, x, 23bin above message, i1 = .i10, x, 3x, 4bi2 = .i10)
   if (nch .eq. 2) write (lun, 50)
50 format (x, 3bin above, i1 = .d21.13, x, 4bi2 = .d21.13)
   abort the run if level = 2.

100 if (level .ne. 2) return
stop
end of subroutine errw
end of block data
c this data subprogram loads variables into the internal common
blocks used by the odepack solvers. the variables are
defined as follows.
illn = counter for the number of consecutive times the package
was called with illegal input the run is stopped when
illn reaches s.
ntep = counter for the number of consecutive times the package
was called with bad = 1 and tout = t. the run is
stopped when ntep reaches 5.
mesfig = flag to control printing of error messages. 1 means print,
0 means no printing.
lunit = default value of logical unit number for printing of error
messages.

integer illn, idum, ntep, idumb, lowns, icomm, mesfig, lunit
double precision rnows, rcomm
common /t0001/iown(209), icomm(19),
1 illn, idum(10), ntep, idumb(2), lowns(6), icomm(19)
common /lch001/mesfig, lunit
data ilin(9), ntep(9)
data mesfig(1), lunit(9)
c end of block data

end subroutine cewset ( n, inol, rtol, atol, yeur, ewt)
cell optimum

integer n, inol
integer i
double precision rtol, atol, yeur, ewt
dimension rtol(1), atol(1), yeur(n), ewt(n)
c go to (10, 20, 30, 40), inol
10 continue
   do 3 i = 1, n
15 ew(ei) = rtol(1)*dabs(yeur(ei)) + atol(1)
   return
20 continue
   do 3 i = 1, n
25 ew(ei) = rtol(1)*dabs(yeur(ei)) + atol(1)
   return
30 continue
   do 35 i = 1, n
35  ewt(i) = rtol(i)*dabs(yewt(i)) + atol(i)
return
40  continue
45  ewt(i) = rtol(i)*dabs(yewt(i)) + atol(i)
return
end

end subroutine ewset

C***********************************************************************

subroutine dgefa(a,lda,n,ipvt,info)
integer lda,n,ipvt(1),info
double precision a(lda,1)

C dgefa factors a double precision matrix by gaussian elimination.
C dgefa is usually called by dgeco, but it can be called
C directly with a saving in time if rcond is not needed.
C (time for dgeco) = (1 + 9n^2)*time for dgefa).
C
C on entry
C a double precision(lda, n)
C the matrix to be factored.
C lda integer
C the leading dimension of the array a .
C n integer
C the order of the matrix a .
C
C on return
C a an upper triangular matrix and the multipliers
C which were used to obtain it.
C the factorization can be written a = L*U where
C L is a product of permutation and unit lower
C triangular matrices and U is upper triangular.
C ipvt integer(n)
C an integer vector of pivot indices.
C info integer
C = 0 normal value.
C = k if u(k,k), eq. 0.0 , this is not an error
C condition for this subroutine, but it does
C indicate that dgesl or dgeci will divide by zero
C if called. use resd in dgeco for a reliable
C indication of singularity.
C
C linpack, this version dated 12/14/78.
C cleve moier, university of new mexico, argonne national lab.
C
subroutines and functions

blas daxpy, dscal, idamax
C internal variables
C double precision t
integer idamax,j,k,kp1,l,nml

C gaussian elimination with partial pivoting
C
info = 0
nml = n - 1
if (nml .lt. 1) go to 70
do 60 k = 1, nml
kp1 = k + 1
C find l = pivot index
l = idamax(n-k+1,a(k,k),1) + k - 1
ipvt(k) = l
C zero pivot implies this column already triangularized
if (a(l,k)) eq. 0.0d0) go to 40
C interchange if necessary
if (l .eq. k) go to 10
l = a(l,k)
a(k,l) = a(k,k)
at(k,l) = t
10 continue
C compute multipliers
C t = -1.0d0/a(k,k)
call dscal(-1.0d0,a(k+1,k),1)
C row elimination with column indexing
C
do 30 j = kp1, n
C t = a(j,k)
if (l .eq. k) go to 20
at(j) = a(k,j)
at(k,j) = t
20 continue
C call daxpy(-1.0d0,a(k+1,k),1,a(k+1,j),1)
30 continue
C go to 50
40 continue
info = k
50 continue
60 continue
70 continue
ipvt(n) = n
if (ai(n,n) .eq. 0.0d0) info = n
  return
end

subroutine daxyy(n,da,dx,incy)

  c constant times a vector plus a vector.
  c uses unrolled loops for increments equal to one.
  c jack dongarra, linpack, 3/11/78.
  c modified 12/9/93, array(1) declarations changed to array(*)
  c
  double precision dx(*),dy(*),da
  integer i,incy,ia,iy,im,nmpl,n
  c
  if(n .le. 0) return
  if(da .eq. 0.0d0) return
  if(incy .eq. 1 .and. incy .eq. 1) go to 20
  c
  c code for unequal increments or equal increments
  c not equal to 1
  c
  is = 1
  iy = 1
  if(incy .eq. 0) x = (-n+1)*incy + 1
  if(incy .eq. 0) y = (-n+1)*incy + 1
  do 10 i = 1,n
      dy(i) = dy(i) + da*dx(i)
      ix = iy + incy
      iy = iy + incy
  10 continue
  return
  c
  c code for both increments equal to 1
  c
  c clean-up loop
  c
  20 m = mod(n,5)
  if (m .eq. 0) go to 40
  do 30 i = 1,m
      dy(i) = dy(i) + da*dx(i)
  30 continue
  c
  40 mpl = n .le. 4) return
  c
  do 50 i = mpl,n,n
      dy(i) = dy(i) + da*dx(i)
      dy(i + 1) = dy(i + 1) + da*dx(i + 1)
      dy(i + 2) = dy(i + 2) + da*dx(i + 2)
      dy(i + 3) = dy(i + 3) + da*dx(i + 3)
  50 continue
  return
end

subroutine dscal(n,da,dx,incy)

c scales a vector by a constant

c uses unrolled loops for increment equal to one.

c jack dongarra, linpack, 3/11/78.

c modified 3/93 to return if incx .le. 0

c modified 12/9/93, array(1) declarations changed to array(*)

c double precision dx(*),dy(*)
  integer i,incy,ia,iy,m,nmpl,n
  c
  if(n .le. 0 .or. incx .le. 0) return
  if(incy .eq. 1) go to 20
  c
  c code for increment not equal to 1
  c
  ninx = n*incy
  do 10 i = 1,ninx,incy
      dx(i) = da*dx(i)
  10 continue
  return
  c
  c code for increment equal to 1
  c
  c clean-up loop
  c
  20 m = mod(n,5)
  if (m .eq. 0) go to 40
  do 30 i = 1,m
      dx(i) = da*dx(i)
  30 continue
  c
  40 mpl = n .le. 4) return
  c
  do 50 i = mpl,n,n
      dx(i) = da*dx(i)
      dx(i + 1) = da*dx(i + 1)
      dx(i + 2) = da*dx(i + 2)
      dx(i + 3) = da*dx(i + 3)
  50 continue
  return
end

integer function idamax(n,dx,incy)

  c finds the index of element having max. absolute value
  c jack dongarra, linpack, 3/11/78.
  c modified 3/93 to return if incx .le. 0
  c modified 12/9/93, array(1) declarations changed to array(*)
  c
double precision dx(*),dmax
  integer incx,i
  c
  idamax = 0
  if n .le. 1 .or. incx .le. 0) return


```
idamax = 1
if(n.eq.1) return
if(ninc eq 1) go to 20

code for increment not equal to 1

ix = 1
imax = dabs(dx(1))
ix = ix + incx
do 10 i = 2,n
if(idabs(dx(i)) le dmax) go to 5
imax = i
5 ix = ix + incx
10 continue
return

code for increment equal to 1

20 dmax = dabs(dx(1))
do 30 i = 2,n
if(idabs(dx(i)) le dmax) go to 30
imax = i
30 continue
return

*****************************************************************************
subroutine dgbsl(abd,lda,n,mu,ipvt,job)
integer lda,n,mu,ipvt,job
double precision abd(lda,n),mu

C dgbis solves the double precision band system
C a * x = b or trans(a) * x = b
C using the factors computed by dgbc0 or dgbf.
C
on entry
C abd double precision(lda,n)
C the output from dgbc0 or dgbf.
C
lda integer
C the leading dimension of the array abd.
C
n integer
C the order of the original matrix.
C
mu integer
C number of diagonals below the main diagonal.
C
ipvt integer(n)
C
C the pivot vector from dgbc0 or dgbf.
C
b double precision(n)
C the right hand side vector.
C
c job integer
C = 0 to solve a * x = b,
C = nonzero to solve trans(a) * x = b, where
C trans(a) is the transpose.
C
on return
C
b the solution vector x.
C
e error condition
C
a division by zero will occur if the input factor contains a
C zero on the diagonal. technically this indicates singularity
C but it is often caused by improper arguments or improper
C setting of lda. it will not occur if the subroutines are
C called correctly and if dgbf has set info.eq. 0.
C to compute inverse(a) * c where c is a matrix
C with p columns
C call dgbsl(abd,lda,n,mu,ipvt,job)
C if (cond is too small) go to...
C do 10 j = 1, p
C call dgbis(abd,lda,n,mu,ipvt,j,0)
C 10 continue
C
C linpack version 05/14/78
C cleve moler, university of new mexico, argonne national lab.
C
C subroutines and functions
C
C blast daxpy,ddot
C fortran min0
C
C internal variables
C
double precision ddot,t
C integer k,kb,lb,ll,lm,mm,mn,mm1
C
mn = mu + ml + 1
C mn1 = n - 1
C if (job ne 0) go to 50
C
job = 0., solve a * x = b
C first solve ly = b
C if (mn eq 0) go to 30
C if (mn1 lt 1) go to 30
C do 20 k = 1, mn1
```
l = \min(0,m-l+1,k)
1 = \text{ipvt}(k)
t = b(1)
if (l \leq k) go to 10
\text{b}(1) = b(1)
\text{b}(k) = 1
continue

