MODELING OF RADIATION EFFECTS ON NUCLEAR WASTE PACKAGE MATERIALS

by

SCOTT ARTHUR SIMONSON

B. S., Nuclear Engineering, University of Illinois (1981)
M. S., Nuclear Engineering, University of Illinois (1983)

SUBMITTED TO THE DEPARTMENT OF
NUCLEAR ENGINEERING
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

at the

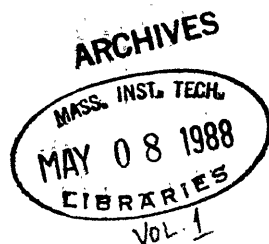MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 1988

c   Scott Arthur Simonson, 1988

Signature of Author..
Department of Nuclear Engineering
September 2, 1988

Certified by..
Ronald G. Ballinger
Associate Professor of Nuclear Engineering
Associate Professor of Material Science and Engineering
Thesis Supervisor

Accepted by..
Allan F. Henry
Chairman, Department Committee
on Graduate Students

1

Vol. 1

MODELING OF RADIATION EFFECTS ON NUCLEAR WASTE PACKAGE MATERIALS

by

SCOTT ARTHUR SIMONSON

Submitted to the Department of Nuclear Engineering
on September 6, 1988 in partial fulfillment of the
requirements for the Degree of Doctor of Philosophy in
Nuclear Engineering

ABSTRACT

A methodology is developed for the assessment of radiation effects on nuclear waste package materials. An assessment of the current status of understanding with regard to waste package materials and their behavior in radiation environments is presented. The methodology is used to make prediction as the the chemically induced changes in the groundwater surrounding nuclear waste packages in a repository in tuff. The predictions indicate that mechansims not currently being pursued by the Department of Energy may be a factor in the long-term performance of nuclear waste

The methodology embodies a physical model of the effects of radiation on aqueous solutions. Coupled to the physical model is a method for analyzing the complex nature of the physical model using adjoint sensitivity analysis. The sensitivity aids in both the physical understanding of the processes involved as well as aiding in eliminating portions of the model that have no bearing on the desired results. A computer implementation of the methodology is provided.

Thesis Supervisor : Ronald G. Ballinger
Title : Associate Professor of Nuclear Engineering

ACKNOWLEDGEMENTS

First and foremost I must thank my wife, in too many way to elaborate she has made this thesis possible. This thesis is dedicated to Fonda.

My parents have always been a source of comfort and strength to me, they deserve my thanks as well. The support of my family was also greatly appreciated. Specifically, I must thank Alison for the typing and editing help.

Ron Ballinger has been both mentor and friend, I cannot thank him enough for his assistance. Terry Sullivan, Maureen Psaila-Dombrowski, Ron Christensen, Russ Jones have been very helpful in this endeavor, my thanks to you.

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 Nuclear Waste Isolation

The Congress of the United States in 1982 determined that radioactive waste created a potential risk and required safe and environmentally sound disposal methods. It was also found that up to that point, the Federal Government had not done an adequate job in finding a permanent solution. Therefore, the Congress empowered the Secretary of Energy to characterize a number of suitable sites for the potential use as a high-level radioactive waste repository. Due to a perceived stagnation in the characterization process, Congress amended the Nuclear Waste Policy Act (NWPA) of 1982 on December 22, 1987, in the Budget Reconciliation Act for Fiscal Year 1988 [DOE, 1987]. In this amendment, Congress directed the Department of Energy (DOE) to characterize a site located near Yucca Mountain, Nevada and cease consideration of other sites. Pending the outcome of a search for a willing state or Indian tribe to take the repository, the Yucca Mountain site will be the nation's first nuclear waste repository unless the site proves unacceptable for technical reasons.

The location of the Yucca mountain site is depicted in Figure 1.1. The repository will be at least 200 meters below the ground surface yet still 200 to 300 meters above the water table. Being located above the water table is advantageous since the most plausible scenarios for the accidental release of radionuclides to the

environment involve the transport of radionuclides in ground water. The site is very arid, having less than six inches of rain per year, another advantage with respect to ground-water intrusion into the repository. The repository is projected to hold 70,000 metric tons of spent nuclear fuel. Based upon current projections, this will accommodate all the fuel produced through the year 2010. Over the next ten years, the DOE will be characterizing the Yucca Mountain site, collecting the data necessary to demonstrate the safety of this site for a nuclear waste repository.

The technical criteria that the site must meet are established by the Nuclear Regulatory Commission (NRC) and the Environmental Protection Agency (EPA), as well as DOE's own regulation (10 CFR 960), as specified in the NWPA [NWPA, 1987, Sec..121]. The NRC first published the required criteria in the Federal Register in 1983, designated 10 CFR 60[1]. The EPA published its required criteria in 1984, designated 40 CFR 191[2].

The specific criteria that bear upon this thesis are those that involve the containment, and release and transport of radionuclides to the accessible environment. The NRC has jurisdiction over the engineered barriers of the repository; therefore, NRC's criteria deal with the barriers and releases at these barriers. The NRC has proposed that the waste packages provide "substantially complete containment" for a period of 1000 years. In addition, the NRC requires that the amount released per year from the engineered

---

[1] 10 CFR Part 60 was revised and republished in 1987.

[2] This set of criteria was remanded in 1987, but for the purposes of this thesis, the intent of the original, remanded rules suffices as general guidelines.

barrier system not exceed one part in one hundred thousand of the curie inventory of the particular radionuclide present at 1000 years.

The EPA criteria govern the releases of radionuclides to the accessible environment. The EPA criteria are based on the already established guidelines for radionuclide releases, based upon maximum permissible releases to water and air. The accessible environment begins at some distance from the repository and therefore, the regulations do not bear directly on the engineered barriers.

The DOE has chosen to introduce its own "working" criteria that are intended to satisfy the NRC's criteria. These criteria indirectly address the compliance issue and provide the DOE's interpretation of the NRC and EPA requirements. The main criterion established by the DOE addresses the issue of "substantially complete containment" [DOE, 1987]:

The Department of Energy understands the requirement for substantially complete containment of high-level waste (HLW) within the set of waste packages to mean that a very large fraction of the radioactivity that results from the HLW originally emplaced in the underground facility will be contained within the set of waste packages during the containment period. Therefore, the requirement would be met if a significant number of the waste packages were to provide total containment of the radioactivity within those waste packages or if the radioactivity released from the set of waste

packages during the containment were sufficiently small. The precise fraction of HLW that should be retained within the set of waste packages, number of waste packages that should provide total containment, or constraints that should be placed on the rate of release from the set of waste packages to meet the requirements for substantially complete containment should not be determined until the site is sufficiently well characterized[1]. Such a precise interpretation depends in large part on the level of waste-package performance needed at the site. Therefore, a specific interpretation of the general requirement cannot be made until additional information regarding site conditions and the characteristics of alternative materials and waste package designs subject to these conditions is available.

The proposal to satisfy these criteria involves the use of a highly corrosion-resistant metallic waste package. Conceptual design of this package is depicted in Figure 1.2. The proposed containers, shown in Configuration 1, hold four boiling water reactor (BWR) and three pressurized water reactor (PWR) fuel elements. Based on projected inventories of spent nuclear fuel [DOE, 1987], a small excess of BWR fuel will result (less than 7% of the total number of waste packages) and these will be

---

[1]The design goals of the DOE are [DOE,1987, Sec. 8.2]: 80% of packages intact at 1000 years; 99 percent of all waste initially emplaced will be retained; any releases in any one year shall not exceed one part in 100,000 of the total inventory of radionuclide activity present within the geologic repository system in that year.

accommodated in Configuration 2, also shown in Figure 1.2. The materials to be used for the containers will be extensively tested to provide the data necessary to assure that the criteria for containment and radionuclide release are satisfied. A more detailed description of the waste package proposed by the DOE is given in Chapter 2.

Ultimately, the DOE must use mathematical models of experimentally-observed behaviors over the range of possible physical and chemical environments to describe the behavior of the waste packages and thereby demonstrate compliance with the criteria. Since it is practically impossible to perform testing over the time periods of interest, models used to make predictions must be extrapolated beyond the existing experimental data. This is a valid approach given that the models explain the experimental data in terms of the fundamental laws of chemistry and physics, and that no additional, unknown at this time, phenomena interfere.

Figure 1.1   Yucca Mountain, Nevada, Showing Proposed Site

for the First Nuclear Waste Repository

Figure 1.2   Configuration of Unconsolidated Nuclear Fuel Container

28 IN. (71 cm)
DIAMETER

FUEL DIVIDERS
0.125 IN. (0.32 cm)
THICK

3 PWR FUEL ASSEMBLIES
8.5 x 8.5 IN. (21.6 x 21.6 cm)

CONFIGURATION 1.
THREE INTACT PWR ASSEMBLIES
FOUR INTACT BWR ASSEMBLIES

4 BWR FUEL ASSEMBLIES
5.5 x 5.5 IN (14 x 14 cm)

0.375 IN.
(0.95 cm)

187.5 IN. (476 cm)

28 IN. (71 cm)

28 IN. (71 cm)
DIAMETER

FUEL DIVIDERS
0.125 IN. (0.32 cm)
THICK

CONFIGURATION 2.
TEN INTACT BWR ASSEMBLIES

10 BWR FUEL ASSEMBLIES
5.5 x 5.5 IN. (14 x 14 cm)

0.375 IN.
(0.95 cm)

1.2   Thesis Objectives

It is well recognized that the environment surrounding nuclear waste packages will contain a significant radiation field [DOE, 1987]. Therefore, it is of interest to know the effects of radiation on the environment surrounding the waste package and to be able to predict how these effects may influence containment and release of radionuclides.

The most notable effects of radiation with regard to nuclear waste packages, aside from the direct effects on workers handling the waste, are the changes that are induced in the chemistry of the surrounding environment. Specifically, it is important to know if any of the changes will adversely affect the corrosion behavior of the metal barriers, or the release characteristics of radionuclides in the event of a canister failure.

Having recognized the potential of radiation to alter the environment surrounding the waste packages and the limited understanding of radiation effects that now exist, it is improtant to develop better modeling capabilities of the phenomena than those to date. This need for modeling capabilities was also called for by Von Konynenburg [1986]:

> " A precise theoretical analysis of this system [radiation effects in the repository environment] would require a time-dependent computer model incorporating at least two compartments to represent the two fluid phases. Within each

compartment, provisions would need to be made for inputting the yields of the primary radiolytic species and calculating their reactions by means of coupled rate equations. The significant reactons and their rates would have to be known for both phases at the temperature of interest. Provisions would have to be made for transport of species between the two phases, and the equations governing such transport would have to be supplied. Significant interactions between the fluid and solid phases would also have to be understood well enough to be modeled mathematically"

This and the other statement of concern supplied the incentive to develop the model for the tuff repository to be sited in Nevada[1]. The ultimate goal of modeling is to predict radiation effects in repository environments.   However, another important aspect of modeling is its usefulness to experimentalists in choosing the best experiments to conduct in the development of the data base necessary to support the characterization of the facility.

Due to the above considerations, a program to model the radiation effects on the materials to be used in the repository environment was undertaken. The goal was to include all the known effects of radiation and then make an assessment of the most important interactions that need to be addressed by further

---

[1]The Nevada repostitory is often referred to as the "tuff" repository in reference to the type of rock that occurs at the expected repository depth. Tuff rock is the result of fine volcanic ash being deposited in deep layers. The depth of the layer insulates the ash and it becomes hot enough to melt into a grainy rock structure.

experimentation. The model uses only experimentally determined parameters, no fitting of data is performed. The means to improve the model are through experimentation using the model as a guide to performing the critical experiments. Additionally, the model is formulated so as to allow for incorporation of effects related to localized corrosion phenomena, being developed in concurrent work [Psaila, 1989].

Phenomena addressed by the models used to assess radiation effects are quite complex; it is therefore useful to have an automatic means of evaluating the important parameters[1] of the model. This is described by a sensitivity-analysis model. Sensitivity analysis tells how large a change we would get in the final results given a small change to any, or all, of the parameters. Put another way, the sensitivity analysis provides the sensitivity of any or all dependendent model variables to perturbations in any or all of the independent variables. Key parameters of the model are thus identified and the unimportant ones can quickly be dismissed.

An integral part of all modeling studies is the verification of the model. Verification and validation involve checking the model to assure that it is; (1) mathematically correct and (2) represents the physical systems being considered. The mathematical verification of this model is performed by analytically solving a simple model for all the quantities that are to be calculated numerically. A consistency check has been made to assure that the underlying theory for the sensitivity analysis is correct as well.

---

[1]Parameters refer to the basic quantities used to define the models, e.g. chemical reaction rate constants.

Validation involves checking the model against physical reality; this is considered as part of the applications.

Although the emphasis of this thesis is toward the determination of the effects of radiation on nuclear waste materials, the formulation developed in this thesis has a wide range of applicability. Many physical systems have mathematical characteristics identical to those presented here (the law of physics and chemistry used in this thesis do not change, just the systems to which they are applied). Additionally, the effects of radiation are of interest to the nuclear industry as a whole, and the models presented can be a contribution to this area as well.

A major effort is underway at the Massachusetts Institute of Technology to understand the nature of radiation effects on aqueous solutions. The key environments being studied are those that would be encountered in nuclear reactor systems. Simulated reactors (Pressurized Water Reactors (PWR) and Boiling Water Reactors (BWR)) are being developed as an experimental tool in these and other investigations. A high pressure water loop through the reactor is also being assembled to perform tests to further the understanding of irradiation-assisted stress corrosion cracking. One of the tools to be used in the design and interpretation of the tests to be conducted is a mathematical model of the effects of radiation on aqueous solutions. The necessary modifications to the models presented herein are outlined so that this model can be adapted to assist in the development of this technology.

1.3 Thesis Organization

Chapter 2 provides a discussion of the relevant aspects of the proposed waste package and its expected environment. Chapter 3 discusses the basic processes of radiation interactions with solution. The theoretical model is presented in Chapter 4. The numerical formulation of the model is given in Chapter 5. A verification to the numerics and theory are given in Chapter 6. Chapter 7 provides applications of the model to experimental data and makes predictions for nuclear waste package performance. Chapter 8 contains the concluding remarks and offers some reccomendations for future work. The references are contained in Chapter 9.

2.0 WASTE PACKAGE SYSTEM

This chapter details the proposed designs of the DOE for the waste package system. An overview of the relevant phenomena with regard to radiation effects is presented. The details of the physical nature of the interactions and the mathematical representations are given in subsequent chapters. Additionally, data and calculations relevant to the repository and radiation effects are presented to supplement the information provided by the DOE.

The first section describes the geometry, materials and important interactions of the container and waste forms. The next section discusses the expected thermal environment. Calculations of the radiation fields expected in and around the waste package are given in Section 2.3. Finally, a review of the work performed by the DOE on the waste containers and the waste form is presented.

2.1 Waste Package Components

A schematic presentation of the waste package is given in Figure 2.1. The actual dimensions and internal layout of the package are given in Figure 1.2. The waste is enclosed within a metal container that has been welded shut. Each container will

have 2.13 metric tons of spent fuel[1] that is at least 10 years old. The container is placed into a hole that has been bored into the tuff rock[2]. The holes are spaced 10 to 20 meters apart along tunnels that have been mined into the rock[3]. The age of the waste, the pitch of the holes, the number of cans per hole and the number of metric tons of waste per container determine how much thermal energy is being produced[4], and hence, determine the temperature history of the repository. The expected temperatures are discussed in Section 2.2.

During the period of containment, the containers are designed go remain intact. Under these conditions, only gamma radiation will escape the container to interact with ground water or the surrounding rock. In the event of a breach of the canister, beta and alpha radiations would also be present to interact with ground water or the rock. The interaction of the radiations with water is termed radiolysis. Radiolysis sets off a chain of events wherein the radiation produces very reactive chemical species that go on to interact with the other chemical entities of the solution and the solids present. Figure 2.2 schematically depicts the relevant physical phenomena that must be evaluated when assessing radiolysis interactions. Four main interactions are addressed in

---

[1]Spent fuel is comprised of uranium dioxide pellets enclosed in long tubes of a zirconium alloy called Zircaloy. The tubes are assembled into square lattices called fuel elements. The fuel elements will be placed into the containers after they have been irradiated in a reactor for some period of time.

[2]The container is depicted vertically but it may be horizontal as well.

[3]The spacing of the holes is called the pitch; the tunnels are often referred to as drifts.

[4]Usually expressed as a "power density" in kilowatts per acre.

Figure 2.2, interaction of the gaseous species with the liquid, interaction of solid species with the liquid, interaction of radiation with the liquid and finally interaction of all the contributions in the liquid phase. Section 2.2 discusses the thermal environment surrounding the waste package since temperature affects many of the processes depicted in Figure 2.2. Section 2.3 discusses the expected radiation levels, and the remainder of the thesis considers the radiolysis interactions and the implications for nuclear waste management.

The most significant means of release of radionuclides to an environment outside the repository involves transport through ground water. Also, the presence of liquid water may play an important role in the degradation of both the container and the spent fuel. The repository is proposed to be well above (200 to 400 m) the local water table at Yucca Mountain [DOE, 1987]. In addition, the expected thermal environment should keep temperatures above the boiling point of water for 1000 years or more (see Section 2.3). However there may be periods of water inflow and evaporation, especially near the periphery of the repository. The cycle of inflow and evaporation may lead to concentration of the electrolytic species (e.g., Cl^-, SO_4^{-2}, F^-) [Juhas, 1984] by as much as a factor of 10 to 100 times [Glass, 1986]. Therefore it is important to consider this concentration effect in the analyses.

The The most important components of the waste package system with regard to this thesis are the nuclear waste container

and the spent nuclear fuel. Details of these two aspects of the waste package are discussed in Sections.2.4 and 2.5.

Figure 2.1   Schematic of a Waste Container in a Borehole

Figure 2.2   D~ 'ction of Relevant Physical Processes With
Rega.d to Radiation Interactions

O_2, N_2, CO_2, H_2, NH_3

Gas-Liquid   Equilibria

Metal
Container

O_2, N_2, CO_2, H_2, NH_3

NH3, NH4+, HCO3-, CO32-, CO3-2, H+, OH-

Liquid Phase Equilibria

Radiation
Chemistry
Model

γ   Radiation
Unbreached
Container

e-, H_2O_2, H, OH, HO_2, H_2

α, β   Radiation
Breached
Container

Radiolytic Production

Ni(OH)+, Ni++, Fe++, Fe(OH)+, Fe+++,
$Fe(OH)_2$ +, FeOH++

Solid Phase Equilibria

Metal Oxide, $Fe(OH)_2$, $Ni(OH)_2$

2.2 Thermal Environment

The thermodynamics and kinetics of the chemical and electrochemical reactions associated with the interaction of the waste container and its environment are strongly temperature dependent. Radioactive decay of the fission products in the spent fuel results in the deposition of heat energy in the fuel which will, in turn, result in heat being deposited in the canister wall.

The calculated thermal history for the DOE reference conceptual design [DOE, 1987] is given in Figure 2.3. As seen in the figure, the outer surface of the container is expected to remain above the boiling point of water at the repository depth (96°C) for well beyond the 1000 year containment period. Deviations from this reference case are discussed below.

The thermal history is approximate and the reference design may be different from the one actually used. The actual thermal loadings may be altered due to other considerations such as the temperature rise at the top of Yucca Mountain. If the oldest fuel is emplaced first, there is the possibility that fuel of the reference age could not be emplaced until many years after it was designed to be emplaced [MIT, 1988]. In addition, the correlations used to determine the heat-transfer characteristics of the fuel [Pescatore, 1988], and borehole walls [St. John, 1985], and the general heat transfer of the moist air environment [Preuss, 1984] may not be accurately represented in the above calculations. They point to a possible lowering of the temperatures; therefore the temperature may drop below the boiling point of water thus allowing liquid to

water contact a significant number of waste packages at times earlier than predicted by DOE [1987].

Figure 2.3   Thermal Profile Near Spent Nuclear Fuel

Containers over a 1000 year time period

TIME AFTER EMPLACEMENT (YEARS)

INITIAL CONDITIONS

WASTE FORM . . . . SPENT FUEL
LOCAL POWER DENSITY . . . . 57.0 kW/acre
AREAL POWER DENSITY 48.4
AVERAGE 10-YR POWER . . . . 3.3 kW
CONTAINER DIAMETER . . . . 0.7 m
DISTANCE BETWEEN CONTAINERS . . . . 5 m
DISTANCE BETWEEN DRIFTS . . . . 47 m

2.3   Radiation   Environment

The overall validity of this work depends upon accurate knowledge of the expected radiation environments surrounding waste packages. The Site Characterization Plan [DOE 1987] puts the estimate of the gamma radiation field at "less than 1 x 10^5 rads/h". The original assessment of dose rate [Van Konynenburg, 1984] included only four radionuclides [^{106}Ru, ^{134}Cs, ^{137}Cs, and ^{144}Ce] and was for a single fuel element that had been out of the reactor for 2.45 years. The reference design calls for at least 10 years out of reactor and a different fuel loading (3 PWR and 4 BWR elements) and configuration in the waste package. As mentioned in Section 2.2, the actual age may even be older than 10 years, resulting in further reduction of the radiation field. There is also no mention of the expected radiation field that would be present in the ground water due to alpha emitters on the fuel surface and in the water. The following assessment of the gamma and alpha radiation is intended to provide a more realistic assessment of dose rates than the DOE study [Van Konynenburg, 1984].

2.3.1   Gamma   Radiation   Fields

This section details calculations made for various container thicknesses and for environmental conditions that would be expected in and around the waste packages. The data for the calculations were formulated assuming the reference geometry given in Figure 1.2, configuration 1. The emplaced fuel is assumed

to be 10 years old. Data for the radionuclide inventories have been generated using ORIGEN II [Croff, 1980] and compacted into appropriate gamma energy groups [Jansen, 1987].

The material within the container was smeared out[1] throughout the interior volume. The effective densities of the various materials are given in Table 2.1. The total loading of the container was calculated to be 2.13 metric tons of spent nuclear fuel. The container was given thicknesses of 1.5 and 2.5 cm of steel (iron was used for the calculation to approximate steel.) The selection of a container thickness had not been made at the time of this writing [DOE, 1987] and therefore two likely thicknesses were used. The dose rates are calculated at the midplane of the active fuel length (192.5 cm from bottom) and 1 cm from the outer surface of the package.

The computer code ISOSHLD [Engel, 1966; Kottwitz, 1984] was used to perform the gamma shielding analysis. ISOSHLD is a point kernel integration package that is set up to solve a wide variety of shielding problems. The code allows for variable energy groups and geometry and has a wide selection of available materials. The geometry chosen for this analysis was a cylinder with cylindrical shields. Uranium, oxygen and zirconium occupy a cylindrical fuel region; iron is used as a cylindrical shield exterior to the fuel region to simulate the container; and water is assumed to surround the package as the final shield. Results are calculated for iron container thicknesses of 1.5 and 2.5 cm, respectively.