call daxpy(lm,la,ab(lab+1,k),1,b(k+1),1)
20 continue
30 continue
c c now solve \text{u}^*x = y

do 40 \text{kb} = 1, n
k = n + 1 - \text{kb}
b(k) = b(k) \cdot \text{ab}(m,k)
lm = \min(0,k,m) - 1
la = m - \text{lm}
lb = k - \text{lm}
t = -b(k)
call daxpy(lm,la,ab(la,k),1,b(lb),1)
40 continue
go to 100
50 continue
c c job = nonzero, solve \text{trans}(a)^*x = b
c c first solve \text{trans}(u)^*y = b
do 60 \text{k} = 1, n
\text{lm} = \min(0,k,m) - 1
la = m - \text{lm}
\text{lb} = k - \text{lm}
t = \text{ddot}(\text{lm},\text{lab}(la,k),1,b(lb),1)
b(k) = (b(k) - t)/\text{ab}(m,k)
60 continue
c c now solve \text{trans}(l)^*x = y
c if (m \leq l) go to 90
if (nm1 \leq l) go to 90
\text{do 80 k} = 1, nm1
\text{k} = n - \text{kb}
lm = \min(\text{mn},m-n,k)
b(k) = b(k) + \text{ddot}(\text{lm},\text{ab}(m+l,k),1,b(k+l),1)
l = \text{ipvt}(k)
if (l \leq k) go to 70
\text{t} = b(1)
b(1) = b(k)
b(k) = t
70 continue
80 continue
90 continue
100 continue
return

c end

\text{subroutine daxpy}(a,dx,incx,dy,incy)
c c constant times a vector plus a vector.
c c uses unrolled loops for increments equal to one.
c c jack Dongarra, Linpack, 3/10/78

\text{c \text{modified 12/30/83, array(1) declarations changed to array(*)}}
c c double precision dx(*),dy(*),da
c \text{integer incx,incy,ix,iy,im,mp1,n}
c c if im = 0 return
\text{if (dx eq 0.0) return}
\text{if (incx eq 1 and incy eq 1) go to 20}
c c c code for unequal increments or equal increments
\text{not equal to 1}
c c c
ix = 1
iy = 1
\text{if (incx lt 0) \text{Dix} = (n+1)*incx + 1}
\text{if (incy lt 0) \text{Diy} = (n+1)*incy + 1}
d0 10 i = 1:n
\text{dy(iy) = dy(iy) + da*dx(i)}
ix = ix + incx
iy = iy + incy
10 continue
return

\text{c c code for both increments equal to 1}
c c c
20 m = mod(n,4)
\text{if (m eq 0) go to 40}
do 30 i = 1:m
\text{dy(i) = dy(i) + da*dx(i)}
30 continue
\text{if n \leq 4) return}
\text{do 40 mp1 = m + 1}
d0 50 i = mp1,n,4
\text{dy(i) = dy(i) + da*dx(i)}
\text{dy(i+1) = dy(i+1) + da*dx(i+1)}
\text{dy(i+2) = dy(i+2) + da*dx(i+2)}
\text{dy(i+3) = dy(i+3) + da*dx(i+3)}
50 continue
return
end
\text{double precision function ddotn(dx,incx,dy,incy)}
c c forms the dot product of two vectors.
c c uses unrolled loops for increments equal to one.
c c jack Dongarra, Linpack, 3/11/78
double precision dx(*),dy(*),demp
integer i,nx,nx,y,x,y,m,n

dot = 0.000

demp = 0.000

if (n < 0.0) return
if (nx.eq.1 and nxy.eq.1) go to 20

code for unequal increments or equal increments
not equal to 1

ix = 1
iy = 1

if (nxy.lt.0.0) nx = (-nx+1)*nx + 1
if (nxy.gt.0.0) ny = (-ny+1)*ny + 1

do 10 i = 1,m

demp = demp + dx(i)*dy(i)

ix = ix + nx

iy = iy + ny
10 continue

dot = demp
return

code for both increments equal to 1

clean-up loop

m = m + mod(n,5)

if (m < 0.0) go to 40

do 30 j = 1,m

demp = demp + dx(i)*dy(i)
30 continue

if (n > 0.5) go to 60

dmp = demp + dx(i)*dy(i) + dx(i+1)*dy(i) + 1

* dx(i+2)*dy(i) + dx(i+3)*dy(i) + 3

* dx(i+4)*dy(i) + 4

50 continue

dot = demp
return
end

******************************************************************************

 subroutine dgesl(la,lda,n,ipvt,b,job)
 integer lda,n,ipvt,job
 double precision a(lda,1),b(1)

dgesl solves the double precision system

 a*x = b or trans(a)*x = b

 using the factors computed by dgeco or dgefa.

 on entry

 a double precision(lda, n)

the output from dgeco or dgefa.

 lda integer

 the leading dimension of the array a.

 n integer

 the order of the matrix a.

 ipvt integer(n)

 the pivot vector from dgeco or dgefa.

 b double precision(n)

the right hand side vector.

 job integer

 = 0 to solve a*x = b,

 = nonzero to solve trans(a)*x = b where

 trans(a) is the transpose.

 on return

 b the solution vector x.

 error condition

 a division by zero will occur if the input factor contains a
 zero on the diagonal technically this indicates singularity
 but it is often caused by improper arguments or improper
 setting of lda . it will not occur if the submatrices are
called correctly and if dgeco has set rcond.gt.0.0
 or dgefa has set info.eq.0.

to compute inverse(a)*c where c is a matrix
with p columns

call dgeco(lda,n,ipvt,rcond,x)

 if (rcond is too small) go to ...

do 10 j = 1,p

call dgesl(lda,lda,n,ipvt,c(j,0))

10 continue

 linpack, this version dated 08/14/78

clever solver, university of new mexico, argonne national lab.

subroutines and functions

 blas daxpy,dot

 internal variables

 double precision ddot,d

 integer k,kb,l,nml
nnl = n - 1
if (job.ne 0) go to 50

job = 0, solve a*x = b
first solve F'y = b

if (nnl .lt. 1) go to 30
   do 20 k = 1, nnl
      l = iplv(k)
      t = -b(l)
      if (l .eq. k) go to 10
      b(l) = b(k)
      b(k) = t
   10 continue
   call daxpy(n-k,La(k+1,k),1,b(k+1),1)
   20 continue
30 continue

now solve u*s = y

do 40 kb = 1, n
   k = n - kb
   b(k) = b(k)*La(k,k)
   t = -b(k)
   call daxpy(1,La(1,k),1,b(1),1)
40 continue
50 continue

job = nonzero, solve trans(a)*x = b
first solve trans(a)*y = b

if (nnl .lt. 1) go to 90
   do 80 k = 1, nnl
      l = iplv(k)
      t = -ddot(k,La(1,k),1,b(1),1)
      b(k) = (b(k) - t)*a(k,k)
   80 continue
90 continue

now solve trans(l)*x = y

if (nnl .lt. 1) go to 70
   do 70 k = 1, nnl
      l = iplv(k)
      t = b(l)
      b(l) = b(k)
      b(k) = t
   70 continue
80 continue
90 continue
100 continue
return
end

subroutine daxpy(n,da,ds,incx,dy,incy)
c
constant times a vector plus a vector
c
uses unrolled loops for increments equal to one.
c
jack dongarra, linpack, 3/11/78.
c
modified 12/30/79, array(1) declarations changed to array(*)
c
double precision ds(*),dy(*),da
c
integer i,incx,incy,ix,iy,m,mp1,n

c
if(n.le.0) return
if (da eq 0.0d0) return
if(incy eq 1 and incx eq 1) go to 20

c
code for unequal increments or equal increments not equal to 1

c
ix = 1
iy = 1
dy(i) = dy(i) + da*dx(i)
ix = ix + incx
iy = iy + incy
10 continue
20 return

c
code for both increments equal to 1

c
clean-up loop

20 m = mod(n,4)
if m eq 0 i = 1
   go to 40
   do 30 i = 1,m
       dy(i) = dy(i) + da*dx(i)
30 continue
40 m = m + 1
   do 50 i = m,nnl
       dy(i) = dy(i) + da*dx(i)
       dy(i+1) = dy(i+1) + da*dx(i+1)
       dy(i+2) = dy(i+2) + da*dx(i+2)
       dy(i+3) = dy(i+3) + da*dx(i+3)
50 continue
60 return
end
double precision function ddot(n,ds,incx,dy,incy)
c
forms the dot product of two vectors
c
uses unrolled loops for increments equal to one.
c
jack dongarra, linpack, 3/11/78
c
modified 12/30/79, array(1) declarations changed to array(*)
double precision dx(*),dy(*),dtemp
integer incx,incy,iy,iy,mp1,n

ddot = 0.0d0
dtemp = 0.0d0
if (n.le.0) return
if (incx.eq.1.and.incy.eq.1) go to 20

  code for unequal increments or equal increments not equal to 1
li = 1
iy = 1
if (incx.eq.1) li = (-n+1)*incx + 1
if (incy.eq.1) iy = (-m+1)*incy + 1
do 10 i = 1,n
dtemp = dtemp + dx(i)*dy(i)
ix = ix + incx
iy = iy + incy
10 continue
ddot = dtemp
return

  code for both increments equal to 1
  
  clean-up loop
20 m = mod(n,5)
   if (m.eq.0) go to 40
do 30 i = 1,m
      dtemp = dtemp + dx(i)*dy(i)
30 continue
   if (n.le.5) go to 60
40 mp1 = m + 1
do 50 i = mp1,n,5
      dtemp = dtemp + dx(i)*dy(i) + dx(i + 1)*dy(i + 1) +
               dx(i + 2)*dy(i + 2) + dx(i + 3)*dy(i + 3) + dx(i + 4)*dy(i + 4)
50 continue
60 ddot = dtemp
return