---

[1]smearing out is simply averaging the amount of material as if it were homogencously distributed throughout the available volume.

Values of 4000 R/hr and 2100 R/hr were determined using the above data in ISOSHLD. These results should be closer to the initial expected dose rates than those predicted previously [Van Konynenburg, 1984] since they consider waste of the proper age, relevant container thicknesses, and an accurate representation of the proposed fuel loadings of the container. These lower values also indicate that the testing conditions being used to evaluate the various materials are too high[1]. The applications presented in Chapter 7 use the values calculated in this section to make predictions.

2.3.2 Radiation Dose to Contaminated Ground Water

It is known that the gamma field associated with spent fuel will decay more rapidly than the fields associated with alpha and beta radiations [Jansen, 1987; Lundgren, 1982]. In the long term (300 to 1000 years), a major source of oxidants in the event of a container failure will be the radiolysis of the water by the alpha and beta emitters.

Two effects are important with regard to the production of oxidants. First, raising the oxidation potential of a solution will in general increase the solubility of the actinide species [see for example, Allard, 1983]. Secondly, if radionuclides migrate from the breached container to the vicinity of the unbreached container they may alter the solution surrounding it. The second effect is

---

[1] Experimental conditions of 3.3 Mrad/hr were used by Glass [1986(1), 1986(2)] and conditions ranging from 1×10^4 to 2×10^5 Rads/hr were used by Van Konynenburg [1986].

important if a breached container is in the vicinity of an intact container. This may lead to the acceleration of degradation of the unbreached container, and subsequently greater possible releases. As an upper bound for the dose rates that may be expected, data from Lundgren [1982], as modified by Christensen [1982], for dose rates near spent fuel are used. Table 2.2 gives the estimates of Christensen [1982].

Table 2.1   Homogenized Densities of Unconsolidated Spent
Nuclear Fuel for Gamma Radiation Field Calculations

Material	Homogenized Density (g/cc)
U	1.65
O (from UO_2)	0.44
Zr	0.36

Table 2.2   Dose Rates on the Surface of Fuel Pellets after
Various Storage Times

Dose Rates in rad/s

Time (y)	40	100	300	1000	10^4	10^5	10^6
BWR α	28	23	15	7	1.5	7.5×10^{-2}	$3. \times 10^{-2}$
β	10	6.9	4.5	2.1	0.45	1.7×10^{-2}	$9. \times 10^{-3}$
PWR α	32	26	17	8	1.7	8.6×10^{-2}	3.4×10^{-2}
β	12	8.3	5.4	2.5	0.54	1.4×10^{-2}	1.1×10^{-2}

2.4 Nuclear Waste Container

The nuclear waste container (hereafter, the container) is the single most important barrier to the containment of the waste. The failure of the container exposes the nuclear fuel elements to the surrounding environment and thereby allows release. Some have claimed that the cladding of the spent nuclear fuel will also play a major role in the containment of the waste [Rothman, 1984]. However, as discussed in the following section, predicting the long-term behavior of this barrier may be too uncertain to rely upon it as an additional safety barrier. The container will have to meet all of the containment criteria, but in the event of a failure of the container, the cladding would provide a margin of safety. This philosophy would give the design a measure of conservatism rather than casting doubt on the reliability of the safety systems.

To meet the containment criteria, the DOE proposes to use a highly-corrosion resistant metal alloy [DOE, 1987]. The candidate alloys currently being discussed and evaluated for the container are Stainless Steel alloys 304L, 316L and 321 (L indicates low carbon content, which is a desirable characteristic with regard to the susceptibility of the material to intergranular attack and stress corrosion cracks), and Incoloy 825. These materials alloys (see Table 2.3) of iron, nickel, and chromium and have been used successfully in nuclear power plant applications. The thickness of the material required depends upon the amount of material needed as a corrosion barrier and presumably some minimum structural support as well. The results of preliminary corrosion testing of

these alloys given are in Table 2.4. As shown in the table (i.e. if the average corrosion rates are multiplied by 1000 y, the result is the number of micrometers of penetration expected in this time, e.g. 304L @ 100 °C = 1.02 cm in 1000 years), if general corrosion were the only mode of degradation of these alloys, all of the materials would make suitable containers for the waste using only a centimeter or two of material.

The more insidious side to the use of the austenitic alloys is the possibility of non-uniform modes of degradation that may rapidly breach the protective containment barrier. Stress corrosion cracking (both intergranular and transgranular) and intergranular attack are the nonuniform mode of most concern [DOE, 1987]. Transgranular stress corrosion cracking usually is associated with the presence of chloride ions and a tensile stress field. The repository will certainly have chloride ions, and there is a good possibility of residual tensile stresses that arise from welding the package.

Stress corrosion cracking requires the concurrent presence of: (1) a susceptible material, (2) a tensile stress, and (3) an agressive environment. Intergranular attack in these alloys is promoted by thermal treatments, particularly welding, that result in grain boundary chromium carbide precipitation. The precipitation process results in the depletion of a narrow region (100-1000 nm), adjacent to the grain boundary, of chromium. Since the corrosion resistance of these alloys is derived from passive film formation that is facilitated by the presence of chromium, an increase in sensitivity to localized attack in these regions occurs. This

phenomena is termed "sensitization". Materials usually become sensitized as the result of heat treatments, such as welding of a material, that promote the growth of chromium carbides. The welding operation can be modified to avoid this condition, but there have been other mechanisms proposed that may lead to sensitization at the low temperatures expected in a repository [Juhas, 1984].

Stress corrosion cracking in these alloys can be either intergranular or transgranular. Transgranular cracking is usually associated with an environment that contains halides, particularly chloride, a minimum temperature of 70°C and a minimum oxygen concentration of 0.1ppm. The presence of halides in the surrounding water and of atmospheric oxygen in the unsaturated environment [see Latanison, 1969], and the changes to the chemistry due to irradiation [see Ruiz, 1988, for efforts to combat this problem in the nuclear reactor industry] virtually guarantee that the environment will be aggressive toward sensitized alloys. Intergranular stress corrosion cracking has been observed in high-temperature, oxygenated high purity water and is aggrevated by the presence of a sensitized microstructure.

The final criterion with regard to stress corrosion cracking is the presence of a tensile stress. Again, the welding operation may result in residual tensile stresses in the material. Stress relief of the individual containers after welding may be necessary to avoid these residual stresses.

It has been demonstrated that for at least one of the alloys tested (304), as part of the ongoing investigations to evaluate

container materials, stress corrosion cracking occurs when radiation is present [Juhas, 1984] (see Figure 2.3). Tests were conducted at 90 °C with three different regions in the test vessel; a pure steam-air region; a steam-air- rock region; and a water-rock region. A dose rate of 1×10^5 rad/hr of cobalt-60 radiation was used to simulate the radiation field from the nuclear waste. The specimens in Figure 2.3 were taken from the steam-air-rock region. The cracking is shown to be intergranular. It appears that the cracking is occuring extensively throughout the specimen. In these same tests, the candidate alloy 304L showed no signs of cracking. Testing simulated a repository environment under the most extreme conditions that are expected. The other alloys have yet to be tested.

Although these preliminary results may be encouraging, experiences in the reactor industry indicate that materials originally thought to be resistant did crack after long exposure periods. These studies are admittedly [Juhas, 1984] incomplete and no other site specific testing has been published to date to assess the cracking issue.

The possibility of accelerated corrosion phenomena coupled with uncertainties concerning the exact mechanisms involved make it paramount that the characteristics of the environment be known.

Table 2.3   Composition of Candidate Nuclear Waste Container
Alloys

| Common alloy designation | UNS[b] designation | Chemical composition (wt % percent)[c] | | | | | | | Other element |
		Carbon	Manganese	Phosphorous	Sulfur	Silicon	Chromium	Nickel	
304L	S30403	0.030	2.00	0.045	0.030	1.00	18.00 20.00	8.00-12.00	N: 0.10 max
316L	S31603	0.030	2.00	0.045	0.030	1.00	16.00-18.00	10.00-14.00	Mo: 2.00-3.00 N: 0.10 max
825	N08825	0.05	1.0	Not specified	0.03	0 5	19.5 23.5	38.0-46.0	Mo: 2.5 3.5 Ti: 0.6-1.2 Cu: 1.5-3.0 Al: 0.2 max

[a]Information adapted from ASTM specifications A-167, B424 (ASTM, 1982).
[b]UNS designation from Unified Numbering System for Metals and Alloys (SAE, 1977).
[c]The values given are maximums except where ranges are given.

Table 2.4 Preliminary General Corrosion Testing of Candidate
Alloys

Alloy	Temp (°C)	Time (h)	Medium	Corrosion rate ($\mu m/yr$)[b]	
				Average	Standard deviation
304L	50	11,512	Water	0.133	0.018
316L	50	11,512	Water	0.154	0.008
825	50	11,512	Water	0.211	0.013
304L	80	11,056	Water	0.085	0.001
316L	80	11,056	Water	0.109	0.005
825	80	11,056	Water	0.109	0.012
304L	100	10,360	Water	0.072	0.023
316L	100	10,360	Water	0.037	0.011
825	100	10,360	Water	0.049	0.019
304L	100	10,456	Saturated steam	0.102	(c)
316L	100	10,456	Saturated steam	0.099	(c)
825	100	10,456	Saturated steam	0.030	(c)
304L	150	3,808	Unsaturated steam	0.071	(c)
316L	150	3,808	Unsaturated steam	0.064	(c)
825	150	3,808	Unsaturated steam	0.030	(c)

[b]Average of three replicate specimens of each alloy in each condition.
[c]Not determined.

Figure 2.4   Cracking Developed in 304 Stainless Steel While Tested in Simulated Repository Conditions Under Irradiation

2.5 Spent Nuclear Fuel

As briefly discussed in Chapter 1, the waste forms will be spent nuclear fuel elements, predominantly from PWR's and BWR's. An analysis of the repository receipt rate, given the projected inventory [MIT, 1988], indicates that the minimum age of the fuel that can be emplaced is approximately 16 years old. The thermal and radiation analyses have assumed that the waste will be 10 years old, so they are conservative due to the 6 year decay time that is not taken into account in the calculations. The decision as to whether or not to consolidate[1] the fuel has not been made yet [DOE, 1987].

The fuel is currently being stored at the reactor sites in either spent fuel pools or in dry storage casks. The failure rate for current fuel elements is approaching the goal of 0.01 to 0.02 percent for new fuel, but the failure rate of older fuel may be an order-of-magnitude higher failure percentage rate [Frost, 1982]. A review performed by Rothman [1984] concludes that the fuel will not undergo significant degradation during the 300 to 1000 years of storage. This review is based upon experience with Zircaloy in autoclave tests and limited experience with dry storage of irradiated fuel. Many of the modes of degradation of spent fuel are dismissed in this review without solid evidence to support such a decision. One type of degradation that may be significant when

---

[1] Consolidation is the dismantling of the fuel assemblies to allow them to be packed closer together and theoretically allow more fuel to be put into each container.

radiation is present is that of hydriding. The fact that significant alpha radiolysis would be occurring in the event of a breach (Section 2.3.2) leads to increased levels of hydrogen that may form hydrides. As noted by Rothman, this issue is not fully resolved. If spent fuel is to be considered as one of the safety barriers to radionuclide release, much more experimental work is needed with actual spent fuel and not just Zircaloy studies.

Rothman's review also does not address the fact that the fuel to be emplaced will have to undergo a significant amount of handling and transportation. One would expect that the handling of literally millions of these rods would result in many of types of failures not currently observed in the spent fuel. With a large enough number of failed rods, the presence of alpha radiation (even at 1000 years, as seen in Section 2.3.2) may play a significant role in the further degradation of the cladding and the magnitude of the release.

In the event of a breach of the container intact cladding will shield the encroaching ground water from the alpha and some of the beta radiations. The failed fuel elements will allow contact of groundwater and the bare fuel elements, with the accompanying alpha and beta radiolysis of the solution. The greater the number of fuel elements failed, the greater the dose to solution. In long-term studies of radiation effects, it is critical to know how many fuel elements may be failed to accurately assess the potential impacts from a radiation point of view.

3.0 RADIATION EFFECTS

This chapter examines the radiolysis interaction, depicted in Figure 2.2. All of the interactions related to equilibria and interaction of the radiolysis products is discussed in Chapter 4. Chapter 4 also ties together all of the concepts presented in Figure 2.2.

Two principal changes occur when materials are used in radiation environments. The first is the direct damage of the material being used by collisions of the radiation with lattice atoms and the subsequent displacement of these atoms. The second type of change, and the one under consideration for this work, is the interaction of the radiation with the aqueous environment in contact with the materials.

The discussion of radiation effects is divided into two sections that describe first, the physical interaction of the radiation that results in the deposition of energy in the solution and the production of chemical species. The second section discusses the chemical interactions of species produced by the energy deposited as a result of the radiation.

3.1 Passage of Radiation Through Aqueous Media

In nuclear waste package and nuclear reactor systems there is a wide range of types of radiations that are encountered. In waste package systems the radiation types of concern are high-

energy photons, alpha particles and beta particles. High energy photons are of concern to the design of the primary barrier (in the case of the proposed repository in tuff rock, this would be 304LSS, 316LSS, or Incoloy 825 [DOE, 1987]) since they would pass through this barrier and affect any ground water near the package. Alpha and beta particles are of more concern in the unlikely event that the primary barrier is breached (since the particles have very short ranges in most materials, they cannot penetrate the primary barrier while it is still intact) and ground water comes in direct contact with the waste.

In nuclear reactor systems, the most important radiations are high energy photons and neutrons. As will be explained later, the interactions with solutions are through electronic interactions. Since neutrons are neutral particles they do not interact directly with the electrons. Neutrons interact with water molecules by colliding with the hydrogen nuclei thus transferring energy and ejecting the hydrogen from the molecule. Ejected hydrogen nuclei are charged (high energy protons at this point) and deposit energy to the medium through electronic interactions. The electronic interactions and subsequent chemical transformations to aqueous solutions is termed radiolysis and is described in more detail below.

It is well known that charged-particle (α, β, p, ...) and photon (gamma and x-ray) radiations deposit energy to the medium through which they are passing by coulombic interaction with the electrons of the medium [Evans, 1955]. For all of the above radiations, the primary interactions release secondary electrons

that lose energy through the same physical mechanisms. The final result is a cascade of electrons and secondary photons that excite and ionize the medium. In aqueous solutions, the assumption is made that all the deposited energy goes into the excitation and ionization of water molecules (this will be termed ionizing, but excitation is implied.) Excited water molecules decompose into a host of chemical species[1]:

$$H_2O ==> H_2O_2, \; HO_2, \; H, \; OH, \; e^-, \; H^+, \; OH^-, \; H_2 \qquad 3.1$$

Amounts of each of the above species that are produced depends upon the ionization density (this is usually differentiated in terms of linear energy transfer (LET)) of the particular radiation. The spectrum of possible LET has been categorized into three distinct classes based upon the geometric nature of the energy distribution of the ionizations [Mozumder, 1966]. The three classes are spurs (photon and beta particles), blobs (protons) and short tracks (alpha and recoil particles); they are depicted in Figure 3.1. The significant differences between these classes result from the proximity of the interactions. The spurs produced by betas and photons are widely separated and thus the probability of interaction of radicals, in seperated zones produced by the radiation, with each other is minimized. The net result is solvation of the radical species by diffusion into the bulk solution. Therefore, the solution is exposed directly to species produced by

---

[1] The non-molecular species in unusual valency states are termed radicals, i.e. HO_2, H, OH, e^-, H_2

the radiation and the primary yields (i.e. those yields that can be thought of as homogeneous distributions in solution) are higher for radicals than for molecular products. There is little variability in the yield with changing the energy of the incident photons or electrons [Schwarz, 1966].

The other extreme from the spur-type reaction is the short tracks produced by alphas and recoil nuclei. More radicals are produced in close proximity to each other [LaVerne, 1986] for these higher LET radiations and therefore significant interaction can occur prior to the solvation of the species in the aqueous medium. The blobs produced by proton irradiations are intermediate to these two cases. Blob and short-track radiations favor the production of the molecular species (e.g., H_2O_2 , H_2) rather than radicals.

Unlike the low LET radiations, the observed yields from ion irradiations vary with particle energy. The result is an increase in the total number of species produced rather than a change in the type of species produced. Numerical values for the yields of the various species (expressed as number of species produced per 100 ev of deposited energy) are given in Tables 3.1, 3.2, 3.3 and 3.4. a comparision of the numerical values given in Tables 3.1 (spur type interactions), 3.2 (blob type interactions), and 3.3 (short track type interactions) support the geometric assumptions discussed in the beginning of this section for the classifications of yields. The experimental techniques used to generate these data sets are discussed in the next section.

Figure 3.1   Schematic of the Distributions of Energy By Various
Particles

Spur Interactions - Widely Seperated Interactions
Associated with beta and gamma radiations

Blob Interactons - Seperated Densly Ionized Interactions
Associated with Proton and Deuteron Radiations

Short Tracks - Densly Ionized Interaction with Little Seperation
Associated with Alpha and High Energy Charged Particle Radiations

Table 3.1 Gamma (and Beta) Radiolysis Yields (species/100 ev) at Low(25- 90 °C) Temperatures

$G(e^-)$	$G(H^+)$	$G(H_2O_2)$	$G(OH)$	$G(HO_2)$	$G(H)$	$G(H_2)$
2.7	2.7	0.61	2.872	0.026	0.61	0.43

Table 3.2   Fast Neutron (P+ and D+) Yields

LET or

neutron   energy

	G(e-)	G(H+)	G(H_2O_2)	G(OH)	G(HO_2)	G(H)	G(H_2)
4ev/A[1]	0.93	0.93	0.99	1.09	0.04	0.50	0.88
2 Mev[2]	0.15	0.15	0.95	0.37		0.41	0.855
18 Mev[3]	1.48	1.48	0.91	1.66		0.64	0.68
Fission[4]	0.8	0.8	1.27	0.68		0.45	0.99
4ev/A[5]					0.08		

[1]Burns, 1976
[2]Gordon, 1983, at high temperatures, T > 100 °C
[3]Appleby, 1969
[4]Katsumura, 1988
[5]LaVerne, 1986

Table 3.3   Alpha Radiolysis Yields

LET or

alpha energy

	$G(e^-)$	$G(H^+)$	$G(H_2O_2)$	$G(OH)$	$G(HO_2)$	$G(H)$	$G(H_2)$	
4-5 Mev[1]		0.30	0.30	1.30	0.50	0.10	0.30	1.40
32 Mev[2]	0.72	0.72	1.00			0.42	0.96	
12 Mev[3]	0.39	0.39	1.08			0.27	1.11	
^{244}Cm[4]	0.13	0.13	0.98	0.18	0.35	0.5	1.28	
^{244}Cm[5]	0.13	0.13	0.92	0.44	0.11	0.14	1.17	
^{244}Cm[6]	0.06	0.06	0.985	0.24	0.22	0.21	1.3	

[1]Gray, 1984
[2]Schwarz, 1966
[3]Schwarz, 1966
[4]Bibler, 1974
[5]Burns, 1981
[6]Christensen, 1982

Table 3.4   Gamma and Beta Radiolysis Yields at High
Temperatures (> 100 °C)

$G(e^-)$	$G(H^+)$	$G(H_2O_2)$	$G(OH)$	$G(O)$	$G(H)$	$G(H_2)$
0.4	0.4	0.0	0.7	2.0	0.3	2.0[1]
3.2	3.2	0.57	5.3	0.0	2.4	0.44[2]
3.2	3.2	0.6	4.7	0.0	3.4	1.2[3]

[1]Burns, 1981
[2]Pikeav, 1988
[3]Katsumura, 1988

3.2   Experimental Determination of Yields

The experimental determination of yields is made using a technique known as pulse radiolysis. The experimental setup used by Burns [1981] for determining yields in the temperature range of 25 to 400 °C is shown in Figure 3.2. In this setup, water is flowing through the main reaction vessel, where the radiolysis is occuring, the irradiated water is run through a cooler and then to an analysis system. The analysis is usually performed with optical absorbtion and other spectrographic techniques (part of the "analysis system" not pictured in Figure 3.2.) Schuler [1987] presents a good review of the history of the spectrographic techniques used to determine the rate constants and yields. The resolution of the techniques is on the order of nano- to picoseconds. This is more than adequate for the processes being modeled in this analysis.

Direct measurements are not routinely made to determine yields of the radical species (molecular species are measured directly, though.) Instead, a scavenger species is introduced to interact with particular radicals. The yield of the products of the reaction of the scavenger with the radicals is measured directly and determines the yield of the radical indirectly. The method that Burns employs to measure the yields (Table 3.2) of the reducing radicals H· and e^-_{aq} are made in saturated nitrous oxide (N_2O) solutions and the yield of N_2 is measured from the following reactions:

$$H_2O + e_{aq} + N_2O = N_2 + OH^- + OH^-$$ 3.2

$$H^- + N_2O = N_2 + OH^-$$ 3.3

Thus the yield $G(N_2)$ is expected to measure $G_{e^-} + G_H$. As a means of differentiating between these two yields, methane is often used to remove H:

$$H + CH_4 = H_2 + CH_3$$ 3.4

In this case the yield of nitrogen, $G(N_2)$, is a measure of G_{e^-}.

An alternative method was used by Pikeav (and Katsumura) to make determinations similar to those of Burns. Instead of a nitrous oxide solution to determine the reducing species, Pikeav used a solution of Fe(II) in 0.4 M H_2SO_4. The yield of Fe(III) is given by:

$$G(Fe^{3+}) = 3(G_H + G_{e^-}) + G_{OH} + 2G_{H_2O_2}$$ 3.5

By combining this with a materials balance of water radiolysis, or

$$G(-H_2O) = G_H + G_{e^-} + 2G_{H_2} = G_{OH} + 2G_{H_2O_2}$$ 3.6

gives the following dependence of reducing radical yield on Fe(III) and H_2 yield:

$$G(Fe^{3+}) = 4(G_H + G_{e-}) + 2G_{H_2} \qquad 3.7$$

Both $G(Fe^{3+})$ and $G(H_2)$ are measured directly and, therefore, the radical yields are determined. To determine other yields, Pikeav used a solution of $Cr_2O_7^{2-}$ (Katsumura used ceric sulphate but the rationale is the same as that of Pikeav) which interacts with the radicals to produce the following:

$$G(-Cr_2O_7^{2-}) = \frac{1}{6}\left[G_H + G_{e-} - G_{OH} + 2G_{H_2O_2}\right] \qquad 3.8$$

Again, by combining this equation with the balance equation for water radiolysis, the following two yields are determined:

$$G_{OH} = G_H + G_{e-} + G_{H_2} - 3G(-Cr_2O_7^{2-}) \qquad 3.9$$

$$G_{H_2O_2} = \frac{3}{2}G(-Cr_2O_7^{2-}) + \frac{1}{2} G_{H_2} \qquad 3.10$$

In general, both of the above techniques should provide the same results. At room temperature this equivalence has been widely demonstrated [e.g., Schwarz, 1966; Burns, 1981; Pikeav, 1988]. The two sets of results given in the previous section have some significant differences that are probably not due to the

differences in the type of reagent used to determine yields. A discussion of the discrepancies is given in the next section.

Figure 3.2 Experimental Setup of Burns [1981]

arrangement of cobalt sources around the reaction vessel

3.3 Discussion

In light of discrepancies in some of the experimental results, the following discussion attempts to rationalize and explain. The literature is rich with information on the evaluation of gamma yields, and it appears that discrepancies can be resolved. The following sections discuss possible resolution of the discrepancies.

3.3.1 Gamma Yields

The yields described in Section 3.2 are generally for aqueous solutions at room temperature and there is a general consensus as to the numerical values listed in Table 3.1. Up to a temperature of 100 °C, the yields are practically independent of temperature [Pikeav, 1988]. As the temperature is increased beyond 100 °C, the values published for the yields differ somewhat. For gamma irradiations, Burns et al. [1981] obtained the following distribution of products at a water density of 0.45 kg dm^{-3}, 300 °C:

$$2.7 \ H_2O \implies 0.4 \ e^- + 0.4 \ H^+ + 0.3 \ H + 0.7 \ OH + 2.0 \ H_2 + 2.0 \ O \qquad 3.11$$

These results are in contrast with the more recent results calculated from work published by Pikeav [1988]:

$$5.87 H_2O \implies 3.2e- + 3.2 \ H^+ + 2.4 \ H + 5.3 \ OH \qquad 3.12$$
$$+ \ 0.44 \ H_2 + 0.57 \ H_2O_2$$

Results from a recent Japanese study [Katsumura, 1988] also show results for high temperature yields similar to that of Pikeav:

$$6.4H_2O \implies 3.2e^- + 3.2H^+ + 3.4H + 4.7OH \qquad 3.13$$
$$+ 1.2H_2 + 0.6H_2O_2$$

The discrepancy between the above results can, in part, be traced back to the experimental method of Burns and the method by which data were generated for this yield determination. The system used by Burns was a flowing system. Figure 3.3 shows the yield of hydrogen $(G(H_2))$ as a function of flow rate. The fact that the yield shows a strong dependence on flow rate is highly suspect since the equilibration time of the reactions from which the yield should be derived is on the order of microseconds[Dorfman,1974]. No plausible argument was arrived at to explain why flow rate should have any affect at all. In fact, when a set of the data from Burns is linearly extrapolated to zero flow rate, the yield becomes precisely the same value as that obtained by Pikeav. Therefore, in this work, the data of Pikeav has been used at the reference yields at temperatures from 100 to 300 °C.

Figure 3.3 Yield of Hydrogen G(H$_2$) as a Function of Test Flow
Rate

G-value vs Flow Rate from Burns, et al.

y = 0.4383 + 0.0733x   R = 0.99

4.0 THEORY OF THE INTERACTION OF RADIOLYSIS PRODUCTS

This chapter discusses the theory applied to arrive at a comprehensive model of radiation chemistry. The discussion of the last chapter is supplemented with discussions of chemical kinetics, temperature dependencies, chemical equilibria, materials interactions and transport considerations. The complete theoretical model of radiation interactions is presented in 4.9. Both theoretical and experimental aspects are discussed where appropriate.

4.1 Chemical Reactions

All of the species produced by radiation are highly reactive. Subsequent interaction of the radical species occur through classical chemical kinetics [Fontijn, 1983][1]. The chemical kinetic interactions of water radiolysis products have been so extensively studied that an entire data center has been established, at the University of Notre Dame, to compile the available reaction rate information [Beilski, 1985, Anbar, 1973; 1975; Buxton, 1978; Farhataziz, 1977; Ross, 1979]. A homogeneous chemical kinetic model of the interaction of the species has been adopted to model the reactions. The species chemically interact with each other and with the constituents of the solution to produce other chemical

---

[1]contains the details of collision and transition state theories from which simpler kinetic expressions are derived

species and to recombine into water. The types of interactions that occur and the rates at which they occur are determined by the principles of chemical kinetics. The rate of reaction is based upon the proximity of the various reacting species to each other within the media. The probability that various species will interact is proportional to the product of the concentrations times a rate constant [Denbigh, 1978]. The applicable types of reactions and the associated rates at which they proceed are:

REACTION	RATE	
Unimolecular		
A ==> B + ...	$R = -k[A]$	4.1
Bimolecular		
A + B ==> C + D + ...	$R = -k[A][B]$	4.2
Catalytic bimolecular		
A + B ==> C + B + ...	$R = -k[A][B]$	4.3
	$d[B]/dt = 0$	
Catalytic Trimolecular		
A + B + C ==> C + D + ...	$R = -k[A][B][C]$	4.4
	$d[C]/dt = 0$	
Trimolecular		
A + B + C ==> D + E + ...	$R = -k[A][B][C]$	4.5

where k represents the reaction rate constant for the particular reaction. Reactions involving more than three molecules are so highly improbable [Fontijn, 1983] (unless water molecules are

involved) that they are neglected. In general for aqueous solutions, water molecules are considered ubiquitous and are not necessary in the formal evaluation of the rates of reaction of the various radical species. All concentrations are normalized to moles per liter of solution. If gas phase species interact with each other, the rate constant must be adjusted to reflect the volume of the gas rather than that of the solution, i.e.:

$$k_{input} \left[\frac{liter_{H_2O}}{mole - s} \right] = k_{actual} \left[\frac{liter_{gas}}{mole - s} \right] \left[\frac{V_{H_2O}}{V_{gas}} \right] \qquad 4.6$$

Rate constants in the gas phase are sometimes given in terms of molecules rather than moles, so a check of the reported rate constant's units is important.

An example of the formulation of chemical kinetic equations is given in the first of the benchmark cases presented in the Appendix E. This case covers bimolecular, catalytic bimolecular, and catalytic trimolecular reaction sequences.

A compilation of reaction rate data important to the radiolysis of aqueous media is provided in Appendix A. Most of the data were taken from the above-mentioned documents obtained from the Notre Dame Data Center. Additional data were taken from numerical studies involving water and air radiolysis. The data sets from the other numerical studies usually have their origins in the Notre Dame work.

4.2 Experimental Determination

The reactions that occur with the radicals are extremely fast (rates ~$1.0\text{x}10^{10}$ molar^{-1}-s^{-1}, generally, see Appendix A). Accurate measurement of these reaction rates is generally carried out by pulse radiolysis as was discussed earlier. These methods have been improved to the point where picosecond resolution is routinely possible [Dorfman, 1974; Schuler, 1987]. The reactive species are monitored *in situ* using optical absorption techniques. A typical experimental setup for the measurements is shown in Figure 3.2 and is discussed in Chapter 3. This setup is identical to the type used to obtain the data on yields presented in the previous chapter.

4.3 Temperature Effects

Temperature effects are treated in two distinct ways; the radical species are calculated using an Arrhenius temperature dependence, while the solubility products are calculated using the Criss-Cobble method for the temperature dependence [Criss, 1964]. These two methods and the implementation are discussed below.

4.3.1 Temperature Dependence of Radical Interactions

As with the data for yields, it is important to be able to determine the changes in the reaction rates as a function of temperature. Burns proposed a method of assigning Arrhenius

expressions and associated activation energies for most reactions. The usual expression for the Arrhenius temperature dependence is:

$$k = k_0 \exp\left[\frac{E_a}{R}\left[\frac{1}{T_o} - \frac{1}{T}\right]\right] \qquad 4.7$$

where E_a is the activation energy, k_0 is evaluated at a reference temperature T_o, R is the universal gas constant, and T is the desired temperature . Arrhenius behavior up to 150 °C was confirmed by Christensen [1987], for reactions involving the hydrated electron. Behavior up to 230 °C for the hydroxyl radical interactions has also been measured by Christensen [1983], as well as by Fontijn [1973] for a wide range of reactions. Burns [1981] assigned numerical values of the activation energies based upon the assumption that the reactions were aqueous diffusion controlled. For most fast reactions ($>10^8$ l-mole^{-1}-s^{-1}), a value of 12.6 kJ/ mole was assigned. Reactions with low rate constants (i.e., on the order of $1C^5$ to 10^8 l-mole^{-1}-s^{-1}) were assigned an activation energy of 18.8 kJ/mole on the assumption that they have low activation barriers. The data bears this out, as most of the measured activation energies [Christensen, 1987, 1983, 1981; Fontijn, 1983] are similar to those given by Burns [1981]. Fontijn [1983] demonstrates some deviation of the Arrhenius temperature dependence, but this only occurs at very high temperatures (T > 1000 °K). Activation energies for particular reactions are given in Appendix A along with the reaction rate.

4.3.2 Solubility-Product Temperature Dependence

The temperature dependencies of the major chemical (i.e. non-radical) species have been more extensively studied, and a more rigorous formulation has been devised to predict solubility constants. The method employed in this thesis is to compute the solubility constants for a particular temperature using the Criss-Cobble method and not to use the Arrhenius behavior described above. The desired quantity for predicting high temperature behaviors is the free energy of formation (ΔG°) at particular temperatures. The most useful function for this calculation is given by:

$$\Delta(\Delta G^\circ) = -\Delta S^\circ_{T_1}\Delta T + \Delta \ddot{C}^\circ_p|_{T_1}^{T_2} * \left[\Delta T - T_2 \ln\frac{T_2}{T_1} \right] \quad 4.8$$

where ΔS° is the entropy, ΔC°_p is the average heat capacity between T_1 and T_2. However, the data for ΔC°_p are scarce at high temperatures. In an attempt to remedy this problem, Criss [Criss, 1964] introduced the linear-correspondence principle of entropy (S°) between 25 °C and T_2:

$$S^\circ_{T_2} = a_{T_2} + b_{T_2} S^\circ_{25}(\text{abs.}) \quad 4.9$$

The coefficients a and b can be obtained with good reliability [Ahluwalia, 1964] and the entropy at T_2 is used in the following equation to obtain the average molar heat capacity:

$$\bar{C}^\circ_{P_2}\Big|^{T_2}_{25} = \frac{S^\circ_{T_2} - S^\circ_{25}}{\ln\left[\dfrac{T_2}{298.2}\right]} \qquad\qquad 4.10$$

The values obtained from this equation are substituted into the free energy equation to obtain free energies at high temperatures. This is the essence of the Criss-Cobble extrapolation of free energy data to high temperatures.

4.4 Chemical Equilibria

In the past [e.g., Christensen, 1982], very simplified data sets have been employed to address the full range of hydrolysis and oxidation/reduction reactions that are probably occurring (i.e. Christensen considers only Fe^{2+} and Fe^{3+}, and neglects the hydrolized forms of iron: $Fe(OH)^+$, $Fe(OH)_2$, $FeOH_2^+$, $Fe(OH)^{2+}$) . This thesis employs a thermodynamic database in an attempt to better describe incorporate the knowledge that we have on relevant chemical equilibria into radiation chemistry models.

Since the equilibria must be expressed in terms of forward and reverse rates of reaction, an arbitrary assignment of a reaction rate is made for the forward reaction (unless one is available) and the reverse reaction rate is adjusted to give the appropriate

chemical equilibrium constant. Equilibrium constants are taken from Barner [1979]. Barners' data were derived from published data from the National Bureau of Standards (NBS) and extrapolated up in temperature using the Criss-Cobble method [Criss, 1964], discussed in the previous section. Very little has been done in the way of confirming the absolute values of most of the high-temperature equilibrium coefficients.

The determination of the solubility constant is made with the following equations:

[Reactants] $<\Longrightarrow>$ [Products]

where the solubility constant (K^{sp}) is expressed as:

$$K^{sp}_{reaction} = \exp\left[-\frac{\Delta G^{\circ}_{products} - \Delta G^{\circ}_{reactants}}{RT}\right] \qquad 4.11$$

where the free energies of formation (ΔG°'s) are taken from the appropriate temperature values in Appendix C. Using the determination of K^{sp}, the forward and reverse reactions are determined from:

$$\frac{k_f}{k_r} = K^{sp}_{reaction} \qquad 4.12$$

Since the forward (k_f) and reverse (k_r) reactions are inversely correlated, an assignment of one of the rates determines the other.

In the absence of data, arbitrary assignment is made to one of the rates (within a range of physically plausible choices) and the other is calculated to give the appropriate solubility constraint.

As an example, a complete solubility reaction is calculated from basic quantities to demonstrate the above procedure. Consider the following equilibrium reaction:

$$Ni^{2+} + H_2O ==> NiOH^+ + H^+ \qquad 4.13$$

The free energies of formation at 25°C given in Barner [Barner, 1979] for the reactants and products are:

$$\Delta G_{NiOH^+} = -54.4 \text{ Kcal/mole}$$

$$\Delta G_{Ni^{2+}} = -10.9 \text{ Kcal/mole}$$

$$4.14$$

$$\Delta G_{H_2O} = -56.7 \text{ Kcal/mole}$$

$$\Delta G_{H^+} = 0.0 \quad \text{Kcal/mol}$$

Subtracting the free energies of the reactants from those of the products and substituting into the equation for the solubility constant:

$$K_{reaction} = e^{\left[-\dfrac{13.2 \times 10^3 \frac{cal}{mole}}{\left[1.987 \frac{cal}{mole \cdot °K} \right] * (298°K)} \right]} = 2.1 \times 10^{-10} \qquad 4.15$$

Comparative values of this solubility constant [Bard, 1985] range from 3.2×10^{-7} to 2.3×10^{-11}, but most of the values are reported near 5×10^{-10}. The above value is reasonably close to the reported values and thus provide some confidence in utilizing this approach for calculating solubility constants.

4.5   Gas Phase Partitioning

For gas phase partitioning of volatile species produced by radiation, an approach has been used that is similar to that employed by others [Ibe 1985a, 1985b], except for the fact that convective effects as described by Ibe [1985] are not relevant to the repository system.   Henry's Law coefficients account for the partitioning of the volatile species between the vapor and aqueous phases:

$$K_H = \frac{P_{gas}}{[\text{concentration in solution}]} \qquad 4.16$$

where P_{gas} is the partial pressure of a particular gas-phase species that is soluble in solution.   Kinetically, the process is described by two exchange reactions: one from the liquid to the gas phase, and the other from the gas to the liquid.   The ratio of the forward to reverse reactions is the volume-averaged Henry's Law coefficient.

As with the equilibria, an arbitrary assignment is made for one of the reaction rates (unless the actual rate is known) and the opposite reaction is adjusted to provide the correct value of the Henry's Law coefficient. This is achieved using the following relation for volume and temperature:

$$k_f = k_r \left[\frac{V_l}{\left[V_l + V_g \left[\frac{K_H}{RT} \right] \right]} \right] \qquad 4.17$$

Here, k_f refers to the rate of transfer from gas to liquid and vice versa for k_r. The Henry's Law coefficient us usually expressed in atmospheres per mole fraction. This equation is essentially a mole balance that accounts for the total number of molecules in the system. V_l and V_g are the liquid and gas volumes, respectively, R is the gas constant, and T is the absolute temperature.

4.6 Materials Interactions

The interaction of materials within the system with species produced by radiation is generally handled using chemical kinetics expressions in the same manner as the reaction of radicals. The actual nature of the interactions must be determined *a priori*, and empirical expressions for corrosion rates can be accommodated. Precipitation reactions are likewise treated with chemical kinetics; ideally, at dynamic equilibrium, precipitation should be the same as the rate of corrosion. The overall solubility of a metallic species is

determined by the solid phase chosen for the precipitation process. It is important to choose relevant solid species so that solubilities reflect the real system.

The function used to describe a corrosion interaction utilizes the Butler-Volmer equation [Newman, 1973]. The current is related to the flux through the following relation [Newman, 1973]:

$$\text{Flux} = -D\frac{dC}{dx} = z\,F\,i\,(10^3) \qquad\qquad 4.18$$

where z is the charge of the ion, F is the Faraday constant (96487 coulombs/mole), i is the current in amps per cm^2, and the factor 10^3 is to account for the molar volume in liters. Current is related to the concentrations in solution through the electrode potential via the Tafel equation:

$$i = i_o \exp\left[\frac{\beta F}{RT}E\right] \qquad\qquad 4.19$$

where β is proportional to the slope of the lines on an Evans diagram (Figure 4.1) and E is now calculated using the Nernst Equation:

$$E = E_o + nF \ln\frac{C_{\text{reductants}}}{C_{\text{oxidants}}} \qquad\qquad 4.20$$

where n is the number of moles of charged species, and C is an activity (assumed equal to the concentration). The appropriate

oxidants and reductants for the system are chosen *a priori* to correspond with the major constituents of the solution [Morel, 1983]. Given an appropriate chemical system, all of the redox couples should indicate the same potential and, therefore, even if the couple chosen does not dominate the redox behavior, the concentrations should still reflect the potential.

It is important to note that for the materials being considered, the interaction of metals plays a small role with regard to radiation interactions. If a material were actively corroding and influencing the solution chemistry on the time scales of interest in radiation chemistry, the material would most likely be too reactive for consideration as a container material in a repository system. Recent studies [Marsh, 1987, 1983, 1988; Walton, 1987; Taylor, 1984] have investigated radiolysis effects for carbon steel containers. Carbon steel is expected to corrode and consume many of the oxidants produced by radiolysis. For the candidate materials chosen by the DOE (see Chapter 2) very little corrosion is expected and therefore the oxidants produced by radiolysis may accumulate in solution. In the event that a stress corrosion crack or pit is formed on the container, the environment will more aggressively attack the material in this localized place and rapidly fail the container. The modeling capabilities developed in this thesis are designed to allow for the prediction of just how aggressive the solution may be toward the container rather than simply allowing the container to uniformly corrode and consume the oxidants. Examples related to the cases of actively corroding and passively corroding materials are presented in Chapter 7.

4.7 Transport

In practical studies, heterogeneties often must be considered. Often this involves reactive surfaces, two phase environments and barriers to aqueous movement. To accommodate these issues, the model was developed with the ability to treat spatially-dependent problems in one dimension. This was represented by classical incompressible, diffusive - convective theory in one dimension (x):

$$\frac{\partial C_i}{\partial t} = D_i \frac{\partial^2 C_i}{\partial x^2} + u_i \frac{\partial C_i}{\partial x} \qquad 4.21$$

where C_i is the concentration, D_i the diffusion coefficient, and u_i the velocity. This rather simple transport formulation has very few restrictions with regard to parameter dependencies and the implementation of additional spatially-dependent effects, e.g. electromigration. In the current form of the transport equation, electromigration can be accommodated as a drift velocity [Newman, 1973], u_i:

$$u_i = z_i v_i \, F \frac{d\phi}{dx} \qquad 4.22$$

making the assumption that the potential drop (the differential term in the above equation) is linear in the spatial dimension, i.e. the drop from point to point is linear.. In the above equation, F is Faraday's constant, z is the charge and v is the ionic mobility.

Concurrent work is being performed to better represent the relevant aspects of this transport mechanisms in the modeling [Psaila, 1989].

4.8 Experimental Aspects of Radiation Effects

One of the main considerations for the corrosion of metals is the electrochemical shifts induced by radiation. Potential shifts can change the environment that the metals are exposed to from one of passivation to one of potential pitting. In the case of stress assisted cracking, the potential shifts may increase the rate of attack of the metals following the rupture of any passive films. The review article of Airey [1973] presents some of the best descriptive material on the subject of electrode behavior in the presence of radiation. Although the material is descriptively correct, there was an inconsistencies in the modeling aspects of the paper when compared with experimental results that is presented in Chapter 7 and supplemented by the work presented in this thesis.

Briefly reviewing the relevant portions of Airey's work, we start by considering iron, steel, and nickel electrodes in acid solution. The results at low temperature (i.e. room temperature) point out two major influences of radiation. First, the electrode responds to a different cathodic reaction, e.g. involving H_2O_2 vs H^+, this is most evident on an Evans diagram that gives the corrosion potential (E_{corr}) at the intercept of the anodic and cathodic reaction lines (see Figure 4.1 [Latanison, 1985]). In this case, a higher corrosion potential also indicates a higher corrosion current (i_{corr})

although if the material passivates at the higher potential, this is not always the case. The higher potential caused by the radiolytically-produced species is further evidenced in the potential scans performed by Glass [1985] and Kim [1987] on candidate waste package materials (Figures 4.2, and 4.3.) These scans show a distinct shift in the corrosion potential with radiation. It is interesting to note that the pitting potential, the point where passive films break down, does not change significantly with and without irradiation. This can be seen as a decrease in the metals passive region and an increase in the susceptibility to pitting attack [Glass 1986]. The observations of Figures 4.2 and 4.3 are explained through Figure 4.4, adapted from Greene [1962], which shows the effect on a passive metal (such as those proposed to be used for the repository) subjected to two different cathodic reactions. The measured E_{corr} is shifted upward (corresponding to E_{corr} of H_2O_2 in Figure 4.1) yet the pitting potential remains the same (i.e. the cathodic reaction has changed, not the anodic one). The schematic representation is generally the same phenomenon being observed in Figures 4.2 and 4.3.

The other notable effect observed in the experiments conducted to simulate a tuff repository [Glass, 1986] is the similarity in the effect of radiation and hydrogen peroxide on the electrochemical behavior of the materials being tested. This effect is depicted in Figure 4.5 which shows the results when a specimen is subjected to both peroxide and irradiation. It appears that the irradiation curve asymptotically approaches the curve subsequently achieved using peroxide. This is strong evidence that

the hydrogen peroxide generated during irradiation is controlling the potential of the solution. It is for this reason that the sensitivity calculations, discussed below, focus on the concentration of hydrogen peroxide

Figure 4.1   Simplified Evans Diagram Adapted from Latanison,
[1985]

$H_2O_2 + 2e\text{-}$

$H^+ + 2e\text{-}$

E'_{corr}

E_{corr}

$Fe \implies Fe^{2+} + 2e\text{-}$

i_{corr} i'_{corr}

Potential

Current

Figure 4.2 Electrochemical Behavior of 304L SS in a Radiation
Environment

Figure 4.3 Electrochemical Behavior of Ti-30Mo in a Radiation Environment

Figure 4.4   Interpretation of Shift in Corrosion Potentials, adapted from Greene, 1962.