*------------------------------------------------------------------
subroutine dghfa(abd,lda,n,ml,nu,ipvt,info)
integer lda,n,ml,nu,ipvt(1),info
double precision abd(lda,1)

dghfa factors a double precision band matrix by elimination

  dghfa is usually called by dpheco, but it can be called
directly without saving in time if recond is not needed

c on entry

c on exit

  abd  double precision(lda, n)
        contains the matrix in band storage, the columns
        of the matrix are stored in the columns of abd and
        the diagonals of the matrix are stored in rows
        ml+1 through 2*ml+nu+1 of abd
        see the comments below for details.

  lda integer
        the leading dimension of the array abd.
        lda must be ge 2*ml + nu + 1.

  n integer
        the order of the original matrix.

  ml integer
        number of diagonals below the main diagonal.
        0 le ml lt n.

  nu integer
        number of diagonals above the main diagonal.
        0 le nu lt n.
        more efficient if ml le nu.
    on return

  abd  an upper triangular matrix in band storage
        and the multipliers which were used to obtain it.
        the factorization can be written as a = l' u
        where l is a product of permutations and unit lower
        triangular matrices and u is upper triangular.

  ipvt integer(n)
        an integer vector of pivot indices.

  info integer
    = 0 normal value
    = k if ul(k,k) eq 0.0, this is not an error
      condition for this subroutine, but it does
      indicate that dpheco will divide by zero if
    called use recond in dpheco for a reliable
    indication of singularity.

  band storage
    if  a  is a band matrix, the following program segment
    will set up the input.

    ml = (band width below the diagonal)
    nu = (band width above the diagonal)
    m = ml + nu + 1
    do 20 j = 1,n
       i1 = max(0, j-ml)
       i2 = min(n, j-nu)
       abd(i1:i2, j)
```c
k = l - j + m
abd(k,j) = a(k,j)
10 continue
20 continue

this uses rows ml+1 through 2*ml+mu+1 of abd.
in addition, the first ml rows in abd are used for
elements generated during the triangulization.
the total number of rows needed in abd is 2*ml+mu+1.
the ml+mu by ml+mu upper left triangle and the
ml by ml lower right triangle are not referenced.

limpack: this version dated 08/14/78.
cleve moler, university of new mexico, argonne national lab.
c
subroutines and functions
cblas daxpy, dscal, idamax
cfortran max0, min0
c
internal variables
c
double precision t
integer i, idamax, i0, j0, j, jk, jk1, k, kp1, l, lm, mm, mm1

m = ml + mu + 1
info = 0

c zero initial fill-in columns

c j0 = mu + 2
j1 = min(n, m) - 1
if (j1 > j0) go to 30
do 20 jz = j0, j1
   if (jz < 0) go to 60
   do 10 i = j0, m
      abd(i,jz) = 0.0d0
   10 continue
20 continue
30 continue
40 jx = j1
50 ju = 0

60 gaussian elimination with partial pivoting

70 nm1 = n - 1
90 if (nm1 > lt) go to 130
   do 120 k = 1, nm1
         kp1 = k + 1
   120 continue

110 zero next fill-in column
120 jz = jx + 1
130 if (jz > n) go to 50
140 if (ml > lt) go to 50
150 do 40 i = 1, ml
   abd(i,jz) = 0.0d0
40 continue
50 continue

find l = pivot index

lm = min0(ml, n-k)
l = idamax(lm+1, abd(m, k), 1) + m - 1
ipvt(k) = l + k - m

zero pivot implies this column already triangularized

if (abd(l, k) eq 0.0d0) go to 100

interchange if necessary

if (l eq ml) go to 60
   t = abd(l, k)
   abd(l, k) = abd(m, k)
   abd(m, k) = t
60 continue
70 continue
80 compute multipliers

    t = -1.0d0*abd(l, k)
    call dscal(lm, t, abd(m-1, k), 1)
90 continue

row elimination with column indexing

   ju = min0(max0(ju, mu+ipvt(l, k)), n)
mn = m
if (ju > kp1) go to 90
   do 80 ji = kp1, ju
      l = l - 1
      mm = mn - l
      t = abd(l, ji)
      if (l eq mm) go to 70
         abd(ji, ji) = abd(mm, ji)
         abd(mm, ji) = t
70 continue
80 continue
90 continue
go to 110
100 continue
110 info = k
120 continue
130 info = n
140 continue
```

return
end subroutine dsyxpy(n,da,dx,incy,dy,jncy)

constant times a vector plus a vector.
uses unrolled loops for increments equal to one.

modified 12/3/93, array(1) declarations changed to array(*)

double precision dx(*),dy(*),da
integer i,incy,ix,iy,inx,n,mpi,n

if(n le 0) return
if(da .eq. 0.0) return
if(incy .eq. 1 and inx.eq.1) go to 20

code for unequal increments or equal increments
not equal to 1

ix = 1
iy = 1
if(n .lt. 0) ix = (-n+1)*incy + 1
if(n .gt. 0) iy = (-n+1)*incy + 1
do 10 i = 1,n
   dy(i) = dy(i) + da*dx(ix)
   ix = ix + inx
   iy = iy + incy
10 continue
return

code for both increments equal to 1
clean-up loop

20 m = mod(n,4)
if(m .eq. 0) go to 40
   do 30 i = 1,m
      dx(i) = da*dx(i)
30 continue
if(n .lt. 4) return

40 mpi = m + 1
   do 50 i = mpi,n,4
      dy(i) = dy(i) + da*dx(i)
      dy(i + 1) = dy(i + 1) + da*dx(i + 1)
      dy(i + 2) = dy(i + 2) + da*dx(i + 2)
      dy(i + 3) = dy(i + 3) + da*dx(i + 3)
50 continue
return
end subroutine dscale(n,da,dx,incy)
scales a vector by a constant.
uses unrolled loops for increment equal to one.
C

integer by default (sometimes called integer*2), you may need to

to change integer to integer*4 or otherwise instruct your compiler

to use full-word integers in the next 5 declarations.

c

c

comments just before the end statement (lines starting with *)

c

c
c

give c source for dimach

c

integer small(2)

integer large(2)

integer right(2)

integer diver(2)

integer log10(2)

integer sc

c

save small, large, right, diver, log10, sc

c

c

double precision dimach(5)

c

equivalence (dimach(1),small(1))

equivalence (dimach(2),large(1))

equivalence (dimach(3),right(1))

equivalence (dimach(4),diver(1))

equivalence (dimach(5),log10(1))

c

machine constants for big-endian ieee arithmetic (binary format)

machines in which the most significant byte is stored first.

such as the at&t 3b series, motorola 68000 based machines (e.g.

sun 3), and machines that use sparce, hp, or ibm risc chips.

c

data small(1),small(2) / 1048576, 0 /
data large(1),large(2) / 2146435071, -1 /
data right(1),right(2) / 1017118720, 0 /
data diver(1),diver(2) / 1018167296, 0 /
data log10(1),log10(2) / 1070810131, 1352628735 /, sc987/
c

machine constants for little-endian (binary) ieee arithmetic

machines in which the least significant byte is stored first.

e.g. ibm pcc and other machines that use intel 80x87 or dec

alpha chips.

c

data small(1),small(2) / 0, 1048576 /
data large(1),large(2) / -1, 2146435071 /
data right(1),right(2) / 0, 1017118720 /
data diver(1),diver(2) / 1018167296 /
data log10(1),log10(2) / 1352628735, 1070810131 /, sc987/
c

machine constants for amdael, machines.

c

data small(1),small(2) / 1048576, 0 /
data large(1),large(2) / 2147483647, -1 /
data right(1),right(2) / 1352628735, 1352628735 /, sc987/

c

machine constants for amdael, machines.