Figure 4.5 Potential Scan of 304L SS in a radiation
Environment and as a Result of Hydrogen Peroxide
Addition

4.9   Model of Radiolysis Interactions

This section collects all of the physical equations presented in Chapter 3 and the previous sections of this chapter into a comprehensive expression of the theoretical model for radiation chemistry. The mathematical description of radiolysis interactions with aqueous solutions is a system of partial differential equations that describe the source terms, chemical kinetics, and transport. The mathematical theory of the radiolysis modeling is described below for the prediction of solution concentrations of the various radiolysis products. The form of the equations is that of convective mass transport with homogeneous chemical reactions and source terms [Bird, 1960]. The equation is presented in a form found to be useful for the radiolysis calculations, but the model can easily be modified to include various dependencies of the parameters on the concentrations, time and position, as well as additional terms such as electromigration. Using C_i to describe the concentrations of the species we are predicting, the system of equations is represented as:

$$\frac{\partial C_i}{\partial t} = \sum_R G_i D_R \qquad \text{Production}$$

$$+ \quad D_i \frac{\partial^2 C_i}{\partial x^2} + u_i \frac{\partial C_i}{\partial x} \qquad \text{Transport} \qquad 4.23$$

$$+ \sum_{j=1}^{nrtn} k_j \mu_{ij} \prod_{n=1}^{neq} C_{in}^{|\mu_{in}|} \qquad \text{Kinetics}$$

where

i is the number of the individual species (of quantity neq),

R is type radiation,

j is the reaction number,

n is synonymous with i.

G is the production rate of species i for radiation type R,

D_r is the dose rate of radiation type R.

D_i is the diffusion coefficient of species i,

u is the velocity of the medium,

nrtn is the total number of reactions,

k is the reaction rate constant of reaction j,

μ is the stoichiometric coefficient for species i in j.

The large sigmas represent summation, and the pi represents the factorial function over the index n. The G-values, rate constants, and stoichiometric coefficients are determined experimentally. A discussion of these quantities and the experimental determination was given previously in this chapter and in Chapter 3.

In addition to the above equations, initial and boundary conditions are needed to complete the description of the model. In general, the initial conditions are used to define different possible initial states of the waste package system. The boundary conditions are used to simulate the behavior of materials degradation or other phenomena. The most common boundary conditions for our purposes are:

$$\frac{dC}{dx}\Big|_{boundary} = 0 \qquad \text{No Flux Boundary}$$

$$C = constant \qquad \text{Constant Concentration} \qquad 4.24$$

$$\frac{dC}{dx}\Big|_{boundary} = f(C,\eta) \qquad \text{Concentration Dependent Flux}$$

The concentration-dependent flux can be used to describe the corrosion reaction.

Additionally, as an aid speeding up the solution of the above system of equations and for use in solving the sensitivity analysis equations of the next section the Jacobian matrix is needed. The following expression gives the exact differential for the Jacobian in a single spatial dimension, x, assuming no convection, u=0:

$$\frac{d}{dC_j}\left[\frac{dC_i}{dt}\right] = \frac{d}{dC_j}\left[D_i\frac{d^2C_i}{dx^2}\right] + \sum_{m=1}^{nrtn} k_m\mu_{im}\mu_{jm}C_j^{|\mu_{jm}|}\prod_{n=1;\ n\neq j}^{neq} C_n^{|\mu_{mm}|} \qquad 4.25$$

The symbols are the same as those provided above. This expression is exact for pure reaction problems, and approximated in the first term on the right-hand side for spatial problems.

4.10 Sensitivity Analysis

In addition to the radiolysis model, a sensitivity analysis model was also developed to aid in the understanding of the complex systems being analyzed by the radiolysis model described above. Sensitivity analysis is needed to obtain a understanding of

the importance of many possible interactions with little *a priori* knowledge of how they interrelate. The method that is presented here was chosen because it gives an analytical representation of the sensitivitivities of the system without requiring a large number of extra solutions of the radiolysis model equations as would be used in a Monte Carlo analysis.

There is extensive associated with the development of sensitivity theory for systems of equations [Dickinson, 1976; Cukier, 1973, 1975; Piepho, 1981; Caucuci, 1982, 1981a, 1981b, 1984; Oblow, 1978, 1983a, 1983b; Worley, ; Kramer, 1982; R. A. Christensen, 1981]. This work has not yet been applied to the radiation chemistry problem as of this writing [see for instance, H. Christensen, 1988]. Three general categories exist of methods for performing sensitivity analyses on systems of equations; the fourier amplitude sensitivity test [Cukier, 1975, 1973], The direct method (DM) [Dickenson, 1976; Oblow, 1983] and the adjoint methods [[Piepho, 1981; Worley, ; Caucuci, 1981, 1981, 1982, 1984; Oblow, 1983]. Each of the above has advantages and disadvantages, but for large systems of equations the adjoint (or importance) method using linear approximations has many advantages [Worley, ; Oblow, 1983].

The most straightforward description of the adjoint sensitivity analysis method is given by Piepho [1981]. The method has been adapted for automatic implementation if the system of equations is solved using the methods described in the Section 5.2.1. The unique aspect of adjoint implementation is that the solution of the radiation chemistry models only need to be

performed once. Once this is done, all the sensitivities of the system can be calculated from the results [Worley, ; Oblow, 1983]

The derivation of the theoretical equations given in Piepho [1981] are recapitulated in Appendix D to provide a complete reference to the work being performed. The results of the derivation used in the numerical assessment are discussed below. The following discussion assumes a system of partial differential equations with appropriate initial and boundary conditions.

For each run of the sensitivity analysis we need to define a response, i.e. a quantity whose sensitivity to the various parameters we would like to investigate. Mathematically, the response, $\Re$, is chosen as:

$$\Re = \int_0^{t_f} L(C,\eta,t)\, dt \qquad\qquad 4.26$$

where L is the response rate and η is the parameter vector. The response is typically taken to be the concentration of a particular species, i.e., we want to know what parameters affect a certain concentration. Therefore, the response rate is simply the time derivative of the concentration, as given in Section 4.9. The adjoint functions for this system are defined as:

$$\delta\Re = \sum_{i=1}^{n} C_i^*(t')\, \delta C_i \qquad\qquad 4.27$$

where the adjoints C_i^* represent the change in response due to a small change in C_i at time t'. The definition of Lewins [1965] is helpful for the interpretation of the adjoint (which he calls importance):

"The Importance, $N^+(x,t)$, is defined as the expected or probable contribution of one particle [molal in our case] at x at time t to meter the reading at time t_f. Thus a particle is "important" to the (future) observable reading"

The following system, used to calculate the adjoints, was obtained by manipulating the system of equations in Section 4.9 assuming small change in the process variables of the last equation (see Appendix D):

$$\frac{dC_i^*}{dt} = -\frac{dL}{dC_i} - \sum_{k=1}^{n} C_n^* \frac{df_k}{dC_i} \qquad 4.28$$

The term within the summation is simply the adjoint times the Jacobian matrix that was explicitly defined for the radiation model calculation. The notation:

$$f_k = \frac{dC_k}{dt} \qquad 4.29$$

is used in the above expression for simplification. In the radiation model calculation, the Jacobian is used to achieve convergence and

its exact value is not needed. However, in our application of the method, the exact Jacobian is needed since it is explicitly used to define the time rate of change of the adjoints. One of the main reasons that this type of analysis can be performed automatically is due to the fact that this Jacobian is defined explicitly. The system of equations for the adjoints is subject to the initial conditions (remember that the calculation is being performed in reverse time, see Appendix D, so the initial condition is defined at the final time):

$$C_i*(t_f) = 0.0 \qquad\qquad 4.30$$

In other words, events that occur at or after the final time do not factor into the sensitivity analysis.

The above set of ordinary differential equations is solved to yield the adjoint functions C_i*. From these functions, the total sensitivity with respect to any of the process variables and/or parameters can be obtained. The expression to solve for the total sensitivity is given as:

$$\frac{\delta \Re}{\delta \alpha} = \sum_{k=1}^{n} \left[C_k*(0) \, \frac{dC_k(0)}{d\alpha} \right] + \int_0^{t_f} \left[\frac{\partial L}{\partial \alpha} + \sum_{k=1}^{n} \left[C_k* \, \frac{\partial f_k}{\partial \alpha} \right] \right] dt \quad 4.31$$

where a is any of the parameters of the system, process variable, or time. The fractional sensitivity is calculated by multiplying the above expression by α and dividing by $\Re$. The fractional sensitivity

times the relative change in α gives the relative change in $\Re$ due to that change.

4.11   Total Sensitivity Functional Formulations

This section derives some quantities required in the solution of the total sensitivity.   In particular, the differential terms within the integration of the total response are discussed.   The parameters considered are rate constants, dose rates, G-values, diffusion coefficients, activation energy, and temperature.   This derivation assumes that the response is a concentration of a particular species, therefore the functionals are just derivatives of the function given in Section 4.9.   Many other formulations could be considered, e.g., combinations of parameters and concentrations.   There are virtually no restrictions with regard to the responses, although the formulation is much simpler if the response is explicitly defined by the concentrations and parameters.

The variable x is used to represent both the response rate L and the rate of change of concentration, dC/dt.   The variability with respect to G-values and dose rates are computed using the simple expressions:

$$\frac{dx_j}{dG_i} = D_R \quad \text{if } i = j; \ = 0.0 \text{ otherwise}$$

$$\frac{dx_j}{dD_R} = G_j \qquad\qquad 4.32$$

The total sensitivity with respect to temperature and activation energy is calculated using:

$$\frac{dx_j}{dT} = \sum_{i=1}^{nrtn} k_i \mu_{ij} \frac{E_a}{RT^2} \prod_{k=1}^{neq} C_k^{|\mu_{ik}|} \qquad 4.33$$

and

$$\frac{dx_j}{dE_a} = -\sum_{i=1}^{nrtn} k_i \mu_{ij} \left[\frac{1}{RT}\right] \prod_{k=1}^{nrtn} C_k^{|\mu_{ik}|} \qquad 4.34$$

The above formulations neglect the temperature dependence of the dose rate, diffusion coefficients and G-values. If these dependencies are deemed important, they can easily be accommodated. The last parameter considered is the diffusion coefficient:

$$\frac{dx_j}{dD_i} = \frac{\delta^2 C_j}{\delta x^2} \quad \text{if } i = j; \quad 0.0 \text{ otherwise} \qquad 4.35$$

The δ's indicate that the derivatives are approximations rather than exact values. It should be noted that the sensitivity calculations are only performed at a boundary in spatially dependent problems. This is due to computing limitations rather than anything related to the theory.

5.0 NUMERICAL METHODS

This chapter describes the numerical implementation of the theory presented in the last chapter. The three main portions for numerical consideration are the radiolysis model equations (Section 4.9), the adjoint equations and the total sensitivity. This chapter focuses on the numerical computations and leaves the aspects of input/output and general data handling to the Users Guide (Appendix E) and the code listings, Appendix B. The numerical methods provided within each of the computer codes are discussed.

5.1 Radiolysis Model Equations

The numerical methods utilized in the solution of these radiolysis models owe much to the work of Gear [1973], who developed a very robust method of solution for ordinary differential equations. In addition to a strict chemical kinetics model, the models presented incorporate an option for modeling the transport of species.

The inclusion of transport, as devised in this thesis, has been similarly used in prediction of atmospheric radionuclide transport in two dimensions [Chang 1974] and to solve the Navier-Stokes equations, also in two dimensions [Painter, 1981]. Christensen [1982] has attempted to include a "proxy" transport in the form of separate source and sink terms for radiolysis calculations, but this is not accurate for more localized calculations. The numerical

method used to incorporate transport directly into the chemical kinetics equations, while still using the method of Gear, is termed the "Method-of-Lines" [Hindmarsh, 1981]. The discussion of these methods is provided in Section 5.2.2.1, below.

The computer code MITIRAD was developed to solve the system of equations defined in Section 4.9. The chemical reaction portion of the solution is identical to the many computer codes now being used around the world for radiation chemistry problems: for example, MAKSIMA-CHEMIST, used in Canada [Carver, 1977] (a variant of which is used in Sweden [Christensen, several citations]); SYMPHONY, used in Japan [Ibe, 1981]; FACSIMILE, used in England [Chance, 1977]; and GENKIN, used at Sandia National Laboratory [Galinas, 1974]. The Listing for MITIRAD is given in Appendix C.

5.1.1 Solving The Partial Differential Equations

The method used to solve the system of equations is a modification of Gear's method that uses backward differentiation formulas for stiff problems (stiff meaning that the characteristic rate constants for the species that are being calculating vary greatly). The basic system to be solved is given as:

$$dC/dt = f(C,t, \eta) \qquad 5.1$$

The required function, f, is the same as the expressions for the system of equations used to calculate the time rate of change of the concentration, Section 4.9, and of the adjoints (Section 4.10). A

closer look at the numerical details of the functions themselves are given in later sections. The following discussion will drop the parameter vectors notation, η, but the presence of the parameters is implied.

Typical methods for solving ODE's look for an estimates of the dependent variables based upon a fixed-point Newton method iterative scheme:

$$C_{n+1} = \sum_{i=0}^{k-1} \alpha_i C_{n-i} + \beta_0 h\ f(C_{n+1}, t_{n+1}) \qquad 5.2$$

where k is the order (typically, $1 < k < 12$), α and β_0 are coefficients dependent upon the order only, h is the step size, and the subscripts correspond to the solution at a particular time point (i.e., n+1 is the next time to be solved for, and n-i are the previous solutions up to time n+1). Because of the stiffness of the reaction-transport systems in the radiolysis models, the above iteration fails to converge for reasonably large time steps. Therefore, the method of Gear that employs a modified Newton iteration is used to overcome the time step restrictions of a fixed point iteration. Gear's method begins by utilizing the above equation with all of the terms summing to zero to allow a Newton-Raphson iterative scheme to be applied:

$$C_{n+1} - \sum_{i=0}^{k-1} \alpha_i C_{n-i} - \beta_0 h\ f(C_{n+1}, t_{n+1}) = 0 \qquad 5.3$$

Applying the Newton-Raphson scheme [Chua, 1975] yields the following iterative equation used to solve for C_{n+1}:

$$C_{n+1}^{(j+1)} = C_{n+1}^{(j)} - \left[I - h\beta_0 J(C_{n+1}^{(j)}, t_{n+1}) \right]^{-1} * \left[C_{n+1}^{(j)} - h\beta_0 f(C_{n+1}^{(j)}, t_{n+1}) - \sum_{i=0}^{k-1} \left[a_i C_{n-i} \right] \right] \qquad 5.4$$

where I denotes the identity matrix and J is the Jacobian matrix. The superscript j is indicative of the current guess (or the supplied initial value); we are seeking a more refined estimate, $j+1$, of the concentration at time $n+1$. The degree of refinement is determined through a tolerance check of the jth and $(j+1)$th estimates. Failure to converge results in a decrease in the step size and a reevaluation as above.

The Jacobian matrix used in the above iteration is the partial derivative of the function f, with respect to all of the concentrations:

$$J(C,t) = \frac{\partial}{\partial C}[f(C,t)] \qquad 5.5$$

Since the this matrix is used only to refine the current estimate, only an approximation of it need be used to solve a given set of equations. In the sensitivity analysis described in Section 4.10, the Jacobian matrix of the system of equations that describe the change in the concentrations must be accurately computed as part of the function f, to solve the adjoint system. Fortunately, the Jacobian

matrix can be calculated explicitly, not only for the concentrations, but for the adjoints as well.

By supplying the functions for the system of ordinary differential equations, f, the Jacobian matricies J and a set of initial values for each of the variables, C, Gear's method can then solve for the variables at any point in time. The numerical implementation of this portion of the model involves the setup of the set of equations in such a way that they can be solved using a tested (i.e., in general use) package for the solution of ordinary differential equations. The package selected was LSODE, developed by Alan Hindmarsh of Lawrence Livermore National Laboratory [Hindmarsh, 1983]. Specific details regarding the numerical implementation are found in the LSODE write-up (Appendix C) and in Hindmarsh [1981, 1983]. The main variation to LSODE that is utilized is the way the inversion of the term involving the Jacobian matrix is handled. LSODE solves the matrix using either a full matrix inversion technique, a banded matrix technique (when appropriate) or a sparse matrix technique. The full and banded matrix techniques are included in the subroutine LSODE, both are accomplished using appropriate LINPACK routines [Dongerra, 1979]. The sparse matrix inversion techniques use a similar subroutine called LSODES. The sparse matrix inversion techniques are from the Yale Sparse Matricies Package [Eisenstat, 1977a,b].

5.1.2   The   Radiolysis   Function

The system of ODE's that were described in Section 4.9 are modeled using the exact chemical kinetics expressions along with an approximation to the tranport term that uses the "method of lines". The method of lines involves simply the discretization of the spatial terms using an appropriate differencing technique. The spatial mesh is divided into M uniform subdivisions but this is not necessarily required. If we are tracking N species and using centeral differencing, we arrive at the following system of N*M ordinary differential equations:

$$\frac{dC_i^j}{dt} = \sum_R G_i D_r + \frac{D\left[C_i^{j-1} - 2C_i^j + C_i^{j+1}\right]}{dx^2}$$

$$+ \sum_{\rho=1}^{nrtn} k_\rho \mu_{i\rho} \prod_{n=1}^{req} C_n^{|\mu_{n\rho}|}$$

5.6

where j indicates a spatial node, and i is a particular species. The above system is implemented with a convective term available as an additional option:

$$u_i\left[C_i^{j+1} - C_i^{j-1}\right]$$

5.7

Three boundary conditions are implemented that need no modifications to utilize automatically(constant concentration, constant flux, and zero flux). The boundary conditions are

implemented by setting the boundary node to one of the following relations:

$$\frac{dC_i^B}{dt} = 0 \qquad \text{constant/zero concentration} \qquad 5.8$$

$$\frac{dC_i^B}{dt} = \frac{2\,D_i}{dx^2}\left[C_i^{j+1} - C_i^j\right] + \text{production} + \text{reactions}$$

$$5.9$$

$$\text{zero flux, i.e. } C^{j+1} = C^{j-1}$$

5.1.3 The Jacobian Matrix Evaluation

As described in Section 5.2.2, the Jacobian matrix is useful in evaluating the radiolysis model and essential in the formulation of the adjoint equations. A numerical differentiation of radiolysis function was developed that exactly evaluates the reactions and approximates the transport terms. The numerical form of the equation solved is:

$$\frac{d}{dC_w^x}\left[\frac{dC_i^j}{dt}\right] = \sum_{\rho=1}^{NRTN} k_\rho \mu_{i\rho} \mu_{w\rho} C_w^{|\mu_{w\rho}-1|} \prod_{n=1}^{NEQ} C_n^{|\mu_{n\rho}|}$$

$$5.11$$

$$+ \frac{D_i}{dx^2} \qquad \text{if } w=i \text{ and } x = j+1 \text{ or } j\text{-}1$$

$$- \frac{2\,D_i}{dx^2} \qquad \text{if } w=i \text{ and } x = j$$

If the Jacobian is evaluated at a boundary node, the value is equal to zero if the boundary is a constant concentration. If the boundary is zero or constant flux, the last two terms of the above equation become:

$$\frac{2\,D_i}{dx^2} \quad \text{if } w = i \text{ and } x = j+1$$

$$-\frac{2\,D_i}{dx^2} \quad \text{if } w = i \text{ and } x = j$$

5.12

5.2 Adjoint Equation Solution

The numerical implementation of the sensitivity analysis is almost identical to that of the forward solution of the original equations. The equations are of the same form as those described in Sections 5.2.1 and 5.2.2 and are solved using the same solvers, LSODE or LSODES. The differences involve the functions that are being integrated and the representation of the values from the forward calculations.

5.2.1 Function to Calculate the Adjoints

The key difference between solving the radiolysis equations and the adjoint equations is the replacement of the radiolysis and Jacobian functions used by the solver to for radiolysis by the functions to calculate the adjoint and its Jacobian. The adjoint function is integrated backward in time to get the adjoint functions.

The adjoint function calls the Jacobian routine of the radiolysis equations to generate the following set of ODE's:

$$\frac{dC_i{}^*}{dt} = -J\left[C_{ia}, C_i\right] - \sum^{NEQ} C_n{}^* J\left[C_n, C_i\right]$$

5.13

where $J[C_n, C_i]$ is the Jacobian matrix term:

$$J\left[C_n, C_i\right] = \frac{\partial}{\partial C_i}\left[\frac{\partial C_n}{\partial t}\right]$$

5.14

which is evaluated using the Jacobian implementation of the previous section. Since the solution routine chooses arbitrary time steps during the solution, a functional form of the forward solutions is needed to evaluate the C_i's to be used in the call to the Jacobian subroutine. This is accomplished using spline fits to the forward calculations, described in Section 5.3.3. As with the forward solution, the Jacobian of the new function is used to achieve convergence of the solution. This is described in the next section.

5.2.2   Jacobian Matrix of the Adjoints

The Jacobian matrix of the adjoint solutions is, by examination of the form of the adjoint equations of Section 4.10, simply deduced as:

By transposing the solution obtained when evaluating the adjoint function, the Jacobian is easily obtained in by transposing.

This is the exact Jacobian analytically and offers interesting food for thought, since we knew the exact Jacobian for the forward solution, we were able to solve for a linear approximation to the adjoints. Now given the Jacobian of the adjoints, would it be possible to adjoint the adjoints and obtain a linear approximation to the second order variation in the forward solution? This was not pursued in this thesis but offers an interesting topic for future work.

5.2.3 Fitting the Forward Solutions

As seen in Section 5.3.1, a means of evaluating the forward solutions at arbitrary times is necessary. To accomplish this, cubic splines are fit to the logarithm of the forward solutions with respect to the logarithm of time. The splines allow interpolation to time points between those evaluated by by the radiolysis equations, and those that are needed by the sensitivity analysis to calculate the right hand side of the time rate of change of the adjoint.

The forward solutions are very well behaved on plots of log concentration versus log time and therefore we felt justified in using this type of interpolation technique. Also, in some of the radiolysis calculations, the solutions span many (>10) orders of

radiolysis calculations, the solutions span many (>10) orders of magnitude in both time and concentrations, therefore, the only practical means of handling the representation is log-log.

The numerical implementation of the spline fit involves fitting a third-order polynomial to each interval between the time points, requiring that the first derivatives be continuous and the second derivatives are zero at the interval boundaries (a so-called natural spline). For a particular interval, the equation being evaluated is:

$$C_i(t) = C_i(t_k) + b_k(t-t_k) + c_k(t-t_k)^2 + d_k(t-t_k)^3$$

$$t_k \leq t \leq t_{k+1}$$

5.16

where the constants b_k, c_k and d_k are determined for each interval. The determination of the constants by the is described by Forsythe [1977]. A spline fit is generated for each species prior to the beginning of the solution of the adjoint equations. Due to the fact that the logarithms of the values and of time are being fit, the spline does not try to fit the values to zero time. A linear extrapolation (in log-log space) from the first time point is used to evaluate points between the first time point and zero. The spline is evaluate by the above equation by sorting for the appropriate interval and determining the constants to use. If the time to be evaluated is the initial time, the initial values are used.

In some instances, the cubic splines do not provide an accurate estimate of the forward solutions. This situation usually

occurs when severe transients are being modeled without enough time points being determined by the radiolysis equations during the acutal transient. Since it is usually difficult to judge when transients will manifest themselves and it is impratical to have thousands of points for each run, an alternative means of fitting the time points is simply a linear interpolation of the log-log plots. This type of interpolation not only avoids possible problems with the spline fits, it also offers a significant enhancement in the speed of the calculation (in one case with 40 species and 80 reactions, the runtime of the adjoint solution went from 30 to 6 minutes with only a 1% difference in the results produced by the total sensitivity analysis)

Significant enhancement of the speed of the the sensitivity calculation is also obtained by extrapolating the adjoints from the smallest time point greater than zero, to zero. The concentrations of the species exhibit a step jump over the first time evaluation (i.e. going from zero to some non-zero value) that the integrator spends a significant amount of time evaluating, at this last time point. In pratice, the adjoints rarely make significant changes in this last small time segment and therefore, the extrapolation saves computing time without sacrificing accuracy. It is important to be aware of this approximation, especially if the number of points being evaluated by the sensitivity analysis equations is significantly reduced.

5.3 Total Sensitivity Equations

Once the adjoints have been obtained through the solution of the ordinary differential equations, the total sensitivity is obtained with respect to all parameters of the model. The total sensitivity equation is integrated numerically using a trapezoidal rule method. Many of the function described previously are used in the integration to determine the various quantities defined in Section 4.11.

The choice of trapezoidal rule integrationwas made after many unpredictable failures in more advanced adaptive quadrature techniques (DQUANC8 [Forsythe, 1977] and CADRE [DeBoor, 1971]). In many instances the more advanced techniques worked and provided better accuracy than obtained with the trapezoidal rule. However, the vigilence required to use these proved too cumbersome and therefore the trapezoidal rule was adopted for automatic implementation.

Numerically, the total sensitivity analysis is very straightforward. The various functions of Section 4.11 are integrated using:

$$\frac{\partial \Re}{\partial \alpha} = \sum_{intervals}^{time} \left[\frac{\left[g(t_k) + g(t_{k+1}) \right]}{2 \left[t_{k+1} - t_k \right]} \right] \qquad 5.17$$

where g represents the appropriate function of Section 4.11. The g's are

integrated for rate constants, production rates (G*D) and dose rates (D). The number of time intervals is evaluated is pre-selected using the since it is a trapezodial integration, the number of points should be at least 100 to keep the error introduced by the integration to ~ 1%.

The form of the total sensitivity calculation equations is ripe for enormous cancellation errors. This means that numbers very close to each other (e.g., 1.00000 and 0.9999999) are being subtracted and the significance in the result is beyond the precision of the calculations. In pratice this occurs all too frequently and a flag has been established in the calculation to sense this. In general, when dealing with radicals as the response function, cancellation errors will occur when evaluating the total sensitivity of rate constants.that have the radical as a reactant. The first part of the LINTY output (after some recapitulation of the input) lists a time period over which the sum of:

$$1 + C_i^*(t) = 0 \qquad\qquad 5.18$$

In most cases when dealing with radicals, this time period is the entire calculational period. All this means is that changes to the concentration of the radical, in and of themselves, will quickly be restored to the "unperturbed" state by other processes. When the output flags this type of situation, the integral sums are displayed along with the time period over which the result was insignificant. When the time period covers the entire interval, the response terms exactly cancel (as far as the calculational methods can

determine) and the residual terms are the total sensitivity. In the case of the example, the residuals for the effect of O_2 on the e⁻ population (the response contribution was insignificant over the time interval) predicted a variation of 0.879%. By applying a 10% change to this rate constant(i.e we predict a 0.10*0.0879=0.00879 fractional change) and rerunning the code, the variation in the response (in this instance e-) was 0.880%.

5.4 Computer Listings

The listings of the computer programs that are used in this thesis are as self explanatory as possible. All variables except reusable indices are defined, along with all subroutines and functions. The multiple indexing that is used to make many of the routines compact and expandable without user intervention must be mastered if modifications to the programs are to be attempted. The computer listing is much more of a working document than other parts of this thesis. The listings of Appendicies B and C include all subroutines except for LSODE/LSODES, which are generally available.

The programs were written in VAX-11 FORTRAN (Version 4.2) compiled on a MicroVAX II computer. Except for the use of NAMELIST and INCLUDE statements for I/O processing, the program conforms to FORTRAN 77 conventions. Portions of this work have been ported to a Macintosh personal computer with a minimum of effort, and the same would be expected for installation on most systems. The optimized code running with moderate

numbers of species and node points may still use considerable dynamic memory and time, so do not expect great results with the personal computers available at the writing of this thesis.

6.0 VERIFICATION

This chapter is provided to verify that the computer code is performing the calculations correctly, and also to provide an analog to the computer code that is used for the analyses. The forward solution of the equations is fairly straightforward, i.e. the integration of ordinary differential equations. The equations used for the sensitivity analyses are somewhat less intuitive and therefore the verification gives a proof of principle of the methods. The so-called Bateman equations were used for this example to provide an analytically simple, yet nontrivial, example.

6.1   Forward Solutions

The solution of the Bateman equations has been well defined [Bateman, 1910; Evans, 1955] and the derivation is included here for completeness. Also, some of the quantities derived in this section are needed for the next section. The system is defined as:

$$N_1 \implies N_2 \qquad \text{rate} = \lambda_1 = \text{Decay constant 1}$$

$$N_2 \implies N_3 \qquad \text{rate} = \lambda_2 = \text{Decay constant 2;}$$

or in differential form, neglecting the N_3 component:

$$\frac{dN_1}{dt} = -\lambda_1 t$$

$$\frac{dN_2}{dt} = \lambda_1 t - \lambda_2 t \qquad 6.1$$

Systems of ODE's such as the above are subject to initial conditions that allow for particular solutions to be obtained. The general initial conditions are defined as:

$$N_1 = N_1^{\emptyset}$$
$$N_2 = N_2^{\emptyset}$$

Setting the second condition equal to zero results in a less general solution, but key points are still demonstrated with less algebraic exercise. Starting with the solution for $N_1(t)$, dividing by N_1 and multiplying by dt gives:

$$\frac{dN_1}{N_1} = -\lambda_1 dt \qquad 6.2$$

Integrating this result from 0 to t gives:

$$N_1(t) = N_1^0 e^{-\lambda_1 t} \qquad 6.3$$

Substituting this expression into the differential equation for N_2, multiplying by the integrating factor, $\exp[\lambda_2 t]$, and rearranging gives:

$$\frac{d\left[N_2 e^{\lambda_2 t}\right]}{dt} = \lambda_1 N_1^0 e^{\left[\lambda_2 - \lambda_1\right]t} \qquad\qquad 6.4$$

The above expression is integrated from 0 to t, some rearrangement is performed and the initial condition for N_2 is applied to arrive at the well known result:

$$N_2(t) = \frac{\lambda_1 N_1^0}{\lambda_2 - \lambda_1}\left[e^{-\lambda_1 t} - e^{-\lambda_2 t}\right] \qquad\qquad 6.5$$

This result along with the result for $N_1(t)$ are essentially the quantities calculated by the forward portion of the computer code. To verify these results, a calculation was performed with λ_1 and λ_2 set to values of 0.01 and 0.02, respectively, and N_1^0 set to 2.0. The comparison of the results is presented in Figure 6.1. The points are the computer calculated results and the solid lines are the results obtained using the analytical expressions of this section. As can be seen, the agreement is excellent.

Figure 6.1   Results of Forward Solution of the Bateman Equations

Log [C]
(moles/l)

Log(Time [s])

—□— N1
══○══ N2
······△······ N3

6.2 Sensitivity Solution

The first aspect of performing a sensitivity analysis, using the methods described in Section 6.2, involves selecting a variable. The selection here is arbitrary but it could be any, or all, of the process variables. As an example we will choose to find the response to the final concentration of N_2 assuming a starting concentration of zero. This means that we want to know how N_2 varies as we change everything else in the model. In particular we look for the following two responses:

$$\frac{dN_2}{d\lambda_1} \; ; \; \frac{dN_2}{d\lambda_2} \qquad\qquad 6.6$$

Since λ_1 and λ_2 are experimentally measured, we will find the effect of small deviations (like experimental error) of the values on the final concentration of N_2. Since we have an explicit expression for N_2, we can differentiate the expression to find the exact values for the above derivatives. The exact values can then be compared to the results obtained using adjoint analysis.

6.2.1 Adjoint Determination

The first quantities that need to be derived, as explained in Section 4.10, are the adjoint functions. The adjoints give us the instantaneous change in the response; here N_2 is the response

(generally denoted by L), with respect to the other variables in the model, i.e. the N_i's:

$$\delta N_2(t_f) = N_i^*(t)\ \delta N_i \qquad\qquad 6.7$$

The system of equations that needs to be solved along with the initial conditions is generally stated:

$$\frac{dN_i^*}{dt} = -\frac{dL}{dN_i} - \sum_{k=1}^{NEQ} N_k^* \frac{df_k}{dN_i}$$

$$N_i^*(t_f) = 0.0 \qquad\qquad 6.8$$

Here, f_k is simply the expression for the kth ODE, defined in Section 4.10. The index NEQ is the number of process variables (in chemical systems, this would be the number of different chemical species being tracked.) The equation requires the above initial condition for each of the N_i^*. Since we start the integration of these equations at the final time and integrate backwards to the initial time, the above initial condition means that things that would happen in the future, $t > t_f$, have no effect on the sensitivities at t_f.

Integration of our example system requires the values of the Jacobian matrix, i.e. the df_k/dN_i term on the right-hand side of the general equation above. In matrix form the Jacobian of our simple system appears as:

$$\begin{bmatrix} & \dfrac{dN_1}{dt} & \dfrac{dN_2}{dt} \\[2ex] \dfrac{d}{dN_1} & -\lambda_1 & \lambda_1 \\[2ex] \dfrac{d}{dN_2} & 0 & -\lambda_2 \end{bmatrix} \qquad 6.9$$

Fully written out, the ordinary differential equations for the adjoints are:

$$\frac{dN_1^*}{dt} = \lambda_1 + \lambda_1 N_1^* - \lambda_1 N_2^* \qquad N_1^*(t_f) = \emptyset$$

$$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad 6.10$$

$$\frac{dN_2^*}{dt} = \lambda_2 + \lambda_2 N_2^* \qquad N_2^*(t_f) = \emptyset$$

Solving the second of these equations first, introducing the integrating factor, $e^{\lambda_2 t}$, to each side of the equation, and rearranging yields:

$$d\left[N_2^* e^{-\lambda_2 t} \right] = \lambda_2 e^{-\lambda_2 t} dt \qquad 6.11$$

Integrating the equation from t_f to 0 and applying the initial condition yields the following analytical expression for $N_2^*(t)$:

$$N_2^* = e^{-\lambda_2(t_f - t)} - 1 \qquad 6.12$$

Substituting this result into the ODE for $N_1{}^*$, introducing an appropriate integrating factor, integrating the equation from t_f to 0, and applying the initial condition for $N_1{}^*$ yields:

$$N_1^*(t) = \frac{\lambda_1}{\lambda_2 - \lambda_1} \left[e^{-\lambda_1(t_f-t)} - e^{-\lambda_2(t_f-t)} \right] \qquad 6.13$$

The adjoints provide us with the intermediate values that are needed to calculate the expressions defined at the beginning of Section 4.11. These expressions are obtained using the equations for the total sensitivity; the details are explained in the next section.

As a means of verifying this intermediate step the values used for the example in Section 6.1 are used in a run of the adjoint portion of the computer code and in the equations of this section. The results of the comparison appear in Table 6.1. The results of Figure 6.1 show a near- perfect match between those numbers calculated using the numerical procedure, and those calculated using the analytic solution. This gives confidence that the calculations are being performed correctly.

Table 6.1   Numerical Solution of The Adjoints of the Bateman
              Equations

Time	Analytic Adjoint		Computed Adjoint	
	N_1^*	N_2^*	N_1^*	N_2^*
37.5	0.2074	-.1175	0.2074	-0.1175
0.00	0.4773	-.3935	0.4773	-0.3935

6.2.2 Total Sensitivity Determination

Once the adjoints have been obtained, a general procedure
has been developed to utilize them to compute general sensitivities
of the response function with respect to the rate constants, i.e.,

$$\frac{dN_2(t_f)}{dk_i} \qquad\qquad 6.14$$

This quantity gives the variation of N_2 with respect to the
parameter k_i. The total sensitivity can then be used to judge the
physical significance of the parameter on the desired result.

6.2.2.1 Analytical Determination of Total Sensitivity

Since we calculated the exact solution for $N_2(t)$ in Section 6.1,
we can differentiate it explicitly to obtain the necessary expression
for the total sensitivity, defined in Section 4.11. The total
sensitivity expression obtained using adjoint theory should be
equivalent (to a first order approximation) to the expression
obtained by direct differentiation. This section offers a proof by
example of the entire sensitivity analysis from an analytical point
of view; this is accomplished by showing the equivalence of the
exact method and the good, yet approximate, method of adjoint
theory. Although some of this section is condescending, this
'workingmans' result gives some feel for the abstraction of the
adjoint theory for some readers.

The following section shows how this equivalence is calculated using a numerical method for the integrations rather than the analytical expressions. The need for the numerical solution is due to the fact that the analytical expressions for the N's and the N*'s cannot be easily obtained for even slightly more complex expressions than the one presented here as an example. But, the adjoint formulation for the total sensitivity is composed of elements that can be solved for numerically; and it therefore gives us a sensitivity analysis without complex analytical expressions.

The first step in calculating the total sensitivity is to obtain the exact result by simply differentiating the expression for N_2 obtained in Section 6.1 with respect to the two parameters, λ_1, and λ_2. Starting with $N_2(t)$:

$$N_2(t) = \frac{\lambda_1 N_1^0}{\lambda_2 - \lambda_1}\left[e^{-\lambda_1 t} - e^{-\lambda_2 t}\right] \qquad 6.15$$

We differentiate with respect to λ_1 and λ_2 to obtain (after some work):

$$\frac{dN_2}{d\lambda_1} = \frac{N_1^0 e^{-\lambda_1 t}}{\lambda_2 - \lambda_1}\left[\left[1 + \frac{\lambda_1}{\lambda_2 - \lambda_1}\right]\left[1 - e^{-[\lambda_2 - \lambda_1]t_f}\right] - \lambda_1 t_f\right] \qquad 6.16$$

and,

$$\frac{dN_2}{d\lambda_2} = \frac{\lambda_1 N_1^0 e^{-\lambda_2 t_f}}{\lambda_2 - \lambda_1} \left[\frac{\left[e^{[\lambda_2 - \lambda_1] t_f} - 1 \right]}{\lambda_2 - \lambda_1} + t_f \right] \qquad 6.17$$

Now we construct the expression for the total sensitivity using adjoint theory, referring back to the equation given in Section 4.11 for total sensitivity. Noting again that N_2 is our response variable, the expression for the derivative of N_2 with respect to $\Rightarrow_1$ is defined as:

$$\frac{\delta N_2}{\delta \lambda_1} = \int_0^{t_f} \left[\frac{dN_2}{d\lambda_1 dt} + N_1^* \frac{dN_1}{d\lambda_1 dt} + N_2^* \frac{dN_2}{d\lambda_1 dt} \right] dt \qquad 6.18$$

The notation using the δ instead of the standard derivative denotes the adjoint result. By substituting the appropriate values of the Jacobian matrix and the appropriate values for the N^*'s, an expression that can be integrated analytically is obtained:

$$\frac{\delta N_2}{\delta \lambda_1} = N_1^0 \int_0^{t_f} \left[\frac{\lambda_1}{\lambda_2 - \lambda_1} \left[e^{-\lambda_2(t_f - t) - \lambda_1 t} - e^{-\lambda_1 t_f} \right] + e^{-\lambda_2(t_f - t)} \right] dt \qquad 6.19$$

Integrating this expression is relatively straightforward and gives us the following result:

$$\frac{\delta N_2}{\delta \lambda_1} = \frac{N_1^0 e^{-\lambda_1 t_f}}{\lambda_2 - \lambda_1} \left[\left[1 - e^{-[\lambda_2 - \lambda_1] t_f} \right] \left[1 + \frac{\lambda_1}{\lambda_2 - \lambda_1} \right] - \lambda_1 t_f \right] \qquad 6.20$$

By inspection of the comparative equation determined exactly in the last section, it is shown that the adjoint method produces an exact result. For completeness, the same steps to solve for the second comparative derivative are given below without narrative. The method is exactly the same.

Through an exact integration:

$$\frac{\delta N_2}{\delta \lambda_2} = -\frac{N_1^0 \lambda_1 e^{-\lambda_2 t_f}}{\lambda_2 - \lambda_1} \left[\frac{-1}{\lambda_2 - \lambda_1} \left[e^{[\lambda_2 - \lambda_1] t_f} - 1 \right] + t_f \right] \qquad 6.21$$

Formulating the adjoint:

$$\frac{\delta N_2}{\delta \lambda_2} = \int_0^{t_f} \left[\frac{dN_2}{d\lambda_2 dt} + N_1^* \frac{dN_1}{d\lambda_2 dt} + N_2^* \frac{dN_2}{d\lambda_2 dt} \right] dt \qquad 6.22$$

Substituting appropriate values:

$$\frac{\delta N_2}{\delta \lambda_2} = \frac{N_1^0 \lambda_1}{\lambda_2 - \lambda_1} \int_0^{t_f} \left[-e^{-\lambda_2 (t_f - t) - \lambda_1 t} + e^{-\lambda_2 t_f} \right] dt \qquad 6.23$$

Integrating to obtain final result:

$$\frac{\delta N_2}{\delta \lambda_2} = \frac{N_1^0 \lambda_1}{\lambda_2 - \lambda_1} e^{-\lambda_2 t_f} \left[\frac{-1}{\lambda_2 - \lambda_1} \left[e^{[\lambda_2 - \lambda_1] t_f} - 1 \right] + t_f \right]$$

6.24

Here again, the exact result is obtained. The method demonstrated above has been implemented numerically to allow for more complex reactions to be considered. The next section presents a verification of the accuracy of the numerical method in reproducing the analytical result.

6.2.2.2   Numerical Determination of Total Sensitivity

Although the numerical implementation of adjoint theory was requisite for considering complex systems, it must be able to model the simple systems as a starting point. In fact, during the development of the code, this test was the benchmark that successive revisions of the code were compared against to check for proper function. The relative sensitivities of the example are calculated to within 4% of the value of the total sensitivity from the analytical solution. This is roughly the numerical error that is introduced in the numerical integration step.

7.0 RESULTS/APPLICATIONS

The tools presented in the previous chapters are used in various analyses of the effects of radiation on nuclear waste containers. The first analysis investigates some basic observations with regard to radiation effects on corrosion and applies the new tools in drawing conclusions from this data set. An expanded data base is used to evaluate the effects of radiation on the external portion of a nuclear waste package in simulated repository environments. A discussion is also presented with regard to the potential long-term issues of radiation effects in the event of a breach of the container. Mixed radiation fields of beta and alpha radiations are considered in this latter case.

This chapter details phenomena that are generally observed in well controlled experimental situations. The modeling results discussed in this chapter are only as good as the available data, no fitting of the parameters of the model are made. Improvements in the model only come with improvements in the available data base. The ability of the model to "point" to key parameters in the model make it a powerful aid to experimentalists, especially when the number of possible parameters is as large as in the radiation chemistry case. Therefore, extensions of the model to the repository environment are made to draw preliminary conclusions and demonstrate the usefulness of the model. The ability to limit the data set to a small subset of controlling reactions using the sensitivity analysis is discussed. The definitive evaluation of the

repository environment will have to wait until more detailed characterization of the expected environments are made.

7.1 Basic Interactions of Radiation With Metals

This section describes the modeling of the behavior of a simple system of iron in water and also a system showing active corrosion of stainless steel when radiation is applied. These model experimental studies conducted by Pourbaix [1974], Airey [1973] and Burns [1983]. This chemical system has been well studied because it is the system used in the Fricke dosimeter, as well as one of the methods employed by various researchers, discussed in Chapter 3, for measuring the yield. The quality of the simulation is a direct result of the high quality of the data.

7.1.1 Simulating Iron Species in a Radiation Field

This simulation assumes a deaerated solution with ferrous iron concentrations in solution at equilibrium with $Fe(OH)_2$ at room temperature. The starting potential, as measured by the Fe^{2+}/Fe^{3+} potential:

$$E = 0.771 + 0.0591 \log \left[\frac{[Fe^{3+}]}{[Fe^{2+}]} \right] \qquad 7.1$$

was set to 0 V (SHE). The initial pH was set at 7 after the work of Pourbaix. The simulation was run until the calculated hydrogen peroxide concentration was relatively constant with time. The data sets employed are the water and iron data given in Appendix A. These reactions represent interactions of water and iron species, including appropriate hydrolysis reactions of iron.

The results of the simulation are summarized in Table 7.1 along with the experimental results of Pourbaix [1974] and Burns [1983] and the results that Airey [1973] would calculate using a pH of 3.4:

$$E = 0.85 - 0.0591pH \qquad\qquad 7.2$$

The work of Pourbaix was not under irradiation but simulated the radiation environment using hydrogen peroxide. The explanation for the behavior of the system was deduced from the sensitivity analysis results. The major means of production of H_2O_2 is through the primary yield of H_2O_2. The destruction of H_2O_2 is through its reduction by Fe^{2+} and H. The specific reactions are:

$$Fe^{2+} + H_2O_2 ==> Fe^{3+} + OH^- + OH \qquad\qquad F10$$
$$H \ \ + H_2O_2 ==> OH + H_2O \qquad\qquad W19$$

The equation numbers refer to the appropriate equations in Appendix A. The more subtle result is that the H concentration is predicted by the sensitivity analysis to be dominated by the interaction with Fe^{2+} and Fe^{3+} through the reduction reactions:

$$Fe^{3+} + H ==> Fe^{2+} + H^+ \qquad\qquad F14$$

$$Fe^{2+} + H ==> FeH^+ \qquad\qquad F6$$

and by its production rate. Both directly and indirectly, the concentration of ferrous species control the concentration of peroxide (i.e., by having a direct effect on the concentration as well as strongly influencing species that also interact directly with hydrogen peroxide) and hence the potential. The effect of ferrous species having a strong influence on the measured solution potential has been observed experimentally by Sunder [1987], wherein addition of ferrous ions[1] to an irradiated solution lowered the potential of a UO_2 electrode. When species such as ferrous ions are introduced, they also contribute to the reduction process, if the ferrous ion concentration is large enough. This behavior is unique to species that can be oxidized by H2O2, such as iron (Fe2+ ==> Fe3+). Nickel is very slowly oxidized to a higher oxidation state and consequently has little influence on the potential. Radiation produces both oxidizing and reducing species, the concentration of the reducing species produced by radiation is found to be controlled by the major redox active species in solution and in turn, the concentration of the reducing species controls the oxidants produced by radiation which strongly influence the potential of the solution. This is a reoccuring theme as shown in the results of Section 7.2.