c

data small(1),small(2) / 1048576, 0 /
data large(1),large(2) / 2147483647, -1 /
data right(1),right(2) / 1352628735, 1352628735 /, sc987/

c

machine constants for amdael, machines.
DATA LOG10(1) / 3777746202341175717B /
DATA LOG10(2) / 0000037514217225465B /, SC987/
MACHINE CONSTANTS FOR THE DATA GENERAL ECLIPSE 5200
SMALL, LARGE, RIGHT, DIVER, LOG10 SHOULD BE DECLARED
INTEGER SMALL(4), LARGE(4), RIGHT(4), DIVER(4), LOG10(4)
NOTE: IT MAY BE APPROPRIATE TO INCLUDE THE FOLLOWING LINE
STATIC DMACH(S)
DATA SMALL/20K,3*0/LARGE/3777777K,3*1777777K/
DATA RIGHT/142K,3*0/DIVER/202K,3*0/
DATA LOG10/4042K,2402K,5023K,74716K/, SC987/
MACHINE CONSTANTS FOR THE HARRIS SLASH 6 AND SLASH 7
DATA SMALL(1), SMALL(2) / '20000000, '00000000 /
DATA LARGE(1), LARGE(2) / '37777777, '37777777 /
DATA RIGHT(1), RIGHT(2) / '20000000, '00000333 /
DATA DIVER(1), DIVER(2) / '20000000, '00000334 /
DATA LOG10(1), LOG10(2) / '23210115, '10237777 /, SC987/
MACHINE CONSTANTS FOR THE HONEYWELL DPS 8/70 SERIES.
DATA SMALL(1), SMALL(2) / 049204000000, 000000000000 /
DATA LARGE(1), LARGE(2) / 076771777777, 077777777777 /
DATA RIGHT(1), RIGHT(2) / 064400000000, 000000000000 /
DATA DIVER(1), DIVER(2) / 064400000000, 000000000000 /
DATA LOG10(1), LOG10(2) / 077646202424, 011571775714 /, SC987/
MACHINE CONSTANTS FOR THE IBM 360/370 SERIES,
THE XEROX SIGMA 5/6 AND THE SEL. SYSTEMS 85/86.
DATA SMALL(1), SMALL(2) / Z0100000, Z00000000 /
DATA LARGE(1), LARGE(2) / Z7FF77777777, Z7FF77777777 /
DATA RIGHT(1), RIGHT(2) / Z33100000, Z00000000 /
DATA DIVER(1), DIVER(2) / Z33100000, Z00000000 /
DATA LOG10(1), LOG10(2) / Z411344313, Z5097797F /, SC987/
MACHINE CONSTANTS FOR THE INTERDATA 8/22
WITH THE UNIX SYSTEM FORTRAN 77 COMPILER.
FOR THE INTERDATA FORTRAN VII COMPILER REPLACE
THE Z'S SPECIFYING HEX CONSTANTS WITH Y'S.
DATA SMALL(1), SMALL(2) / Z'OO100000', Z'00000000 /
DATA LARGE(1), LARGE(2) / Z'7F7777777777', Z'7F7777777777 /
DATA RIGHT(1), RIGHT(2) / Z'33100000', Z'00000000 /
DATA DIVER(1), DIVER(2) / Z'33100000', Z'00000000 /
DATA LOG10(1), LOG10(2) / Z'431344313', Z'5097797F' /, SC987/
MACHINE CONSTANTS FOR THE PDP-10 (KA PROCESSOR)
DATA SMALL(1), SMALL(2) / '033400000000, '000000000000 /
DATA LARGE(1), LARGE(2) / '377777777777, '344444444444 /
DATA RIGHT(1), RIGHT(2) / '113400000000, '000000000000 /
DATA DIVER(1), DIVER(2) / '114400000000, '000000000000 /
DATA LOG10(1), LOG10(2) / '17746202324, '144117571776 /, SC987/
MACHINE CONSTANTS FOR THE PDP-11 (K1 PROCESSOR)
DATA SMALL(1), SMALL(2) / '000400000000, '000000000000 /
DATA LARGE(1), LARGE(2) / '377777777777, '377777777777 /
DATA RIGHT(1), RIGHT(2) / '101400000000, '000000000000 /
DATA DIVER(1), DIVER(2) / '104400000000, '000000000000 /
DATA LOG10(1), LOG10(2) / '17746202324, '0746747777774 /, SC987/
MACHINE CONSTANTS FOR PDP-11 FORTRAN SUPPORTING
32-BIT INTEGERS (EX PressED IN INTEGER AND OCTAL).
DATA SMALL(1), SMALL(2) / 8388608, 0 /
DATA LARGE(1), LARGE(2) / 2147483647, -1 /
DATA RIGHT(1), RIGHT(2) / 612345678, 0 /
DATA DIVER(1), DIVER(2) / 620150992, 0 /
DATA LOG10(1), LOG10(2) / 1076765498, -206387208 /, SC987/
DATA SMALL(1), SMALL(2) / 000040000000, 000000000000 /
DATA LARGE(1), LARGE(2) / 017777777777, 037777777777 /
DATA RIGHT(1), RIGHT(2) / 004400000000, 000000000000 /
DATA DIVER(1), DIVER(2) / 004400000000, 000000000000 /
DATA LOG10(1), LOG10(2) / 007746202424, 0204767477770 /, SC987/
MACHINE CONSTANTS FOR PDP-11 FORTRAN SUPPORTING
16-BIT INTEGERS (E Xpressed IN INTEGER AND OCTAL).
SMALL, LARGE, RIGHT, DIVER, LOG10 SHOULD BE DECLARED
INTEGER SMALL(4), LARGE(4), RIGHT(4), DIVER(4), LOG10(4)
DATA SMALL(1), SMALL(2) / 128, 0 /
DATA SMALL(3), SMALL(4) / 0, 0 /
DATA LARGE(1), LARGE(2) / 12767, -11 /
DATA LARGE(3), LARGE(4) / -1, -11 /
DATA RIGHT(1), RIGHT(2) / 9344, 0 /
DATA RIGHT(3), RIGHT(4) / 0, 0 /
DATA DIVER(1), DIVER(2) / 9472, 0 /
DATA DIVER(3), DIVER(4) / 0, 0 /
DATA LOG10(1), LOG10(2) / 16282, 0346 /
DATA LOG10(3), LOG10(4) / -31493, -12296 /, SC987/
DATA SMALL(1), SMALL(2) / 0000200, 0000000 /
DATA SMALL(3), SMALL(4) / 0000000, 0000000 /
DATA LARGE(1), LARGE(2) / 027777, 0177777 /
C DATA LARGE(3), LARGE(4) / 0177777, 0177777 /
C DATA RIGHT(1), RIGHT(2) / 0022200, 0000000 /
C DATA RIGHT(3), RIGHT(4) / 0000000, 0000000 /
C DATA DIVER(1), DIVER(2) / 0022400, 0000000 /
C DATA DIVER(3), DIVER(4) / 0000000, 0000000 /
C DATA LOG0(1), LOG0(2) / 0037802, 0020232 /
C DATA LOG1(3), LOG1(4) / 0024739, 0147777, 0000000 /
C MACHINE CONSTANTS FOR THE PRIME 50 SERIES SYSTEMS WITH 32-BIT INTEGERS AND 64-VAL MODE INSTRUCTIONS,
SUPPLIED BY IGOR BREY.
C DATA SMALL(1), SMALL(2) / 0000000000, 00000100000 /
C DATA LARGE(1), LARGE(2) / 1777777777, -3777777777 /
C DATA SMALL(1), LARGE(2) / -10000000000, 0000000000 /
C DATA DIVER(1), DIVER(2) / 0000000000, 0000000000 /
C DATA LOG0(1), LOG0(2) / 1450416501, 0000000000 /
C MACHINE CONSTANTS FOR THE SEQUENT BALANCE 8000
C DATA SMALL(1), SMALL(2) / 0000000000, 0000000000 /
C DATA LARGE(1), LARGE(2) / 0000000000, 0000000000 /
C DATA SMALL(1), LARGE(2) / 3777777777, -3777777777 /
C DATA SMALL(1), LARGE(2) / 10000000000, 0000000000 /
C DATA DIVER(1), DIVER(2) / 1150416501, 0000000000 /
C DATA LOG0(1), LOG0(2) / 0037802, 0020232 /
C MACHINE CONSTANTS FOR THE UNIVAC 1100 SERIES.
C DATA SMALL(1), SMALL(2) / 0000000000, 0000000000 /
C DATA LARGE(1), LARGE(2) / 0000000000, 0000000000 /
C DATA SMALL(1), LARGE(2) / 0000000000, 1270000000 /
C DATA LARGE(1), LARGE(2) / 0000000000, 0000000000 /
C DATA DIVER(1), DIVER(2) / 0000000000, 0000000000 /
C DATA LOG0(1), LOG0(2) / 0037802, 0020232 /
C MACHINE CONSTANTS FOR THE VAX UNIX P77 COMPILER
C DATA SMALL(1), SMALL(2) / 128, 0 /
C DATA LARGE(1), LARGE(2) / -128, 0 /
C DATA RIGHT(1), RIGHT(2) / 9344, 0 /
C DATA DIVER(1), DIVER(2) / 54979738, 0 /
C DATA LOG0(1), LOG0(2) / -1370000000 /
C MACHINE CONSTANTS FOR THE VAX-11 WITH FORTRAN IV-PLUS COMPILER
C DATA SMALL(1), SMALL(2) / 000000000, 000000000 /
C DATA LARGE(1), LARGE(2) / 000000000, 000000000 /
C DATA SMALL(1), LARGE(2) / -128, 0 /
C DATA RIGHT(1), RIGHT(2) / 9344, 0 /
C DATA DIVER(1), DIVER(2) / 54979738, 0 /
C DATAN LOG0(1), LOG0(2) / 0037802, 0020232 /
C C MACHINE CONSTANTS FOR VAX/VMS VERSION 2.2
C C DATA SMALL(1), SMALL(2) / '0'X, '0'X /
C DATA LARGE(1), LARGE(2) / 'FFFFF7FF'X, 'FFFFF7FF'X /
C DATA RIGHT(1), RIGHT(2) / '280'X, '280'X /
C DATA DIVER(1), DIVER(2) / '280'X, '280'X /
C DATA LOG0(1), LOG0(2) / '280'X, '280'X /
C C *** ISSUE STOP 72 IF ALL DATA STATEMENTS ARE COMMENTED...
C IF (SNCK .EQ. 987) THEN
C DMACH(1) = 1.103
C IF (SMALL(1) .EQ. 111792533)
* AND SMALL(2) .EQ. -448790528) THEN
* *** IEEE BIG ENDIAN ***
C SMALL(1) = 1048576
C SMALL(2) = 0
C LARGE(1) = 2146435071
C LARGE(2) = -1
C RIGHT(1) = 107118720
C RIGHT(2) = 0
C DIVER(1) = 107118720
C DIVER(2) = 0
C LOG0(1) = 107118720
C LOG0(2) = 1352628735
C ELSE IF (SMALL(2) .EQ. 111792533)
* AND SMALL(1) .EQ. -448790528) THEN
* *** IEEE LITTLE ENDIAN ***
C SMALL(1) = 0
C SMALL(2) = 2146435071
C LARGE(1) = -1
C RIGHT(1) = 107118720
C RIGHT(2) = 0
C DIVER(1) = 107118720
C DIVER(2) = 0
C LOG0(1) = 107118720
C LOG0(2) = 1352628735
C ELSE IF (SMALL(1) .EQ. -2065213935)
* AND SMALL(2) .EQ. 1073072) THEN
* *** VAX WITH D FLOATING ***
C SMALL(1) = 128
C SMALL(2) = 0
C LARGE(1) = 32768
C LARGE(2) = 0
C RIGHT(1) = 9344
C RIGHT(2) = 0
C DIVER(1) = 9344
C DIVER(2) = 0
C LOG0(1) = 54979738
C LOG0(2) = 0
C ELSE IF (SMALL(1) .EQ. 1267827943)
C DATA LOG0(1), LOG0(2) / 706641072 /
* AND SMALL(2) .EQ. 706641072) THEN
* *** IBM MAINFRAME ***
C SMALL(1) = 1048576
SMALL(2) = 0
LARGE(1) = 2147483647
LARGE(2) = -1
RIGHT(1) = 45686592
RIGHT2 = 0
DIVER(1) = 0
DIVER2 = 0
LOG10(1) = 1091781651
LOG102 = 135268735
ELSE
WRITE(*, 9000)
STOP 779
END IF
SC = 987
END IF
C
C *** ISSUE STOP 778 IF ALL DATA STATEMENTS ARE OBVIOUSLY WRONG...
IF (DMACH(4).GE. 1.0D0) STOP 778
*CAS
*C IF(LT.1, OR.1.GT.5)
*C 1 CALL SETERR(24HDIMACH -1 OUT OF BOUNDS.24,1,2)
*C77
*C IF(LT.1, OR.1.GT.5)
*C 1 CALL SETERR("DIMACH(1) -1 OUT OF BOUNDS",24,1,2)
*C77
IF (L1T.1, OR.1. GT. 5) THEN
WRITE(*,*) 'DMACH(1): 1=.',1,' is out of bounds.'
STOP
END IF
DIMACH = DMACH(1)
RETURN
9000 FORMAT(46H Adjust Dimension by uncommenting data statements/
*3OH appropriate for your machine.)
C
C /* C source for DIMACH -- remove the * in column 1 */
C#include <stdin.h>
C#include <float.h>
C#include <math.h>
C
C*double dimach_floong *)
C*
C switch(*c)
C case 1: return DBL_MIN;
C case 2: return DBL_MAX;
C case 3: return DBL_EPSILON/FLT_RADIX;
C case 4: return DBL_EPSILON;
C case 5: return log10(FLT_RADIX);
C }
C printf(sidebar; "invalid argument: dimach(%d)", *c);
C exit(1);
C return 0; /* for compilers that complain of missing return values */
C}
END
C
C***************************************************************
C SUBROUTINE FUNC (NEQ, T, Y, YDOT)
C C This subroutine supplies the set of ODE's to be solved for the above
C mentioned reaction set. They are:
C C dA/dt = -A * (kpo*exp(-Ep(R*T(t)))+kbo*exp(-Eh(R*T(t))))
C dP/dt = A * kpo*exp(Ep(R*T(t)))
C dB/dt = A * kbo*exp(Eh(R*T(t)))
C T(t) is the specified temperature-time profile
C C YDOT     Y
C dA/dt     A
C dP/dt     P
C dB/dt     B
C
C COMMON PARAMTR, TEMP, IMAX, KPO, KBO, E, EP, EH, AO, RGAS
C INTEGER I, IMAX
C C Defining variables to allow for calculation of temperatures between
C the 0.01 sec intervals used to collect the temperature-time data.
C Linear interpolation between two adjacent points used.
C C ILOW temp at lower time   TEMPILOW
C from Basc - Turner profile
C C IACT actual temperature   TEMPIACT
C C IHIGH temp at higher time  TEMPIHIGH
C from Basc - Turner profile
C C DOUBLE PRECISION IACT, TEMPIACT
C INTEGER ILOW, IHIGH
C
C Universal gas constant = 1.987 cal/mole*K
Rgas = 1.987
C C Calculating which time interval to interpolate between
C for the true temperature at that time (since not all times
called in this subroutine will not be increments of 0.01 sec)
or 0.01 sec, the intervals we have stored temperature data.
Also checking to see if the desired calculation range is
within bounds of stored temperature-time profiles. Reseting
place in array to limit in cases where needed value is beyond
C
C ILOW = INT(T/100) + 1
IHIGH = ILOW + 1
IACT = (T/100) + 1.
IF (IHIGH GT IMAX) THEN
IACT = IACT - (IHIGH - IMAX)
IHIGH = IMAX
ILOW = IMAX - 1
END IF
IF (ILOW .LT. 0) THEN
   IACT = IACT - ILOW
   ILOW = 0
   IHIGH = 1
END IF
TEMPACT = (IACT-ILOW)*(HIGH-ILOW)**(TEMP(HIGH)-TEMP(ILOW))
          1 + TEMP(ILOW)

C Using the ordinary differential equations to solve for A(t), C
P(t), and B(t)
C
YDOT(1) = Y(1)*(-1)*(KPO*EXP(-EP/(RGAS*TEMPACT)) +
          * KRO*EXP(-EB/(RGAS*TEMPACT)))
YDOT(2) = Y(1)*KPO*EXP(-EP/(RGAS*TEMPACT))
YDOT(3) = Y(1)*KRO*EXP(-EB/(RGAS*TEMPACT))
RETURN
END
C******************************************************************************
CRM3.F
(partial listing, as discussed on the Appendix C title page)
PROGRAM CRM3

*------------------------------------------------------------

CRM3 = Competitive Reaction Model No. 3

This program uses the Levenberg-Marquardt Method to solve for the
kinetic parameters of a set of competitive reactions by fitting
experimental data for the amount of product P formed.
For each experimental data point to be fitted, LSODEF (the
Livermore Solver for ODE's) is used to predict the amount of
product P formed over that data point's corresponding
experimental temperature-time profile for the parameter values
guessed. For details on the reaction set used in this model,
refer to subroutine PLSEODE.

The user will have already input into text files using HSTORE.F the
following information:
  File name    Information held
  CRMDAT1.DAT  the no. of data points, total no. of adjustable
                parameters, and estimated std dev. for each
                of the data points
  CRMDAT2.DAT  names of files containing the temperature
                (Celsius)-time (sec) profiles for each
                data point/experiment
  CRMDAT3.DAT  the values we want to fit for, i.e. the
                dependent variables values, in the same
                order as in CRMDAT2.DAT, to match their
                values with the corresponding T-T profiles

This program is a combination of two programs: MRQMIN obtained from
MIT's FORTRAN Recipes Library (files/recipes/fortran/recipes)
and PLSEODE written by Hiroshi Saito. The multiple independent
reaction model evaluations were removed from MPRF and pasted
in its place was PLSEODE to numerically calculate the result of
the system of ODE's describing the reaction set (where a
time-dependent temperature profile is imposed).