---

[1]The concentration was 1×10^{-5} molar Fe^{2+}.

The shift to a lower pH (-log(H$^+$)) is indicative of the hydrolysis of both ferrous and ferric species as the potential is raised. The hydrolysis reactions are often neglected and can cause differences in the computed results since the most species concentrations are sensitive to pH, as well as the fact that the solubilities of important species change. A similar change in pH (starting at pH = 6.2) was noted without explanation by Burns [1983] for the irradiation corrosion of mild steel. Unfortunately, no potential measurements were made for these experiments.

7.1.2 Simulation of Active Corrosion

In this section a system in active corrosion, followed by passivation is modeled. The results are presented to simulate in a qualitative manner the results of Airey [1973], presented in Figure 7.2. Figure 7.2 depicts type 347 stainless steel being irradiated by Co-60 gamma rays in an acid solution. The data of interest in this figure are curve 1. The first portion of this curve is simulated by assuming a net corrosion reaction:

$$Fe + 2H_2O \Rightarrow Fe(OH)_2 + 2e^- + 2H^+ \qquad 7.3$$

By assuming this net rate, the precipitation kinetics of $Fe(OH)_2$ are not factored into the system explicitly. Physically, this restricts the system to saturation conditions with regard to ferrous species. The rate for the above reaction is relative to the surface area of the

metal, the effective volume of the liquid and the active current. The choice of rates was made by trial and error and a value of 1×10^{-4} s^{-1} was selected as the "active" rate.

The model results simulating the first half of curve 1 (the second "on" arrow from the left) are given in Figure 7.3. It of interest to note that the brief induction period (i.e. the potential stays the same after the "on" arrow) prior to the precipitous drop of the potential of curve 1 in Figure 7.2, this also appears qualitatively in the simulated results (note the log-log scale in seconds rather than minutes). The sensitivity analysis indicates that the aqueous electron population is controlled by the reactions involving oxidants produced by radiation and oxygen. This induction period is related to the period of time it takes for the electrons produced by the corrosion reaction to consume the oxidants being produced by the radiation. This period is very brief and the oxidants are calculated to disappear in approximately 1 minute. The second half of curve 1 is simulated assuming a "passive" rate of 1×10^{-8} s^{-1}. The return of the solution to its potential value during the induction as in curve 1 is properly simulated, as shown in Figure 7.4 (compare the H_2O_2 concentration at the upper plateau of Figure 7.3 with the final plateau of Figure 7.4). The passive rate of electron formation is not sufficient to consume the oxidants produced by the radiation and therefore, the potential is high.

The simulation of the passive environment mimics the behavior of stainless steel used in radiation environments. Since they do not actively corrode, the oxidants produced by radiation build up in solution causing high potentials. If a pit or stress

corrosion crack develops, the potential of the solution makes it particularly aggressive to the small anodic area and a rapid failure may result. Conversely, the "active" behavior consumes the oxidants produces by the radiation. Therefore, if a sufficient thickness of actively corroding material is used in the waste packages to both shield the solution from radiation and to provide a wastage margin (i.e., allow enough material to consume all of the oxidants), the rapid non-uniform modes of failure associated with stainless steels, such as stress corrosion cracking, would be avoided.

Table 7.1. Potential and pH Comparison for Irradiation of Ferrous
Solutions

Study	Potential (V SHE)	pH
Pourbaix	0.72	3.4
Burns	----	3.7
Airey (calculated)	0.64	using 3.4
This work	0.72	3.04

Figure 7.1 The Time Evolution of Key Species in the Irradiation of
Ferrous Solutions

Figure 7.2 Potential vs Time from Airey [1973] for Corroding 347
SS.

Figure 7.3 Simulation of the "Active" Corrosion Portion of Figure
7.2

Figure 7.4   Simulation of the "Passive" Corrosion Portion of

Figure 7.2

Log [C]
(moles/l)

Log(Time [s])

—□—	NI2+
═○═	H+
······△······	NI(OH)2
---▽---	FEOH2+
—·◇—·	FE3+
—·■·—	HO2-
—●—	H2O2
═▲═	O2

7.2 Modeling the Repository Environment

The only attempts at a better understanding of the effects of radiation on the waste package and spent fuel have been performed at Lawrence Livermore National Laboratory[Von Konynenburg, 1986, Glass, 1986]. Experimentally, the issue of the effects on the waste package has been better addressed, although incompletely, than effects on the waste form. Radiolysis effects have been acknowledged in the experiments utilizing actual spent fuel, but no attempt has been made to quantify these effects experimentally. Theoretical studies to better understand how well the current data explain the observed effects have not been performed, see quote in Section 1.3.

The two cases of particular interest are, first, the effect of gamma radiations on the environment surrounding an intact waste package and secondly, the effect of alpha and beta radiations on the solutions that may contact spent fuel in a breached container. The first case represents the early life of the container when significant gamma radiation will escape from the package (see Chapter 2) and thereby cause radiolysis interactions. The second case represents the behavior when spent fuel is contacted by water in the event of a breach. In both cases, the effects of concentration of the halides are evaluated, the systems are equilibrated with atmospheric oxygen, carbon dioxide, and for the first run, nitrogen species. The yield of fixed nitrogen was calculated from Burns' equation [1982]:

$$G(N) = 2C_oR\left[1 - e^{-1.45 \times 10^{-5}GDt} \right]$$ 7.4

This is equivalent to the reaction sequence:

$$N_2 \Rightarrow 2N \qquad k = RGD(1.45 \times 10^{-5})$$

7.5

$$N \Rightarrow \frac{1}{2}N_2 \qquad k = GD(1.45 \times 10^{-5})$$

the variables are defined as: R is the ratio of gas to liquid volume (assumed = 1), G is an experimentally determined yield (=1.9, Burns, 1982), D is the dose rate in MRad/hr and C_o is the concentration of nitrogen molecules in the gas phase. The reaction set, air reactions in Appendix A [Ibe, 1988], employed in the analysis accounts for the back reactions of N, therefore the second reaction above is not used. The first equation can be treated as a yield of N atoms and by substituting the appropriate values, this was determined to be 0.01. Since the candidate material is stainless steel, the systems are also equilibrated with $Fe(OH)_2$ and $Ni(OH)_2$ using the same reactions as in Section 7.1. Additional reactions sets used for chloride, sulphate and carbonate species are given in Appendix A. The resulting reaction set involves 136 reactions and 50 different species. Dose rates of 1×10^5 and 4100 rad/hr were used, the former being associated with the experiments of Juhas [1984] and the latter being the dose rates calculated in Chapter 2.

The analysis is specific to the causes of changes to the solution potential, i.e. the H_2O_2 and e^- concentrations, as these are seen as the most important for consideration for waste package degradation and radionuclide release. Investigation of other effects, e.g. hydrogen gas generation, methane production, etc. can be performed by a similar analysis only using a different response. Since the entirety of the data sets is unwieldy, only an initial run was made with the complete data set and by using the sensitivity analysis techniques, the reactions that do not significantly effect the response were removed for more extensive analysis with the more important species. The analysis is at 90 °C with Arrhenius temperature extrapolations for the rate constants. The equilibrium reactions used equilibrium constants calculated at 90 °C using Barner [1979] data or using values in EPRI-NP-2400 [1982]. Henry's Law coefficients were taken from Von Konynenburg [1986] or Gray [1985]. Figure 7.5 and 7.6 gives some of key species produced in the complete data set. Figure 7.6 is for 100 days to demonstrate the equilibration of the H_2O_2. Hydrogen continues to increase in this simulation because a closed system (gas to liquid volume of 100:1) was used. In the real repository, H_2 would escape the waste package region.

Since the solution is equilibrated with oxygen and hydrogen peroxide, they control the concentrations of the reducing species at a lower value than the ferrous species discussed in Section 7.1.1. At the repository pH, the solubility of ferrous species is very low, and most of the iron species are hydrated forms of ferric ions which do not react with H_2O_2. This in turn allows the H_2O_2

concentration to build to a very high level (1.1 V SHE vs 0.72 V SHE) as compared to the case when ferrous species control the potential. By comparison, Airey [1973] measured a potential of 1.05 V (SHE, actually measured 0.81 V SCE[1]) when the materials were not actively corroding.

The variability in the solubility of the metal species, discussed briefly in Chapter 4, may have moderate to significant impacts on the potential. Therefore, knowing the basic equilibrium chemistry without radiation effects becomes as important as knowing the radiolysis equations themselves, in terms of long-term predictability.

The sensitivity analysis points to the interactions of H_2O_2 and O_2 with H and e⁻ as being dominant for the back reaction of H_2O_2. They equilibrate to roughly the same concentration. This is in general related to similar reaction rates with the reductants. An additional back reaction is introduced to this system, the reduced form of carbonate, CO_3^-, does exhibit some back reaction with H_2O_2. Again, the level of CO_3^- is determined by the level of reductants available to reduce HCO_3^- to CO_3^-. These reductants are determined by their production rates and the concentrations of H2O2 and O_2.

The pH was in general very important in determining the concentrations of most of the species. The pH is dominated by carbonate interactions, the pK of water and any significant hydrolysis reactions that occur as a result of increasing the potential. In the repository system, the pH will in general be

---

[1]SCE denotes a standard calomel electrode, while SHE denotes a standard hydrogen electrode. SHE = SCE + 0.246.

buffered by the carbonates so that no significant changes are expected, or observed [Juhas, 1984].

The concentration of halides did not directly affect the concentration of peroxide, even when concentrated 100x over the nominal concentration. The lack of a direct effect is certainly not indicative of no effect, as chloride ion is well known to promote stress corrosion cracking without the need for radiation to be present [Latanison, 1969]. A more subtle effect of concentrating the halides may be the effect they have on the solubility of the gases through the salting out effect [Harned, 1958]. Since the reactions involving the solubility of O_2 were flagged as being significant, the effect of changes to the solubility of the dissolved gases should be more thoroughly studied in the specific repository environments. Again, this points to a need for a good understanding of the non-irradiation physical chemistry to truly understand the long-term effects of radiation in repositories.

The effect of air was predicted to be minimal on the determination of the H_2O_2 concentrations. The concentration of NO_3^- increases slightly (See Figure 7.5) over the course of the simulation. Production of nitrate is of importance since it may lower the pH if significant quantities are produced (essentially producing nitric acid). Experimentally, a pH change has not been observed in short term experiments [Glass, 1986]. The presence of calcuim carbonate or other carbonate species is expected to probable buffer most of the additionally produced acid over the long term.

The data for the radiolysis of nitrogen species is particularly poor. Ibe [1988] recently published G-values for N atoms of 16. These were substituted for the G-value of 0.01 from Burns [1982] to produce the results in Figure 7.7. Obviously, the G-values that work for reactor environments do not accurately depict the lower temperature repository behavior. The pH in this simulation was calculated to be ~2.5 and the ammonia concentrations were an astronomical ~ .01 molar. Even when using the lower G-value of Burns and running the simulation to a long enough time (up to a year) predicted dramatic lowering of the pH through production of nitric acid. This indicates that the precipitation/dissolution kinetics of the carbonate species are important since the experiments of Juhas [1984] did not show large pH changes. These large discrepancies in the observed and predicted results indicate that more experimental work is needed but judging from the experimental evidence so far, the effect of nitrogen species is secondary. This was echoed by a recent review on the subject for the tuff repository by Reed [1987]. Reed states that the large discrepancies can only be resolved by more experimentation. The report goes on to implicate ammonia for the potential of cracking a copper container. The ammonia concentrations predicted in the model calculations associated with Figure 7.5, were very low using the G-value provided by Burns. Juhas also observed just traces of ammonia in the irradiation tests discussed in Chapter 2, over a year long period at 1×10^5 rad/hr. The two highly varied results presented above do indicate that although not highly influential on the corrosion potential, the other products of the irradiation of air

may be a factor in materials degradation. In terms of pH, the actual value of the G-values may not be as important as the buffering capacity and the kinetics of the carbonates.

By using the sensitivity analysis to judge the importance of the various reactions, 55 reactions involving the halides and nitrogen species were removed from the original data set of 136 reactions. The results of a run without these species is shown in Figure 7.8. As predicted, the H_2O_2 concentrations is seen to be unchanged by the removal of these reactions. This is a major result of this work, i.e., being able to limit the reaction sets to the ones that demonstrate the largest effects on the quantities being studied. This type of capability has been called for in the literature [Ibe, 1988] to aid in the determination of which rate constants need further investigation. With the complicated chemical systems of the repository environment, it is virtually impossible to evaluate all of the potential interactions. To get the most from each experiment, analyses such as those presented must be used in conjunction with experimental design. The speed at which this determination can be performed with the techniques developed in this work makes the method of varying a single coefficient and re-running the code[1] [Christensen, 1988; Ibe, 1988] to determine the effect virtually obsolete.

Comparing the effect of dose rates, Figure 7.5 was at 1×10^5 rad/hr while Figure 7.9 is at 4100. rad/hr. The time is one day and

---

[1] This method was used to check the results of the sensitivity analyses for key reactions and the comparisons were excellent, see Figure 5.4 and associated discussion.

the peroxide concentration has not yet equilibrated. By running the simulation to 100 days at the lower dose rate, the peroxide concentration is shown to equilibrate at virtually the same concentration as with the higher dose rate (Figure 7.10). The competition of oxygen and hydrogen peroxide for the available reductants will always result in equilibration to the same values since the oxygen concentration is always constant in solutions equilibrated with air.

This effect is also demonstrated when the radiation is not gamma but rather alpha and beta radiations which would result from a breached container. Using the values for the PWR dose rates given in Chapter 2 for 1000 years, Figure 7.11 was generated. The time to reach the equilibrium was 1000 days, but the same concentration of hydrogen peroxide was achieved as in the gamma radiation case. The potential of the solution in this case becomes dependent upon the residence time of the solution near a breached container. The longer the solution stays in contact with the spent fuel, the greater the potential will be and for most of the actinide species their solubility will increase. Since soluble actinides emit their own radiation, if enough of them are dissolved into solution the need for direct contact with the fuel is no longer necessary to raise the potential of the solution.

This is of particular importance for the potential corrosion of unbreached containers that may be contacted by water that has spent some time near a breached container. The implications of this case have not been addressed in any previous studies of radiation effects on the waste container. Since the possibility of the

temperature being below the boiling point of water is much greater at 1000 years, this situation actually would be more probable than the unexpected contact of water with the waste packages during the period when gamma radiation is present. A complete analysis of this effect requires detailed information concerning the solubilities of the radionuclides at the temperatures expected in the repository.

The work of Wolery [1983] attempts to provide the necessary equilibrium chemistry for the repository system. A marriage of the techniques and data being used by Wolery with the techniques presented in this thesis would provide a powerful tool for investigating the long-term effects of radionuclide release and its effects on unbreached containers.

Figure 7.5 Species Chosen From Complete Data Set. H_2O_2, O_2 and NO_3^- are major species, Ni_2^+ and $NiOH^+$ are present in small concentrations, CO_3^- is the reduced form of carbonate.

—□—	H2O2
=○=	O2
·····△·····	CO3-
---▽---	NO3-
-·-◇-·-	NI2+
----■----	NIOH+

Figure 7.6 Prediction Using Ibe [1988] Data for Nitrogen Fixation
Yield

Log [C]
(moles/l)

Log(Time [s])

—□—	H+
═○═	O2
·····△······	N2G
---▽---	H2O2
—·◇—·—	NH4+
—·■—·····	N
—●—	HCO3-
═▲═	CO32-
······▼·······	OH-

Figure 7.7 Prediction of the Same Species as Figure 7.5, with Reduced Reaction Set.

Figure 7.8 Prediction Using Repository Dose Rate of 4100 Rad/hr
and Reduced Reaction Set.

Figure 7.9 Long-Term (100 days) Prediction Using Repository Dose
Rate and Reduced Reaction Set.

Log [C]
(moles/l)

Log(Time [s])

—□— H2O2
═○═ CO3-
······△······ H
---▽--- e-

Figure 7.10   Long-Term Prediction of H2O2 Under Alpha and
Beta Dose Rates for 1000 Year Old Waste

Log [C]
(moles/l)

Log(Time [s])

──□── H2O2
══o══ O2
┄┄△┄┄ CO3−

8.0 DISCUSSION, CONCLUSIONS AND FUTURE WORK

This chapter summarizes the work and discusses some of the key conclusions deduced from the work presented. A brief description of future work related to issues in the nuclear power industry is also discussed.

8.1 Discussion and Conclusions

Providing assurance that the containers for the nuclear waste will provide substantially complete containment over a long period of time is a very complicated matter. The presence of radiation only enhances the complexities of the systems that must be modeled in the long term. **(1)** Based upon the results generated for this thesis, the experimental approach that is currently considered by the DOE for degradation of waste packages needs to address the likely scenarios of irradiation of high-temperature steam by gamma irradiation[1] [Reed, 1987] and irradiation of liquid water by low-level alpha and beta radiations in the presence of container material, Section 7.2. The decision to use stainless steels as the container material is also drawn into question since they are susceptable to rapid non-uniform modes of attack, particularly at

---

[1]The key conclusion here is that no data were available to make a considered analysis of this likely scenario, other than that conducted by the nuclear industry for high purity systems which is a completely different environment than expected in the repository. If the DOE is standing by the thermal calculations, Chapter 2, the repository will most certainly be a high-temperature steam environment for hundreds of years. Experimental assessment of the candidate materials in a gamma irradiated, high-temperature (>100 C) environment is vital.

potentials predicted by the radiolysis models, even in the long-term. Consideration of a material such as a mild steel that is not susceptable to cracking, and used in sufficient thicknesses to account for its more rapid general corrosion rate may be appropriate.

(2) The analyses indicate that the potential is determined by the interaction of radiolysis products with the major constituents of the solution. The prediction of radiation effects in the repository environment is strongly affected by the ability to predict the physical chemistry of the solutions expected in the repository in the long-term. Integrating the methods of this thesis with those being developed by Wolery [1983], to predict equilibrium chemistry, would provide the necessary combination of a detailed description of the physical chemistry and an appropriate radiation chemistry model. This is particularly important for assessing the effects of radiation on the solubility of the actinides [DOE, 1987] and thus amounts that would be released.

The most likely time for liquid water to be present in the repository is after hundreds of years, therefore, the gamma radiations would be negligible. The possibility that alpha and beta radiations leaking from a breached container in the vicinity of unbreached containers[1] would have significant effects on the degradation of the unbreached containers, also needs to be addressed. (3) This effect is shown to be dependent upon the amount of time the solution is exposed to the radiations (~ 1 year at

---

[1]The DOE proposes to use several containers per borehole if the choice is made to emplace the containers horizontally.

1000 years to reach the same potential achieved in ~ 1 day from the high levels of gamma radiation expected during the very early years of the repository, Section 7.2). Therefore, ground water flow models also become important in the prediction of the total release. (4) The implication here is that although it may be desirable to have slow flow rates, when the radionuclides finally are released, higher concentrations are present. The higher concentrations would be the result of higher solubilities of the actinides at higher potentials, as well as the possibility of more packages being breached due to the more rapid degradation of the container at higher solution potentials.

In the complex chemical systems of the repository it is simply not possible to assess the impact of all the potential effects. It is very important to be able to glean from the best available data the interactions that are predicated to be the most important and target them for further study. This is not to say that other effects should not be looked for just because the available data (if there is any) does not indicate an effect. But cn the other hand, the available data cannot be ignored when important interactions are predicted. (5) The abiltiy to get the most of the available data is a key result of this work. By being able to identify key interactions, a better physical understanding of the processes involved in certain behaviors becomes evident, e.g. the role of the radiolytically produced reductants and oxygen in the control of hydrogen peroxide.is physically understandable but hardly deducible *a priori* from the hundred reactions used for the models in Section 7.2.

(6) The sensitivity analysis proves very useful in deducing physical effects and, from a more pratical point of view, it proved very useful in debugging the computer runs of the non-physical effects. Its role as a debugging tool cannot be overlooked since results in the literature if not carefully checked, contain non-physical data. As an example, the following reaction is given by Bielski [1985] in, what is usually reliable, a compilation of data produced by the Notre Dame Radiation Chemistry Data Center:

$$O_2^- + HO_2^- \implies H_2O_2 + O_2 \qquad\qquad 8.1$$

This equation contains two non-physical aspect: (1) no charge balance and (2) no material balance[1]. Since HO_2^- only exists in significant quantities at high pH, thus the impact of this reaction is not discovered unless assessments are being performed at the high pH. This reaction was flagged early when looking at the sensitivity of OH^- (charge was mysteriously going away!). Other instances of non-physical results often come from simple mistakes made in the handling of the many data sets, and species concentrations that must be used in radiation chemistry. The sensitivity analysis proves very helpful in getting the data sets properly prepared.

(7) The methodology presented is a powerful tool for both: (1) predicting the complex interactions of radiation with aqueous solutions in terms of potential effects on nuclear waste containers

---

[1] In general, H2O can be added to either side to produce the material balance without being explicitly stated since the activity of water is taken to be 1, but this is not the case here.

and (2) determining the important interactions to be targeted for further experimental investigation. These are both critical aspects with regard to the assurance of safety that must be provided by the DOE with regard to the nuclear waste repository. Combining these efforts with other investigations is seen as key to developing the complete understanding that is necessary to properly predict long-term effects.

8.2 Future Work

In addition to the repository environment, the methods developed can be directly applied to radiation chemistry problems in the nuclear industry. Water radiolysis along with convective effects in nuclear reactor cores result in waters being concentrated in oxygen from 100 to 300 ppb. This concentration of oxygen increases the susceptability of austenitic stainless steels to intergranular stress corrosion cracking (IGSCC) [Ruiz, 1988]. Currently, a program is being undertaken to by General Electric to reduce the concentration of oxygen in the water and thus lessen the chance of IGSCC. The program being undertaken is called Hydrogen Water Chemistry. The predictability of the physical processes of the system is very important to developing an effective program. Since the model developed here is based upon established laws of physics and chemistry, the application to assisting in the HWC simply requires supplying the necessary physical constants. The incorporation of spatial resolution in the model allows for localized corrosion phenomena to be accurately modeled with regard to

spatially dependent effects. This is especially important when evaluating localized aspects of cracking phenomena in reactor waters.

9.0 REFERENCES

Allard, B., H. Kipatsi and J. Rydberg. 1977. "Radiolysis of Filling Materials," KBS-TR-56, KarnBranselesakerhet, Stockholm, Sweden.

Ahluwalia, J. C. and J. W. Cobble. 1964. J. Am. Chem. Soc., 86:5381-5384.

Airey, P. L. 1973. "The Effect of Radiation on Electrode Processes," Rad. Res. Rev., 5:341-371.