*------------------------------------------------------------

C Main Program - Non-linear regression

Note from H. Salto: Since the codes for CRM2.F and
CRM3.F are the same except for FUNCTION
PLSEODE and SUBROUTINE FUNC, only these
changed subroutines/functions are listed here.
Please refer to CRM2.F for the rest of the code.
INTEGER MA
DOUBLE PRECISION A(MA)

C Declaring as external (as required by LSODE.F) subroutines FUNC
C (contains system of ODE's) and JDUMMY (dummy subroutine name
C for the jacobian I'm not using).
                AO, TINT
INTEGER I, J, IMAX
CHARACTER *10 FILEIN

C Declaring variables for LSODE subroutine

EXTERNAL FUNC
DOUBLE PRECISION ATOL, RTOL, RWORK, T, TOUT, Y
INTEGER ITO, ITASK, ISTATE, IOPT, LRW, LIW, MF
DIMENSION Y(3), ATOL(3), RWORK(63), IWORK(20)

C Declaring common parameters I input which will be passed from
C this main program to the subroutine
C
COMMON /PARAMTR/ TEMP, IMAX, KBO, KPO, EP, EB, AO

C Assigning the parameter values passed by array X, in the order:
C Ep  [e]  cal/gmole
C Eb  [e]  cal/gmole
C Ao  [e]  arbitrary
C kpo [e]  1/sec
C kbo [e]  1/sec

C PRINT *, 'CALL PLSODE'
C PRINT *, FILEIN
EP  = A(1)
EB  = A(2)
AO  = A(3)
KPO = A(4)
KBO = A(5)

C Initializing variables
C
I = 1
IMAX = 2000
J = 1
TINT = 0.0;

C Initializing the array to hold temp and time data
DO 100 J = 1, 2000, 1
    TIME(J) = 0.0
    TEMP(J) = 0.0
100 CONTINUE

C LOADING THE TEMP-TIME PROFILE from the specified data file
C and reading in Time (sec) & Temperature (Celcius) respectively
C
IMAX = 0

OPEN(UNIT=1,FILE=FILEIN,STATUS='OLD')
DO 1010 J=1,2000
    READ(J,*,END=2000,TO=C(TIME(J)),TEMP(J))
    TEMP(J) = TEMP(J) + 273.15
    IMAX = IMAX + 1
1010 CONTINUE
2000 CLOSE(UNIT=1)
C
C Preparing to call LSODE to solve system of ODE's for Pi(t)
C
C no. of ODE's = 3
NEQ = 3
C use initial conditions for initial values of A, P, and B.
Y(1) = AO
Y(2) = 0.0
Y(3) = 0.0
C integrate from time = 0 sec.
T = 0.10
C initially integrate to first interval, TINT
TOUT = TINT
C tolerances stored in an array
ITOL = 2
C relative tolerance parameter
RTOL = 1.0D-3
C absolute tolerance parameters for A, P, and B
ATOL(1) = 1.0D-6
ATOL(2) = 1.0D-6
ATOL(3) = 1.0D-6
C computes for trend of profile
ITASK = 1
C required by LSODE
ISTATE = 1
C no optional inputs used
IOPT = 0
C required for mf = 10
LRW = 68
C required for mf = 10
LIW = 20
C for nonstiff (adams) method, no jacobian used
MF = 10
C
C Using LSODE to solve for P(t)
C
DO 1000 J = 2, IMAX, 1
    CALL LSODE(FUNC,NEQ,Y,T,TOUT,ITOL,RTOL,ATOL,ITASK,
                ISTATE,IOPT,RWORK,LRW,IWORK,LIW,JDUMMY,MF)
99 FORMAT(1X,F5.2,4X,F4.3,4X,F4.3,4X,F4.3,4X,F4.3,4X,F4.3)
C WRITE(*,99) TOUT, Y(1), Y(2), Y(3)
99 CONTINUE
C C Stops program if error in LSODE calculation detected
C IF (ISTATE .LT. 0) THEN
    WRITE(*,990) ISTATE
990 FORMAT(2X,'ERROR HALT. ISTATE=',I3)
STOP
END IF
TOUT = TOUT + TNT
CONTINUE
FORMAT(12X,'NO. STEPS =',I4,11X,'NO. F-S =',I4)
WRITE(*,60) IWORK(11), IWORK(12), IWORK(13)

C Assigning the LSODE calculated value to PLSODE
PLSODE = Y(2)
RETURN
END

*******************************************************************************

C*******************************************************************************

SUBROUTINE FUNC (NEQ, T, Y, YDOT)
C
C This subroutine supplies the set of ODE's to be solved for the above
C mentioned reaction set. They are:
C
dA/dt = -A * kpo * exp(-Ep/R*T(t))

dP/dt = (A * kpo * exp(-Ep/R*T(t)))/((P * kbo * exp(Eb/R*T(t)))))

dB/dt = P * kbo * exp(Eb/R*T(t))

T(t) is the specified temperature-time profile

C YDOT     Y
C dA/dt    A
C dP/dt    P
C dB/dt    B

C*******************************************************************************

DOUBLE PRECISION T, Y(NEQ), YDOT(NEQ), TEMP(2000), R, KBO, KPO, EP, EB, AO, RGAS
INTEGER I, IMAX
COMMON /PARAMTR/ TEMP, IMAX, KBO, KPO, EP, EB, AO

C Defining variables to allow for calculation of temperatures between
C the 0.01 sec intervals used to collect the temperature-time data
C Linear interpolation between two adjacent points used
C
C ILOW   temp at lower time
C from Basc.-Turner profile
C
C IACT   actual temperature
C
C IHIGH  temp at higher time
C from Basc.-Turner profile
C
C*******************************************************************************

DOUBLE PRECISION IACT, TEMPACT
INTEGER ILOW, IHIGH

C Universal gas constant = 1.987 cal/mole*K
RGAS = 1.987

C Calculating which time interval to interpolate between
C for the true temperature at that time (since not all times
C called in the subroutine will not be increments of 10 m.sec)
C or 0.01 sec, the intervals we have stored temperature data
C Also checking to see if the desired calculation range is
C within bounds of stored temperature-time profiles. Resetting
C place in array to limit in cases where needed value is beyond.
C
C ILOW = INT(T*100.) + 1
C IHIGH = ILOW + 1
C IACT = (T*100.) + 1.
C IF (HIGHL GT IMAX) THEN
C IACT = IACT - (HIGHL - IMAX)
C IHIGH = IMAX
C ILOW = IMAX - 1
C END IF
C IF (ILOW .LT. 0) THEN
C IACT = IACT - ILOW
C ILOW = 0
C IHIGH = 1
C END IF
C TEMPACT = (1*(ILOW)) + (IHIGH-ILLOW)*(TEMP(IGH)-TEMP(ILLOW))
C
C*******************************************************************************

C*******************************************************************************

C Using the ordinary differential equations to solve for A(t),
C P(t), and R(t)
C
C YDOT(1) = Y(1)*(1.0-KPO*EXP(-EP/RGAS*TEMPACT))
C YDOT(2) = (Y(1)*KPO*EXP(-EP/RGAS*TEMPACT)) - (Y(2)*
C KBO*EXP(EB/RGAS*TEMPACT))
C YDOT(3) = (Y(2)*KBO*EXP(EB/RGAS*TEMPACT))
RETURN
END

*******************************************************************************
Appendix D

Heat Transfer Calculations and Measurements

Contents of this section

1. Intra-Soil Pan Temperature Gradients Calculation
2. Double-Thermocouple Experimental Measurements
3. Soil Intraparticle Thermal Gradients Calculation
Intra-Soil Pan Temperature Gradients Calculation

This calculation is to verify that there are no significant thermal gradients in the soil pan leading to discrepancies between the temperature the soil experiences and the temperature measured by a thermocouple placed beneath the soil pan. Assuming that free convection from the bottom of the pan is the main mode of heat loss leading to thermal gradients, a Biot number calculation can be used to examine the rate of heat loss from the bottom of the pan vs. conduction rates in the 0.001" thick pan. For a worst case calculation, it is supposed that the pan is at 1000°C in a 25°C reactor bath gas to maximize convective heat loss.

The stainless steel foil used to form the pan is a Type 302 Stainless Steel obtained from Shop-Aid Inc. in Woburn, MA. Typical composition for this type is 0.15% C, 2% Mn, 0.45% P, 0.03% S, 1.00% Si, 17-19% Cr, 8-10% Ni with the balance iron [1]. The thermal conductivity of the metal is [2]:

\[
\text{thermal conductivity} = \begin{cases} 
25^\circ\text{C} & 16.3 \text{ W/m°C} \\
1000^\circ\text{C} & 31 \text{ W/m°C}
\end{cases}
\]

To estimate the heat transfer coefficient at 1000°C, the equation (Table 7.2 in Holman [2]) for a heated plate facing downward was used:

\[
h = 0.59 \left( \frac{T_{\text{wall}} - T_\infty}{L} \right)^{1/4} = 0.59 \left( \frac{1000^\circ\text{C} - 25^\circ\text{C}}{0.001 \text{ in} \cdot \frac{0.3048 \text{ m}}{12 \text{ in.}}} \right)^{1/4} = 46.44 \text{ W/m}^2\text{C}
\]

where \(T_{\text{wall}}\) is the temperature of the pan, \(T_\infty\) is the bath gas temperature, and \(L\) is the thickness of the pan.

The Biot number is:

\[
Bi = \frac{hL}{k} = \frac{(46.44 \text{ W/m}^2\text{C})(0.001 \text{ in.})}{31 \text{ W/m°C}} \frac{0.3048 \text{ m}}{12 \text{ in.}} = 3.81 \times 10^{-5}
\]

Since the \(Bi \ll 0.1\), the pan can be assumed to be isothermal.

References


Double Thermocouple Measurement Notes

Experimental Description

In these runs, the temperatures of foils loaded with ~50 mg of neat EPA soil were measured with a new double thermocouple equipment setup. The thermocouples were placed on:

- bottom of the foil (convention with previous work)
- top of the foil (thermocouple placed on the foil surface but below the soil layer at the beginning of the run)

The thermocouples in both cases were the 0.0005" thick foil Type K thermocouples from Omega Engineering Inc. sheathed in asbestos paper except for the last 2 mm of the tip.

General Comments

These results indicate that:

- there is little difference (at most 35°C) between the temperatures measured from the bottom of the foil and the top of the foil
- temperatures measured from the top and below do not follow a systematic trend (i.e., temperatures measured from below are not always less than that from the top)
- from the lower temperature work, there is no great temperature difference between the foil surface and the soil above it

Other Comments

Thermocouple Movement - As the foil sag increases with increasing temperature, the amount of thermocouple movement varies similarly. Thus for top thermocouple measurements, the soil was not disturbed at 400°C while at 1000°C the top thermocouple moved as much as about 0.4" significantly disturbing the soil layer. This may have two implications:

- at least for low temperatures, we can claim small temperature gradients between the foil surface and the soil
- at the higher temperatures, the top thermocouple is initially below a soil layer but is drawn out due to thermocouple movement as the temperature increases. As the foil cools, the thermocouple moves back towards in initial position redepositing itself below a layer of soil. Hence, radiative losses or relative thermocouple placement may have an effect on temperatures measured depending on proximity of soil to each thermocouple (and each thermocouple to each other - Initial work with just foil and two thermocouples showed temperatures measured were more similar when they were located far from each other)

Reproducibility - As can be seen from Runs 2 and 3, the temperature differences measured were quite similar, indicating reasonable reproducibility of the method used.

Hiroshi Saito 4/8/94
## Double TC Measurements

<table>
<thead>
<tr>
<th>Run #</th>
<th>Bottom Thermocouple Temp. (°C)</th>
<th>Top Thermocouple Temp. (°C)</th>
<th>Difference (°C)</th>
<th>Abs. Value of Difference</th>
<th>Diff. (% of Btm. Temp.)</th>
<th>( R_{uv} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>370</td>
<td>377</td>
<td>-7</td>
<td>7</td>
<td>-1.088393065</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>512</td>
<td>497</td>
<td>15</td>
<td>15</td>
<td>1.910462969</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>544</td>
<td>509</td>
<td>35</td>
<td>35</td>
<td>4.283179343</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>769</td>
<td>792</td>
<td>-23</td>
<td>23</td>
<td>-2.206975963</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>853</td>
<td>841</td>
<td>12</td>
<td>12</td>
<td>1.06557741</td>
<td>5</td>
</tr>
</tbody>
</table>

Calculated with Abs. Temp. (K)

- Mean: 6.40
- Std. Deviation: 22.2
Double Thermocouple Measurements From Neat EPA Soil Runs

Comparison of Bottom and Top Thermocouple Temperatures Double Thermocouple Experiments Neat EPA Soil Runs
Temperature Difference Between Thermocouple Measurements
Double Thermocouple Experiments
Neat EPA Soil Runs

Percent Difference in Temperatures Measured
Double Thermocouple Temperature Measurements
Neat EPA Soil Runs
Intra-Soil Particle Temperature Gradients Calculation

This calculation is to verify that there are minimal thermal gradients in the soil particles during the temperature-time profile used for thermal treatment of soil. The thin layer of soil particles in the experimental soil pan can be viewed as a thin plate of the thickness of the soil particle diameter being heated from the bottom by the hot stage. The time necessary for the top of the plate to reach 95% of peak temperature is a good measure of the thermal gradients present.

For a 700°C hot stage temperature:

\[ 0.95 = \frac{T_{\text{top}} - 25^\circ C}{700^\circ C - 25^\circ C} \]

\[ T_{\text{top}} = 666^\circ C \]

where \( T_{\text{top}} \) is the temperature at the top of the soil particles. Using quartz (crystalline silicon dioxide) as an approximation for soil, the thermal conductivity and diffusivity of the soil particle, respectively, are [1]:

\[ k_{\text{soil}} \approx 0.04 \frac{W}{\text{cm} \cdot \text{K}} \]

\[ \alpha_{\text{soil}} \approx 0.012 \frac{\text{cm}^2}{\text{s}} \]

To calculate the Biot number necessary for this calculation, a heat transfer coefficient (h) is needed. Assuming that the hot soil bed is being cooled from below by natural convection:

\[ T_f = \frac{T_w + T_\infty}{2} = \frac{700^\circ C + 25^\circ C}{2} = 362.5^\circ C = 635.5 \text{ K} \]

\[ L = \frac{\text{pan area}}{\text{pan perimeter}} = \frac{\pi r^2}{2\pi} = \frac{r^2}{2} = \frac{0.79 \text{ in.} \cdot 0.3048 \text{ m}}{12 \text{ in.}} = 0.01 \text{ m} \]

where \( T_f \) is the gas film temperature underneath the hot stage, \( T_w \) is the hot stage temperature (lower soil surface temperature), \( T_\infty \) is ambient temperature, \( L \) is the characteristic length, and \( r \) is the soil pan radius. At the film temperature, the thermal conductivity, kinematic viscosity and thermal diffusivity of helium [2] respectively are:

\[ k_{\text{He}} = 0.236 \frac{W}{\text{m} \cdot \text{C}} \]

\[ \nu = 425.6 \times 10^{-6} \frac{\text{m}^2}{\text{s}} \]

\[ \alpha = 5.823 \times 10^{-4} \frac{\text{m}^2}{\text{s}} \]

Calculating the film Grashof (Gr_f) and Prandtl (Pr_f) numbers:
\[ \beta = \frac{1}{Tr(K)} \]

\[
Gr_f = \frac{\varepsilon \beta (T_w - T_\infty) L^3}{\nu^2} = \frac{9.8 \text{m}^2}{s} \frac{1}{635.65 \text{K}} \frac{(700^\circ \text{C} - 25^\circ \text{C}) (0.01 \text{ m})^3}{(425.6 \times 10^{-6} \text{ m}^2 \cdot \text{s})^2} = 57.45
\]

\[ Pr_f = \frac{\nu}{\alpha} = \frac{425.6 \times 10^{-6} \text{ m}^2}{5.823 \times 10^{-4} \text{ m}^2 \cdot \text{s}} = 0.73 \]

\[ Gr_f Pr_f = (25.88) (0.73) = 41.9 \]

Using the approximation for the lower heated surface of a horizontal plate from Table 7-1 in Holman [2] for a rough estimate,

\[ Nu_f = \frac{hL}{k_{\text{He}}} = 0.27 (Gr_f Pr_f)^{0.25} = 0.69 \]

yielding

\[ h = \frac{(0.69)k_{\text{He}}}{L} = \frac{(0.69)(0.236 \text{ W m}^{-1} \text{K}^{-1})}{0.01 \text{ m}} = 16.2 \text{ W m}^{-2} \text{K}^{-1} \]

Calculating the Biot number for the thin slab of quartz using the heat transfer coefficient \( h \) and the thermal conductivity of quartz:

\[ Bi = \frac{hL_{\text{slab}}}{k_{\text{soil}}} \]

for 63 \( \mu \text{m} \) soil particles:

\[ = \frac{(16.2 \text{ W m}^{-2} \text{K}^{-1}) (63 \times 10^{-6} \text{ m})}{0.04 \text{ W m}^{-1} \text{K}^{-1} \text{cm}^{-1} \cdot \text{cm}} = 2.5 \times 10^{-4} \]

for 125 \( \mu \text{m} \) soil particles:

\[ = \frac{(16.2 \text{ W m}^{-2} \text{K}^{-1}) (125 \times 10^{-6} \text{ m})}{0.04 \text{ W m}^{-1} \text{K}^{-1} \text{cm}^{-1} \cdot \text{cm}} = 5.1 \times 10^{-4} \]

These Biot numbers indicate from their small value that the midplane thermal internal resistance is very small compared to the surface film resistance.

Approximating \( 1/Bi \) to be 100 and using the Heisler charts in Holman [2], the characteristic Fourier number for \( T_{\text{top}} \) to reach 666\(^\circ \text{C} \) for both particle sizes is 2.5. Hence, for each particle size, the characteristic times are:

\[ Fo = 2.5 = \frac{\alpha \tau}{L_{\text{slab}}^2} \]

269
\[ \tau = \frac{2.5 \ L_{\text{slab}}^2}{\alpha_{\text{soil}}} \]

for 63 \( \mu \text{m} \) soil particles: \[ \tau = \frac{(2.5) \ (63 \times 10^{-6} \ \text{m})^2}{0.012 \ \text{cm}^2 / \text{s} \ \text{m}^2 / 100^2 \text{cm}^2} = 0.0083 \ \text{s} = 8.3 \ \text{ms} \]

for 125 \( \mu \text{m} \) soil particles: \[ \tau = \frac{(2.5) \ (125 \times 10^{-6} \ \text{m})^2}{0.012 \ \text{cm}^2 / \text{s} \ \text{m}^2 / 100^2 \text{cm}^2} = 0.030 \ \text{s} = 32 \ \text{ms} \]

According to these estimates, temperature disturbances should be relaxed in times that are short compared to the total heatup time (i.e. \(~32 \ \text{ms} \) vs. \( 100 \ \text{ms} \) in our worst case of heating our largest particles from room temperature to 500\(^\circ\)C at 5000\(^\circ\)C/s). Alternatively, the temperature lag between the particle top and bottom surface could be as high as 5000\(^\circ\)C/s \times 0.032 \ \text{s} \approx 160\(^\circ\)C, but this pertains to only a fraction of the soil particles used in one run. Since most of the experimental runs were at much slower heating rates (100 or 1000\(^\circ\)C/s) and 125 \( \mu \text{m} \) is an upper limit on our particle size, it is estimated that internal temperature gradients are \(< 32\(^\circ\)C \) in most cases, and that intrasample spatial temperature gradients are not a major factor.