Anbar, M., M. Bambenek, A. B. Ross. 1973. NSRDS-NBS 43.

Anbar, M, and F. and A. B. Ross. 1975, NSRDS-NBS 51

Appleby, A. and H. A. Schwarz. 1969. "Radical and Molecular Yields in Water Irradiated by γ Rays and Heavy Ions," J. Phys. Chem., 75(6):1937:1941.

Bard, A. J., R. Parsons and J. Jordan. 1985. Standard Potentials in Aqueous Solution, International Union of Pure and Applied Chemistry, Marcel Dekker, New York, NY.

Barner, H. E. and R. V. Scheuerman. 1979. Handbook of Thermochemical Data For Compounds and Aqueous Species, John Wiley and Sons, New York.

Bateman, H. 1910. Proc. Cambridge Phil. Soc. 15:423.

Bibler, N. 1974. J. Phys. Chem., 78:211

Bielski, B. H., D..E. Cabelli, and R. L. Arudi, 1985. J. Phys. Chem. Ref. Data, 14(4):1041-1100.

Bird, R. B., W. E. Stewart and E. N. Lightfoot. 1960. Transport Phenomena, John Wiley & Sons, New York.

Burns, W. G. and P. B. Moore. 1976. "Water Radiolysis and its Effect Upon In-Reactor Zircaloy Corrosion," Radiation Effects, 30:233-242.

Burns, W. G. and W. R. Marsh. 1981. "Radiation Chemistry of High-temperature (300-410 °C) Water," *J. Chem. Soc, Faraday Trans. I,* 77:197-215.

Burns, W. G. and P. B. Moore. 1976. "Water Radiolysis and its Effects Upon In-Reactor Zircaloy Corrosion," *Radiation Effects,* 30:233-242.

Burns, W. G., A. E. Huges, J. A. C. Marples, R. S. Nelson, and A. M. Stoneham 1982. "Effects of Radiation on the Leach Rates of Vitrified Radioactive Waste," *J. Nucl. Mater.,* 107:245

Burns, W. G. and H. E. Sims. 1981. J. Chem. Soc. Faraday Trans. 1, 77:2803.

Buxton, G. V. and R. M. Sellers. 1978. NSRDS-NBS 62

Cacuci, 1981. J. Math. Phys., 22(12):2794-2802

Cacuci, 1981. J. Math. Phys., 22(12):2803-2812

Cacuci, 1982. Nucl. Sci. Eng, 82:461-468.

Cacuci, D. G. 1984. EPRI Research Project Report 144-1, ORNL, Oak Ridge, Tennessee.

Carver, M. B., D. V. Hanley and K. R. Chaplin. 1979. AECL-6413, Chalk River Nuclear Laboratories, Chalk River, Ontario.

Chance, E. M. et al. 1977. "A computer Programe for flow and chemistry simulation and general initial value problems," AERE-R8775.

Chang, J. S, A. C. Hindmarsh, and N. K. Madsen. 1974. "Simulation of Chemical Kinetics Transport in the Stratosphere," in Stiff Differential Systems, Plenum Press, New York.

Christensen, H., B. Persson and K. Sehested. 1988. "Radiolysis of Reactor Water," Proceedings of the 1988 JAIF International Conference on Water Chemistry in Nuclear Power Plants, Tokyo, Japan.

Christensen, H. 1984. "Formation of nitric and organic acids by the irradiation of ground water in a spent fuel repository," SKB-KBS-84-12, Studsvik Energiteknik AB, Nykoping, Sweden.

Christensen, H., K. Sehested and H. Corfitzen. 1982. "Reactions of Hydroxyl Radicals with Hydrogen Peroxide at Ambient and Elevated Temperatures," J. Phys. Chem., 86:1588-1590.

Christensen, H. and K. Sehested. 1981. "Pulse Radiolysis at High Temperatures and High Pressures," Radiat. Phys. Chem., 18(3-4):723-731.

Christensen, H. and K. Sehested. 1983. "Reaction of Hydroxyl Radicals with Hydrogen at Elevated Temperatures. Determination of the Activation Energy," J. Phys. Chem., 87:118-120.

Christensen, H. and K. Sehested. 1986. "The Hydrated Electron and Its Reactions at High Temperatures," J. Phys. Chem., 90:186:190.

Christensen, H. and E. Soderman. 1984. "Radiolytic Gas Generation After Loss-of-Coolant Accident in a Boiling-Water Reactor," Nucl. Safety, 25(3):329-340.

Christenssen, H. 1978. "Bedomning Av Radiolys I Grundvatten," KBS-78, AB Atomenergi, Nykoping, Sweden.

Christensen, H. and E. Bjergbakke. 1982. "Radiolysis of Groundwater From HLW Stored in Copper Canisters," KBS-TR-82-02, Studsvik Energiteknik AB, Nykoping, Sweden.

Christensen, R. A., R. F. Eilbert, R. A. Rohrer, and G. S. Was. 1981. "Adjoint Sensitivity Analysis in Nuclear Reactor Fuel Behavior Modeling," Nuclear Engineering and Design, 66:125-139.

Chua, L. O. and P-M Lin, 1975. Computer-Aided Analysis of Electronic Circuits, Prentice-Hall, New Jersey.

Criss, C. M. and J. W. Cobble. 1964. "The Thermodynamic Properties of High Temperature Aqueous Solutions. V. The Calculation of Ionic Heat Capacities up to 200°. Entropies and Heat Capacities above 200°," J. Am. Chem. Soc., 86:5390-5393.

Croff, A. G. 1980. A Users Manual for the ORIGEN2 Computer Code, ORNL-TM-7175, Oak Ridge National Laboratory, Oak Ridge, Tennessee.

Cukier, R. I., J. H. Schaibly, and K. E. Shuler. 1975. "Study of the Sensitivity of Coupled Reaction Systems to Uncertainties in Rate Coefficients. III. Analysis of the Approximations," J. Chem. Phys., 63(3):1140-1149.

Cukier, R. I. et al. 1973. "Study of the Sensitivity of Coupled Reaction Systems to Uncertainties in Rate Coefficients. I. Theory," J. Chem. Phys., 59(8): 3873-3878.

De Boor, 1971. "CADRE: An Algroithm for Numerical Quadrature," in Mathematical Software, J. R. Rice, ed. Academic Press, Orlando.

Denbigh, K. 1978. The Principles of Chemical Equilibrium with Applications in Chemistry and Engineering, Third Edition, Cambridge University Press, Cambridge, England.

DOE (U. S. Department of Energy) 1987. Site Characterization Plan - Consultation Draft: Yucca Mountain Site, Nevada Research and Development Area, Nevada, DOE/RW-0160, Office of Civilian Radioactive Waste Management, Washington, D.C.

Dorfman, L. M. 1974. "Pulse Radiolysis," in Techniques in Chemistry, VI, G. G. Hammes, ed., John Wiley & Sons, New York.

Dickinson, R. P. and R. J. Galinas. 1976. "Sensitivity Analysis of Ordinary Differential Equation Systems - A Direct Method," J. Comp. Phys., 21:123-143.

Dongarra, J. J. et al. 1979. LINPACK Users' Guide, SIAM, Philadelphia

Eisenstat, S. C., M. C. Gursky, M. H. Schultz, and A. H. Sherman. 1977a. "Yale Sparse Matrix Package I. The Symmetric Codes," YALE-112, Yale University, New Haven, Connecticut.

Eisenstat, S. C., M. C. Gursky, M. H. Schultz, and A. H. Sherman. 1977b. "Yale Sparse Matrix Package I. The Non-Symmetric Codes," YALE-114, Yale University, New Haven, Connecticut.

Engel, R. L., J. Greenborg and M. M. Hendrickson. 1966. "ISOSHLD - A Computer Code for General Purpose Isotope Shielding Analysis," BNWL-236, Pacific Northwest Laboratory, Richland, Washington.

Evans, R. D. 1955. The Atomic Nucleus, McGraw Hill, New York.

Fontijn, A. and R. Zellner. 1983. "Influence of Temperature on Rate Coefficients of Bimolecular Reactions," in Reactions of Small Transient Species, A. Fontijn, ed., Academic Press, London, England.

Forsythe, G. E., M. A. Malcolm, and C. B. Moler. 1977. Computer Methods for Mathematical Computations, Prentice-Hall, New Jersy.

Frost, B. R. 1982. Nuclear Fuel Elements, Pergamon Press, New York.

Galinas, R. J. and R. P. Dickenson. 1974. "GENKIN-I: A General Kinetics Code: Atmospheric Applications," UCID-16577, LLNL, Livermore, California.

Gear, C. W. 1971. Algorithm 407, DIFSUB, Commun. ACM, 14:185

Glass, R. S., R. A. Von Konynenburg and G. E. Overturf. 1986. "Corrosion Processes of Austenitic Stainless Steels and Copper-Based Materials in Gamma-Irradiated Aqueous Solutions," Corrosion '86, Houston, Texas.

Gordon, S., K. H. Schmidt and J. R. Honekamp. 1983. "An Analysis of the Hydrogen Bubble Concerns in the Three-Mile Island Unit-2 Reactor Vessel," Radiat. Phys. Chem., 21(3):247-258.

Greene, N. D. 1962. "Predicting Behavior of Corrosion Resistant Alloys," Corrosion, 18:136-142.

Gray, W. J. and S. A. Simonson. 1985. "Gamma and Alpha Radiolysis of Salt Brines," in Proc. of the Materials Research Society Symp. on the Scientific Basis for Nuclear Waste Management, C. M. Jantzen, et al. eds., Materials Research Society, Pittsburg, Pennsylvania.

Guy, A. G., and F. N. Rhines. 1962. "Pourbaix diagrams, A firm basis for understanding Corrosion," Metal Treatment and Drop Forging, February:45:54.

Hindmarsh, A. C. 1983. "ODEPACK, A Systematized Collection of ODE Solvers," Scientific Computing, R. Stepleman, ed., North-Holland Publishing Co.

Hindmarsh, A. C. 1981. "ODE Solvers for Use with the Method of Lines," Advances in Computer Methods for Partial Differential Equations IV, R. Vichnevetsky and R. Stepleman, eds. IMACS.

Ibe, E., M. Nagase, M. Sakagami and S. Uchida. 1987. "Radiolytic Environments in Boiling Water Reactor Cores," Nucl. Sci. Tech., 24(3):48-54.

Ibe, E. and S. Uchida. 1985. "Numerical Techniques for Quantitative Evaluation of Chemical Reaction Systems with Volatile Species and Their Application to Water Radiolysis in BWRs," J. Nucl. Mater., 130:45-50.

Ibe, E. and S. Uchida. 1985. "A Water Radiolysis Model in a Circulating Flow System with a Boiling Region and Its Application to Hydrogen Alternate Water Chemistry of Boiling Water Reactors," Nucl. Sci. Eng., 90:140-157.

Ibe, E. and S. Uchida. 1985. "Radiolytic Aspects in Boiling Water Reactor Primary Systems: Results from Numerical Simulations and Statistical Regression Analyses," Nucl. Sci. Eng., 89:330-350.

Ibe, E. and S. Uchida. 1985. "A Water Radiolysis Model in a Circulating Flow System with a Boiling Region and Its Application to Hydrogen Alternative Water Chemistry of Boiling Water Reactors," Nucl. Sci. Eng., 90:140-157.

Jansen, G. 1987. "Expected Nuclear Waste Repository Waste Package Performance in Three Salt Formations," BMI/ONWI-655, ONWI, Columbus, Ohio.

Juhas, M. C., R. D. McCright and R. E. Garrison. 1984. "Behavior of Stressed and Unstressed 304L Specimens in Tuff Repository Environmental Conditions," UCRL-91804, Lawrence Livermore National Laboratory, Livermore, California.

Katsumura, Y., H. Shiraishi, D. Hiroishi and K. Ishigure. 1988. "Water Radiolysis at High Temperatures," ???

Kim, Y. J. and R. A. Oriani. 1987. "Electrochemical Behavior of Ti-30Moin a Gamma Radiolysis Environment," Corrosion, 43(1):56-58.

Kottwitz, D. A. 1984. Memo:"User Guide for ISOSHLD on IBM_PC," Pacific Northwest Laboratory, Richland, Washington.

Kramer, et al. 1982. SAND-82-8231, Sandia National Laboratory, New Mexico

Latanison, R. M. and R. W. Staehle. 1969 "Stress Corrosion Cracking of Iron-Nickel-Chromium Alloys," in Proceedings of the Conference on Fundamental Aspects of Stress-Corrosion Cracking, NACE, Houston, Texas, pp. 214-307.

Latanison, R.M. 1985. "Corrosion Engineering," Course Notes 3.54, Massachusetts Institute of Technoloy, Cambridge, Massachusetts.

LaVerne, J. A., R. H. Schuler, and W. G. Burns. 1986. "Track Effects in Radiation Chemistry: Production of HO2 within the Track Core in the Heavy-Particle Radiolysis of Water," *J. Phys. Chem,* 90:3238-3242.

LaVerne, J. A. and R. H. Schuler. 1986. "Track Effects in Water Radiolysis with High-Energy Heavy Ions," *J. Phys. Chem.,* 90:5995-5996.

Lewins, J. 1965. IMPORTANCE, The Adjoint Function, Pergamon Press, Oxford, England.

Lin, C. C. 1988. "An Overview of Radiation Chemistry in Reactor Coolants," Proceedings of the 1988 JAIF International Conference on Water Chemistry in Nuclear Power Plants, Tokyo, Japan.

Lundgren, K. 1982. "Radiation levels and absorbed doses around copper canisters containing spent LWR fuel," SKBF-DBS-TR-82-11, Svensk Kaernbraenslefoersoerjning AB, Stockholm, Sweden.

Marsh, G. P. and K. J. Taylor. "An Assessment of Carbon Steel Containers for Radioactive Waste Disposal," *Corrosion Science,* 28(3):289-320.

Marsh, G. P., K. J. Taylor, S. M. Sharland and P. W. Tasker. 1987. "An Approach for Evaluating the General and Localised Corrosion of Carbon-Steel Containers for Nuclear Waste Disposal," in Scientific Basis For Nuclear Waste Management VIII, Materials Research Society, Boston, Massachusetts, pp. 227-238.

Marsh, G. P., I. W. Bland, J. A. Desport, C. Naish, C. Westcott and K. J. Taylor. 1983. "Corrosion Assessment of Metal Overpacks for Radioactive Waste Disposal," DoE/RW/83.021, AERE Harwell, Harwell, England.

MIT, Course 22.33, 1988. "Design of a High-Level Waste Repository System for the United States," MITNE-281, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Morel, F. M. M. 1983. Principles of Aquatic Chemistry, John Wiley & Sons, New York.

Mozumder, A. and J. L. Magee. 1966. Radiat. Res., 28:215-231

Newman, J. 1973. Electrochemical Systems, Prentice Hall, Engelwood Cliffs, New Jersey.

NWPA (Nuclear Waste Policy Act), 1983. "Nuclear Waste Policy Act of 1982," Public Law 97-425, 42 USC 10101-10226, Washington, D. C

Ruiz, C. P., C. C. Lin, T. L. Wong, R. N. Robinson and R. J. Law. 1988. "Hydrogen Water Chemistry Modeling for BWR Applications," EPRI Research Project 2816-2, General Electric Company, San Jose, California.

Oblow, E. M. 1978. Nucl. Sci. and Eng. 68:322-337.

Oblow, E. M. 1983. ORNL/TM-8339, Oak Ridge National Laboratory, Oak Ridge, Tennessee.

Painter, J. F. 1981. "Solving the Navier-Stokes Equations with LSODI and the Method of Lines," UCID-19262, Lawrence Livermore National Laboratory, Livermore, California.

Pescatore, et al. 1988. "The Potential Uses of Lead in DOE's Waste Package Designs for the Basalt, Salt and Tuff Sites," BNL-52112 (Draft), Brookhaven National Laboratory, Upton, New York.

Piepho, M. G., K. B. Cady, and M. A. Kenton. 1981. "An Importance Theory for Nonlinear Ordinary Differential Equations," HEDL-TME 81-24, UC-79T, Tp, Hanford Engineering Development Laboratory, Richland, Washington.

Pikeav, A. K., S. A. Kabakchi and G. F. Egorov. 1988. "Some Radiation Chemical Aspects of Nuclear Engineering," Radiat. Phys. Chem., 31(4-6):789-803.

Pourbaix, M. 1974. Atlas of Electrochemical Equilibria in Aqueous Solutions, NACE, Houston, Texas.

Preuss, K. and G. M. Gordon. 1984. "TOUGH - A Numerical Model for Nonisothermal Unsatuuated Flow to Study Waste Canister Heating Effects," in Scientific Basis For Nuclear Waste Management VIII, G. L. McVay, ed., Materials Research Society, Boston, Massachusetts.

Psaila-Dombrowski, M. J. 1989 (expected). Ph. D. Thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Reed, D. T. and R. A. Van Konynenburg. 1987. "Effect of Ionizing Radiation on Moist Air Systems," UCRL-97936, Lawrence Livermore National Laboratory, Livermore, California.

Ross, F, and A. B. Ross. 1977. NSRDS-NBS 59.

Rothman, A. J. 1984. "Potential Corrosion and Degradation Mechanisms of Zircaloy Cladding on Spent Fuel in a Tuff Repository," UCID-20172, Lawrence Livermore National Laboratory, Livermore, California.

St. John, C. M. 1985. "Thermal Analysis of Spent Fuel Disposal in Vertical Emplacement Boreholes in a Welded Tuff Repository," SAND84-7207, Sandia National Laboratory, New Mexico.

Schuler, R. H. 1987. "Modern Trends in Experimental Radiation Chemistry", in Radiation Research, Proceedings of the 8th International Congress of Radiation Research, Edinburgh, eds. E. M. Fielden, J. F. Fowler, J. H. Hendry and D. Scott, Taylor and Francis, London, pp.42-47.

162

Schwarz, H. A. 1966. "Diffusion Kinetics," Proc. 5th Internat. Conf. Rad. Chem. Water.

Sehested, K. and H. Christensen. 1987. "The Radiation Chemistry of Water and Aqueous Solutions at Elevated Temperatures," in Radiation Research, Proceedings of the 8th International Congress of Radiation Research, Edinburgh, England, Vol. 2, pp. 199-204.

Takagi, J. and K. Ishigure. 1985. "Thermal Decomposition of Hydrogen Peroxide and Its Effect on Reactor Water Monitoring of Boiling Water Reactors," *Nucl. Sci. Eng.*, 89:177-186.

Takagi, J., N. Ichikawa and Y. Hemmi. 1988. "Evaluation of Corrosion Environment in BWR Primary Circuit by Water Radiolysis Model," Proceedings of the 1988 JAIF International Conference on Water Chemistry in Nuclear Power Plants, Tokyo, Japan.

Taylor, K. J., I. D. Bland and G. P. Marsh 1983. "Corrosion Studies on Containment Materials for Vitrified High Level Nuclear Waste," AERE-G2970, United Kingdom Atomic Energy Authority, Harwell, England.

Van Konynenburg, R. A. 1986. "Radiation Chemical Effects in Experiments to Study the Reaction of Glass in an Environment of Gamma-Irradiated Air, Groundwater, and Tuff, UCRL-53719, Lawrence Livermore National Laboratory, Livermore, California.

Van Konynenburg, R. A. 1984,. "Radiation Doses in Granite Around Emplacement Holes in Spent Fuel Test - Climax (Final Report)," UCRL-53580, Lawrence Livermore National Laboratory, Livermore, California.

Walton, J. C. and B. Sagar 1987. "A Corrosion Model for Nuclear Waste Containers," in Scientific Basis For Nuclear Waste Management VIII, Materials Research Society, Boston, Massachusetts.

Worley, B. A. et al. no date, "Deterministic Methods for Sensitivity and Uncertainty Analysis in Large-Scale Computer Models," ORNL, Oak Ridge, Tennessee

Worley, B. A. 1987. ORNL-6428.Oak Ridge, Tennessee

10 CFR Part 60 (Code of Federal Regulations), 1987. Title 10, "Energy," Part 60, "Disposal of High-Level Radioactive Wastes in Geologic Repositories," U. S. Government Printing Office, Washington, D. C.

40 CFR Part 191 (Code of Federal Regulations), 1986. Title 40, "Protection of the Environment," Part 191, "Environmental Standards for the Management and Disposal of Spent Nuclear Fuel, High-Level and Transuranic Radioactive Wastes," U.S. Government Printing Office, Washington, D. C.

10 CFR Part 960 (Code of Federal Regulations), 1987. Title 10, "Energy," Part 960, "General Guidelines for the Recommendation of Sites for Nuclear Waste Repositories," U.S. Government Printing Office, Washington, D. C.

APPENDIX A.

Chemical species reaction data

APPENDIX A

This appendix is a compilation of all reaction rate data. Appendix B contains the appropriate equilibrium data that are used to establish rates involving equilibrium reactions. Some equilibrium rates are listed in this appendix when appropriate forward and reverse reactions have been determined, others are inferred in this thesis from the equilibrium data given in the next appendix. The majority of the rate data were taken from the work of Alberta Ross and co-workers at the Radiation Chemistry Data Center, Notre Dame University. Their efforts have been very valuable to this thesis and the author is deeply appreciative of this.