In addition, these calculated \( \text{"lag"} \) times are expected to be higher than the actual values, since approximating \( 1/Bi = 100 \) gave a large Fourier number, and since a layer of spherical particles contains less material than a slab of equivalent thickness. The space between the spherical particles would allow also for conductive heat transfer through the gas, improving heat transfer to the soil particle and reducing the \( \text{"lag"} \) times to values less than those calculated above.

References


Appendix E

Intraparticle CO/CO₂ Equilibrium Calculation

The purpose of this calculation is to determine if mass transfer limitations could be leading to the observed trends in the experimental data. If mass transfer is controlling and carbon monoxide is produced via the reaction

\[ \text{CO}_2(\text{g}) + \text{C}(\text{s}) \rightarrow 2 \text{CO}(\text{g}) \]

an equilibrium composition of CO₂ and CO would be expected in the volatilized gases. Since equilibrium for this reaction is determined by both temperature and pressure, the possibility of mass transfer limitations can be tested by determining the equilibrium pressure within the particle required to produce the molar CO/CO₂ ratios obtained experimentally for a few select temperatures. Here it is assumed that carbonate degradation to produce carbon dioxi'de is not rate limiting, which is not unreasonable since carbon dioxide is the product produced in the largest quantities in the soil experiments.

From the Gibbs Phase rule: \[ L = n + 2 - \pi - r \]
\[ = 3 + 2 - 2 - 1 \]
\[ = 2 \] (divariant system)

where
\[ n = 3 \text{ components (CO}_2, \text{ C, CO)} \]
\[ \pi = 2 \text{ (solid, gas)} \]
\[ r = 1 \text{ (1 reaction, species molar ratios unknown)} \]

From thermodynamics, the equilibrium constant can be written as follows:

\[
K = \exp\left(\frac{-\Delta G^o(T)}{RT}\right) = \prod_{j=1}^{n} \left(\frac{f_j}{f_j^0}\right)^{\nu_j}
\]

where \(\Delta G^o(T)\) is the Gibbs free energy of the reaction at the reference temperature, \(R\) is the Universal Gas Constant, \(T\) is the reaction temperature in Kelvin, \(f_j\) is the fugacity of component \(j\) in the reaction mixture, \(f_j^0\) is the fugacity of component \(j\) at the reference temperature as a pure material, and \(\nu_j\) is the stoichiometric coefficient of component \(j\) in the reaction. Applying this to the above reaction:

\[
K = \left(\frac{f_{\text{CO}_2}}{f_{\text{CO}_2}^0}\right)^{\nu_{\text{CO}_2}}\left(\frac{f_{\text{C}}}{f_{\text{C}}^0}\right)^{\nu_{\text{C}}}\left(\frac{f_{\text{CO}}}{f_{\text{CO}}^0}\right)^{2} = \exp\left(\frac{-\Delta G^o(T)}{RT}\right)
\]

Taking the reference state to be the reaction temperature for each component in its pure state at 1 atmosphere, and assuming an ideal gas mixture
\[ \hat{f}_{\text{CO}_2} = \Phi_{\text{CO}_2} y_{\text{CO}_2} P = y_{\text{CO}_2} P \quad f_{\text{CO}_2}^0 = (1)(1 \text{ atm.}) = 1 \text{ atm.} \]
\[ \hat{f}_{\text{CO}} = \Phi_{\text{CO}} y_{\text{CO}} P = y_{\text{CO}} P \quad f_{\text{CO}}^0 = (1)(1 \text{ atm.}) = 1 \text{ atm.} \]

where \( \Phi_j \) is the fugacity coefficient for component \( j \) (\( \Phi_j = 1 \) in ideal gas mixtures), \( y_j \) is the mole fraction of component \( j \) in the gas phase, and \( P \) is the reaction pressure. Assuming that the solid is pure carbon and pressure effects are negligible

\[ \hat{f}_C = f_C^0 \]

Substituting these five expressions into equation (I),

\[ K = \exp \left( \frac{-\Delta G^o(T)}{RT} \right) = \left( \frac{y_{\text{CO}_2} P}{1 \text{ atm.}} \right)^{-1} \left( \frac{y_{\text{CO}} P}{1 \text{ atm.}} \right)^2 = \frac{y_{\text{CO}_2}^2 P}{(1 - y_{\text{CO}})} \quad (\text{II}) \]

Rearranging equation (II),

\[ P = \frac{K(1-y_{\text{CO}})}{y_{\text{CO}_2}^2} \quad (\text{III}) \]

Thermodynamic data for carbon dioxide, carbon monoxide, and graphite allows for \( \Delta G^o(T) \) (and hence \( K \)) to be calculated at different temperatures.

Using experimental yields of carbon dioxide and carbon monoxide to calculate \( y_{\text{CO}} \) (assuming the C1-C2 hydrocarbons contribute negligibly), the equilibrium pressures were calculated for both neat and pyrene-contaminated soil runs at 800, 1000, and 1200 K (see next page). As can be seen from the calculation results, pressures of order \( 10^9 - 10^{10} \) atm. are necessary for equilibrium for the measured CO/CO2 ratios, indicating that mass transfer limitations are probably not important in this study.
<table>
<thead>
<tr>
<th>Temperature(K)</th>
<th>H-H298</th>
<th>H-H298</th>
<th>S</th>
<th>H-H298</th>
<th>S</th>
<th>ΔH(T)</th>
<th>ΔS(T)</th>
<th>ΔG(T)</th>
<th>K</th>
<th>CO (μg, exp.)</th>
<th>CO2 (μg, exp.)</th>
<th>yCO (mole frac)</th>
<th>P (atm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>800</td>
<td>5.453</td>
<td>61.520</td>
<td>1.825</td>
<td>4.739</td>
<td>3.627</td>
<td>54.296</td>
<td>-0.024</td>
<td>42.333</td>
<td>-33.8904</td>
<td>1.816E+09</td>
<td>27.82</td>
<td>177.47</td>
<td>0.198</td>
</tr>
<tr>
<td>1000</td>
<td>7.984</td>
<td>64.342</td>
<td>2.819</td>
<td>5.845</td>
<td>5.183</td>
<td>56.031</td>
<td>-0.437</td>
<td>41.875</td>
<td>-42.312</td>
<td>1.770E+09</td>
<td>89.35</td>
<td>543.32</td>
<td>0.205</td>
</tr>
<tr>
<td>1200</td>
<td>10.632</td>
<td>66.754</td>
<td>3.881</td>
<td>6.813</td>
<td>6.794</td>
<td>57.499</td>
<td>-0.925</td>
<td>41.431</td>
<td>-50.6422</td>
<td>1.675E+09</td>
<td>286.99</td>
<td>1663.34</td>
<td>0.213</td>
</tr>
</tbody>
</table>

H = kcal/gmole
S = cal/gmoleK
G = kJ/kmole

<table>
<thead>
<tr>
<th>Temperature(K)</th>
<th>H-H298</th>
<th>H-H298</th>
<th>S</th>
<th>H-H298</th>
<th>S</th>
<th>ΔH(T)</th>
<th>ΔS(T)</th>
<th>ΔG(T)</th>
<th>K</th>
<th>CO (μg, exp.)</th>
<th>CO2 (μg, exp.)</th>
<th>yCO (mole frac)</th>
<th>P (atm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>800</td>
<td>5.453</td>
<td>61.520</td>
<td>1.825</td>
<td>4.739</td>
<td>3.627</td>
<td>54.296</td>
<td>-0.024</td>
<td>42.333</td>
<td>-33.8904</td>
<td>1.816E+09</td>
<td>27.82</td>
<td>177.47</td>
<td>0.503</td>
</tr>
<tr>
<td>1000</td>
<td>7.984</td>
<td>64.342</td>
<td>2.819</td>
<td>5.845</td>
<td>5.183</td>
<td>56.031</td>
<td>-0.437</td>
<td>41.875</td>
<td>-42.312</td>
<td>1.770E+09</td>
<td>89.35</td>
<td>543.32</td>
<td>0.347</td>
</tr>
<tr>
<td>1200</td>
<td>10.632</td>
<td>66.754</td>
<td>3.881</td>
<td>6.813</td>
<td>6.794</td>
<td>57.499</td>
<td>-0.925</td>
<td>41.431</td>
<td>-50.6422</td>
<td>1.675E+09</td>
<td>286.99</td>
<td>1663.34</td>
<td>0.503</td>
</tr>
</tbody>
</table>