WATER REACTIONS

```
W 1 e-                >H      OH-                    0.40D+02  0.13D+02
W 2 e-    H+          >H                             0.60D+11  0.13D+02
W 3 e-    OH          >OH-                           0.75D+11  0.13D+02
W 4 e-    H2O2        >OH     OH-                    0.32D+11  0.13D+02
W 5 H     H           >H2                            0.25D+11  0.13D+02
W 6 e-    HO2         >HO2-                          0.50D+11  0.13D+02
W 7 e-    O2          >O2-                           0.47D+11  0.13D+02
W 8 e-    e-          >OH-    OH-     H2             0.12D+11  0.13D+02
W 9 OH    OH          >H2O2                          0.11D+11  0.13D+02
W10 OH-   H           >e-                            0.78D+08  0.19D+02
W11 e-    H           >H2     OH-                    0.62D+11  0.13D+02
W12 e-    HO2-        >OH     OH-     OH-            0.87D+10  0.13D+02
W13 H     OH          >                              0.50D+11  0.13D+02
W14 OH    H2          >H                             0.11D+09  0.13D+02
W14 H                 >H2     OH                     0.49D-01  0.85D+02
W15 H     O2          >HO2                           0.47D+11  0.13D+02
W16 H     HO2         >H2O2                          0.50D+11  0.13D+02
W17 H     O2-         >HO2-                          0.50D+11  0.13D+02
W18 e-    O2-         >HO2-   OH-                    0.51D+11  0.19D+02
W19 H     H2O2        >OH                            0.24D+09  0.14D+02
W20 OH    H2O2        >HO2                           0.41D+08  0.82D+01
W21 OH    HO2         >O2                            0.30D+11  0.13D+02
W22 OH-   H2O2        >HO2-                          0.70D+09  0.19D+02
W22 HO2-             >OH-     H2O2                   0.22D+07  0.19D+02
W24 H+    O2-         >HO2                           0.12D+12  0.13D+02
W24 HO2              >H+      O2-                    0.20D+07  0.13D+02
W25 HO2   O2-         >HO2-   O2                     0.58D+08  0.19D+02
W26 O2-   O2-         >H2O2   O2      OH-    OH-     0.66D+08  0.19D+02
W27 HO2   HO2         >H2O2   O2                     0.11D+08  0.19D+02
W28 H+    OH-         >                              0.14D+12  0.00D+00
W28                  >H+      OH-                    0.78D-01  0.00D+00
W29 OH    O2-         >O2     OH-                    0.30D+11  0.13D+02
AE1 O2               >O2G                            0.10D+07  0.00D+00
AE1 O2G              >O2                             0.42D+03  0.00D+00
```

IRON REACTIONS

```
F  1 FE2+      OH            >FE3+     OH-                        0.57D+09  0.13D+02
F  2 FE2+      e-            >FE+                                 0.55D+11  0.13D+02
F  3 FE+       OH            >FEOH+                               0.70D+09  0.13D+02
F  4 FE+       H2O2          >FE2+     OH-     OH                 0.13D+03. 0.13D+02
F  5 FE+       O2-           >FE2+     OH-     OH-     H2O2       0.88D+09  0.13D+02
F  6 H         FE2+          >FEH2+                               0.34D+07  0.13D+02
F  7 FEH2+     H+            >FE3+     H2                         0.26D+05  0.13D÷02
F  8 FE+       H             >FEII+                               0.30D+07  0.13D+02
F09  FEH+      H+            >FE2+     H2                         0.23D+05  0.13D+02
F10  FE2+      H2O2          >FE3+     OH-     OH                 0.15D+03  0.13D+02
F11  FE2+      O2-           >FE3+     OH-     OH-     H2O2       0.99D+09  0.13D+02
F12  FE3+      e-            >FE2+                                0.50D+11  0.13D+02
F13  FE3+      O2-           >FE2+     O2                         0.99D+09  0.13D+02
F14  FE3+      H             >FE2+     H+                         0.65D+07  0.13D+02
F15  FE3+                    >FEOH2+   H+                         0.75D+08  0.13D+02
F16  FEOH2+    H+            >FE3+                                0.12D+11  0.13D+02
F17  FEOH2+                  >FE(OH)2+H+                          0.15D+06  0.13D+02
F18  FE(OH)2+H+              >FEOH2+                              0.20D+11  0.13D+02
F19  FE(OH)2+FE(OH)2+        >FEOOH    FEOH2+                     0.12D+04  0.13D+02
FQ1  H+        FEOH+         >FE2+                                0.10D+11  0.00D+00
FQ1  FE2+                    >H+       FEOH+                      0.40D+03  0.00D+00
FQ3  FEOH2+    H+            >FE3+                                0.10D+11  0.00D+00
FQ3  FE3+                    >FEOH2+   H+                         0.13D+10  0.00D+00
```

NICKEL REACTIONS

```
NI1 NI2+    e-              >NI+                      0.55D+11  0.13D+02
NI2 NI2+    H               >NI+     H+               0.50D+06  0.13D+02
NI3 NI+     H2O2            >NI2+    OH-      OH       0.99D+08  0.13D+02
NI4 NI+     OH              >NI2+    OH-               0.50D+11  0.13D+02
NI5 NI+     O2              >NI2+    O2-               0.55D+11  0.13D+02
NE1 H+      NIOH+           >NI2+                      0.10D+11  0.00D+00
NE1 NI2+                    >H+      NIOH+             0.14D+03  0.00D+00
NE2 NI2+    OH-     OH-     >NI(OH)2                   0.10D+11  0.00D+00
NE2 NI(OH)2                 >OH-     OH-      NI2+     0.10D-06  0.00D+00
```

CARBONATE REACTIONS

```
CO1 H+      HCO3-      >H2CO3                              0.10D+11  0.00D+00
CO2 H2CO3              >HCO3-   H+                         0.37D+04  0.00D+00
CO3 H+      CO32-      >HCO3-                              0.10D+11  0.00D+00
CO4 HCO3-              >CO32-   H+                         0.71D+00  0.00D+00
CO5 H2CO3             >CO2G                                0.10D+07  0.00D+00
CO6 CO2G              >H2CO3                               0.30D+02  0.00D+00
CO7 HCO3-   O2-        >HO2-    CO3-                        0.37D+07  0.13D+02
CO8 CO3-    O2-        >CO32-   O2                         0.99D+09  0.13D+02
CO9 HCO3-   OH         >CO3-                               0.12D+09  0.13D+02
C10 HCO3-   H          >CO3-    H2                         0.85D+05  0.13D+02
C11 HCO3-   e-         >CO32-   H                          0.15D+07  0.13D+02
C12 CO32-   OH         >CO3-    OH-                        0.94D+09  0.13D+02
C13 CO3-    CO3-       >H2CO3   H2CO3   HO2-    HO2-       0.30D+08  0.13D+02
C14 CO3-    H2O2       >HCO3-   HO2                        0.20D+07  0.13D+02
C15 CO3-    HO2-       >HCO3-   O2-   ,                    0.14D+09  0.13D+02
```

CHLORINE REACTIONS

```
CL1  CL      CL-                >(CL)2-                        0.52D+11  0.13D+02
CL2  (CL)2-                     >CL     CL-                    0.27D+06  0.13D+02
CL3  CL-     OH                 >CLOH-                         0.11D+11, 0.13D+02
CL4  CLOH-                      >CL-    OH                     0.15D+11  0.13D+02
CL5  CLOH-   H+                 >CL                            0.52D+11  0.13D+02
CL6  CL                         >CLOH-  H+                     0.32D+04  0.13D+02
CL7  CL2                        >HOCL   H+      CL-            0.27D+02  0.13D+02
CL8  HOCL    H+      CL-        >CL2                           0.45D+05  0.13D+02
CL9  (CL)2-  H                  >H+     CL-     CL-            0.25D+10  0.13D+02
C10  (CL)2-  (CL)2-             >CL-    CL-     CL2            0.17D+09  0.13D+02
C11  (CL)2-                     >OH     H+      CL-    CL-     0.18D+05  0.13D+02
C12  OH      CL-     CL-        >(CL)2-                        0.50D+04  0.13D+02
C13  (CL)2-  OH                 >HOCL   CL-                    0.25D+10  0.13D+02
```

```
                              SULPHATE REACTIONS
SO1 SO4-2  OH              >SO4-   OH-                    0.40D+07  0.13D+02
SO2 SO4-   HCO3-           >SO4-2  CO3-      H+           0.23D+08  0.13D+02
SO3 SO4-   OH-             >SO4-2  OH                     0.17D+09  0.13D+02
SO4 SO4-                   >SO4-2  OH        H+           0.25D+04  0.13D+02
SO5 SO4-   H2O2            >SO4-2  HO2       H+           0.30D+08  0.13D+02
```

AIR/NITROGEN REACTIONS

A01	O	O		>O2				0.55D+11	0.13D+02
A02	O	HO2		>O2	OH			0.50D+11	0.13D+02
A03	O	OH		>O2	H			0.16D+03	0.70D+02
A04	O			>OH	OH			0.37D+05	0.41D+02
A05	O	OH		>HO2				0.50D+11	0.13D+02
A06	O	H2		>H	OH			0.59D+05	0.35D+02
A07	O	H2O2		>HO2	OH			0.46D+07	0.10D+02
A08	O	H		>OH				0.50D+11	0.13D+02
A09	N	O2		>O	NO			0.56D+05	0.13D+02
A10	NO	H		>NH	O			0.25D+05	0.13D+02
A11	N	H		>NH				0.50D+08	0.13D+02
A12	N	OH		>H	NO			0.80D+11	0.13D+02
A13	N	N		>N2				0.25D+10	0.13D+02
A14	NO	O		>NO2				0.41D+10	0.13D+02
A15	NO	O2		>NO2	O			0.25D+04	0.13D+02
A16	N	H2		>NH2				0.25D+00	0.13D+02
A17	NO	OH		>NO2-	H+			0.22D+11	0.13D+02
A18	NO2	H		>NO	OH			0.19D+12	0.13D+02
A19	NO2	NO2		>H+	H+	NO2-	NO3-	0.25D+08	0.13D+02
A20	NH	OH		>N				0.25D+05	0.13D+02
A21	NH	H2		>NH3				0.12D+02	0.13D+02
A22	NH	O2		>NO	OH			0.25D+08	0.13D+02
A23	NH2	OH		>NH				0.19D+11	0.13D+02
A24	NH2	O2		>NH	HO2			0.25D+05	0.13D+02
A25	NH2	H2		>NH3	H			0.39D+07	0.36D+02
A26	NH3	O		>NH2	OH			0.25D+10	0.13D+02
A27	NH3	OH		>NH2				0.25D+10	0.13D+02
A28	NO2-	H2O2		>NO3-				0.25D+11	0.13D+02
A29	NO2-	H		>NO	OH-			0.40D+11	0.13D+02
A30	NO3-	H		>NO2	OH-			0.12D+08	0.13D+02
A31	NO	NO	O2	>NO2	NO2			0.35D+08	0.13D+02
A32	NH3	H		>NH2	H2			0.17D+11	0.26D+02
A33	NH3			>NH2	H			0.25D+03	0.13D+02
A34	NH2	NH2		>N2H4				0.25D+04	0.13D+02
A35	H	H	N2H4	>N2	N2	H2	H2	0.25D+06	0.13D+02
A36	e-	NO3-		>NO2	OH-	OH-		0.27D+11	0.13D+02
A37	H+	NO2-	e-	>NO	OH-			0.25D+10	0.13D+02

APPENDIX B
COMPUTER CODE LISTINGS

```
         PROGRAM MITIRAD
C
C*******************************************************************
C                         MITIRAD CODE PACKAGE
C*******************************************************************
C        VERSION: MIT 1.0
C        CODE CUSTODIAN: S. A. SIMONSON
C                        MASSACHUSETTS INSTITUTE OF TECHNOLOGY
C                        RM 24-304
C                        77 MASSACHUSETTS AVENUE
C                        CAMBRIDGE, MA 02139
C        VERSION DATE: 7/25/88
C        OPERATING SYSTEM: MICROVMS VERSION 4.4
C        VAX FORTRAN COMPILER VERSION 4.2
C*******************************************************************
C        MITIRAD COMPUTES THE CONCENTRATIONS OF VARIOUS SPECIES
C        PRODUCED BY RADIATION AS A FUNCTION OF TIME USING A
C        VARIATION OF GEAR'S METHOD FOR SOLVING THE STIFF NON-
C        LINEAR SYSTEM OF EQUATIONS.  THE ROUTINES ARE IMPLEMENTED IN
C           DOUBLE PRECISION
C
C  PROGRAM ELEMENTS:
C
C        SOLVER: DRIVER SUBROUTINE FOR THE MITIRAD CALCULATION
C                HANDLES MOST INPUT AND OUTPUT AND CALLS THE
C                APPROPRIATE SUBROUTINES
C
C        FUNS:   FUNCTION EVALUATION SUBROUTINE
C
C        JACS:   JACOBIAN EVALUATION SUBROUTINE
C
C        READIN: READS THE REACTION MATRIX INTO THE VARIOUS
C                INDEX ARRAYS AND ADJUSTS THE REACTION RATE
C                CONSTANTS FOR TEMPERATURE
C
C        LSODE:  LIVERMORE SOLVER OF ORDINARY DIFFERENTIAL
C                EQUATIONS -  A SET OF SUBROUTINES
C                PROVIDED BY ALAN HINDMARSH OF LLNL
C                WHICH SOLVES A GENERAL SET OF ORDINARY
C                DIFFERENTIAL EQUATIONS USING GEARS METHOD FOR
C                STIFF NONLINEAR DIFFERENTIAL EQUATIONS THE CURRENT
C                VERSION MAY HAVE BEEN MODIFIED TO ONLY INCLUDE THE
C                STIFF OPTION TO SAVE ON SPACE
C
C  VAX VMS FILES CALLED:
C
C        SET.FIL:        CONTAINS THE NAMES OF THE INPUT, OUTPUT, AND
C                        PLOT FILES (CHANNEL 5, 6 AND 8, RESPECTIVELY)
C
C  VARIABLES:
C
C        ARS:    ARRAY OF SPECIES NAMES
C        DSRATE: DOSE RATE (RAD/S)
C        DHRATE: DOSE RATE FOR HIGH-LET PARTICLES (RAD/S)
C        EA:     ACTIVATION ENERGY (KJ/MOLE-K)
C        FILNAM: OUTPUT FILE NAME
C        G:      G-VALUE (# SPECIES/100 ev)
C        GH:     G-VALUE FOR HIGH-LET PARTICLES (# SPECIES/100 ev)
C        GAS:    GASEOUS SPECIES ARRAY
C        ID1:    ARRAY SIZING PARAMETER
```

```
C        ID2                "
C        IN1:     INDICIES ARRAYS FOR FUNCTION EVALUATION
C        IN2                "
C        IN3                "
C        INFIL:   INPUT FILE NAME
C        IFLG:    INDEX USED TO SET UP REACTION ORDER
C        ILOGLOG: OUTPUT SPECIFIER 0=LIN,LIN; 1=LOG,LOG; 2=LIN,LOG
C        IND:               "
C        JFLAG:   SPECIFIES NUMBER OF OUTER ITERATIONS
C        KOEF:    MODIFIED REACTION COEFFICIENT MATRIX
C        NEQ:     NUMBER OF SPECIES
C        NJ:      ORIGINAL REACTION COEFFICIENT MATRIX
C        NORSOUT: FLAG FOR RS1 OUTPUT 1=NO , 0 = YES(DEFAULT)
C        NRTN:    NUMBER OF REACTIONS
C        MULTIME: TIME MULTIPLIER FOR NEXT SOLUTION TIME
C        PDJ:     COLUMN VECTOR FOR JACOBIAN MATRIX
C        PLOTFIL: NAME OF THE VMS PLOT FILE TO BE USED FOR RS/1
C        R:       UNIVERSAL GAS CONSTANT (VARIOUS UNITS)
C        RC:      REACTION RATE CONSTANT VECTOR (MOLES/L-S IN GENERAL)
C        TDOSE:   TOTAL DOSE (RADS)
C        TEM:     TEMPERATURE (KELVIN)
C        TEMR:    REFERENCE TEMPERATURE FOR RATE CONSTANTS (KELVIN)
C        TFINAL:  FINAL SOLUTION TIME (SEC)
C        TIM:     TIME VECTOR FOR OUTPUT (SEC)
C        VG:      VOLUME OF GAS PHASE (ML)
C        VL:      VOLUME OF LIQUID PHASE (ML)
C        Y:       SPECIES CONCENTRATION VECTOR (MOLES/L)
C        YSPEC:   CONCENTRATION ARRAY FOR OUTPUT (MOLES/L)
C
C        LSODE VARIABLES ARE DEFINED IN THE LSODE WRITE-UP
C
C FUNCTIONS CALLED:
C
C        DEXP, DFLOTJ, DLOG10
C
C        LSODE FUNCTION CALLS ARE DEFINED IN THE LSODE WRITE-UP
C*************************************************************************
C
         INCLUDE 'PARAMETER.BLK'
         INCLUDE 'DIMENSION.BLK'
         NAMELIST /SIZE/ NEQ,NRTN,MESHPTS,XHIGH,XLOW
         DATA NEQ/1/,NRTN/1/,MESHPTS/0/,XHIGH/0.0/,XLOW/0.0/
         CHARACTER*35 INFIL,FILNAM,PLOTFIL,NLFILE
C
C*************************************************************************
C     OPEN FILE WHICH CONTAINS THE NAMES OF THE INPUT,
C     OUTPUT, AND PLOT FILES FOR THIS RUN
C*************************************************************************
C
       OPEN (5,FILE='SET.FIL',STATUS='OLD',ERR=90)
       READ(5,100)INFIL,FILNAM,PLOTFIL,NLFILE
100    FORMAT(A35)
90     IF(PLOTFIL.EQ.' ') PLOTFIL=' - NONE - '
       CLOSE(5)
       OPEN(5,FILE=INFIL,STATUS='OLD')
       OPEN (6,FILE=FILNAM,STATUS='NEW')
C
C*************************************************************************
C        READ THE NUMBER OF SPECIES(NEQ) AND THE
C        NUMBER OF REACTIONS(NRTN), NUMBER OF MESHPOINTS (MESHPTS)
```

```
C        AND THE RANGE (XHIGH, XLOW)
C*****************************************************************************
C
      READ (5,NML=SIZE)
        IF(MESHPTS.EQ.0) GO TO 115
C
C        CALCULATE PANEL BETWEEN MESHPOINTS __!__XINC__!__
C
        XINC = (XHIGH-XLOW)/DFLOAT(MESHPTS-1)
        XINCSQ = XINC**2.0D0
        NPTS = NEQ*(MESHPTS)
110    FORMAT (37X,I3)
115      CONTINUE
      WRITE (6,120)
120      FORMAT(/
     +,10X,      '                                                      '
     +,/ ,10X,'*              MITIRAD CODE PACKAGE OUTPUT              *'
     +,/ ,10X,'*                 MITIRAD VERSION: MIT  1.0             *'
     +,/,10X,'                                                         ')
      WRITE (6,140)
      WRITE (6,150)
      WRITE (6,140)
      WRITE (6,135) INFIL, FILNAM, PLOTFIL,NLFILE
      WRITE (6,130) NRTN,NEQ, MESHPTS, XLOW, XHIGH
130    FORMAT (//5X,38H  NUMBER OF CHEMICAL REACTIONS        = ,I3,/,
     +          5X,38H  NUMBER OF SPECIES BEING EVALUATED = ,I3,/,
     +          5X,38H  NUMBER OF UNIFORM MESH POINTS       = ,I3,/,
     +          5X,38H  LEFT COORDINATE                     = ,D10.3,/,
     +          5X,38H  RIGHT COORDINATE                    = ,D10.3)
135    FORMAT(  5X,38H  INPUT FILE NAME                     = ,A35,/
     +          5X,38H  OUTPUT FILE NAME                    = ,A35,/
     +          5X,38H  PLOT FILE NAME                      = ,A35,/
     +          5X,38H  NAMELIST FILE                       = ,A35)
140    FORMAT(/80(1H )/)
150    FORMAT(37X,'INPUT')
C
C*****************************************************************************
C        CALL THE ROUTINE THAT READS THE DATA
C        COMPUTES THE SOLUTIONS, AND OUTPUTS
C        THE RESULTS.
C*****************************************************************************
C
        IF(MESHPTS.EQ.0) THEN
        CALL PURE(NEQ,PLOTFIL,NLFILE)
        STOP
        ENDIF
        IF(MESHPTS.GE.5) THEN
                CALL SOLVER (NEQ,NPTS,PLOTFIL,NLFILE)
        ELSE
                PRINT *,'NOT ENOUGH MESH POINTS TO PERFORM CALCULATION'
                PRINT *,'MESHPTS MUST BE => 5'
                PRINT *,'PROGRAM TERMINATED'
        ENDIF
      STOP
      END

      SUBROUTINE SOLVER (NEQ,NPTS,PLOTFIL,NLFILE)
C
C*****************************************************************************
C        VERSION: MIT 1.0
```

```
C         DATE OF LAST CHANGE TO SUBROUTINE: 7/27/88
C         CODE CUSTODIAN: S. A. SIMONSON
C
C         PART OF THE MITIRAD CODE PACKAGE
C         THIS SUBROUTINE IS CALLED BY PROGRAM MITIRAD
C***********************************************************************
C         SOLVER IS THE WORKING ROUTINE FOR THE PROGRAM MITIRAD
C         SOLVER CALL ROUTINES TO READ THE INPUT DATA, SET UP THE NECESSARY
C         WORKING ARRAYS, CALL THE ODE SOLVER AND FINALLY,
C         OUTPUT THE RESULTS TO PRINT AND PLOT FILES.
C***********************************************************************
C  SUBROUTINES CALLED:
C
C         INPUT
C         LSODE
C         INTDY
C***********************************************************************
C
C         INCLUDE 'PARAMETER.BLK'
C
C***********************************************************************
C         DIMENSION BLOCK
C***********************************************************************
C
C         INCLUDE 'DIMENSION.BLK'
          EXTERNAL FLT
          INTEGER FLAG
          DIMENSION RWORK(LRW),IWORK(LIW)
C
C         DIMENSION THE FOLLOWING OPTIONAL PARAMETERS WHEN NEEDED
C
C         DIMENSION gas(id2),TDOSE(ID2)
          DIMENSION YSPEC(ID2,ID2),TIM(ID2), Y(ID2),DKY(ID2)
C         DIMENSION B(ID2,ID2),C(ID2,ID2),D(ID2,ID2)
          REAL*8 MULTIME
          CHARACTER*35 PLOTFIL
          NAMELIST /LSIN/ IOPT,ITASK,RTOL,ITOL,ATOL,RWORK,IWORK
     +    ,RATOL ,IDERV,ISTATE, NPOT, NORSOUT, NPOINTS,MF,NORSDIA
          NAMELIST /STATE/ VL,VG,TEM,TEMR,DSRATE,DHRATE,TOUT,TFINAL
     +                ,TSTEP, MULTIME, IA, VEL
          NAMELIST /NAMES/ ARS
          DATA VL/0.0D0/, VG/0.0D0/, TEM/298./, TEMR/298./, TOUT/0.D0/
          DATA TSTEP/1.D-5/,TFINAL/100./, VEL/0./
          DATA ATOL/0.D0/,MF/222/,ITOL/1/,ISTATE/1/,ITASK/1/,RTOL/1.D-5/
          DATA IDERV/0/,ILOGLOG/1/,NORSOUT/0/,MULTIME/3.16227/
C
C***********************************************************************
C         BEGIN READING THE INPUT DATA LUN=5, NAME LIST FOR STATE VARIABLES
C
C         READ THE LIQUID VOLUME, GAS VOLUME, EVALUATION TEMPERATURE,
C         AND REFERENCE TEMPERATURE FOR RATE CONSTANTS
C***********************************************************************
C
          READ (5,NML=STATE)
100       FORMAT (37X,D18.8)
          WRITE(6,110)VL,VG,TEM,TEMR
110       FORMAT( 5X,38H  LIQUID VOLUME                      = .D14.5,
     +    / 5X,  38H  GAS VOLUME                         = ,D14.5,
     +    / 5X,  38H  TEMPERATURE                        = ,D14.5,
     +    / 5X,  38H  REFERENCE TEMPERATURE              = ,D14.5)
```

```
      WRITE (6,230) DSRATE,DHRATE
230   FORMAT (5X,38H  LOW LET DOSE RATE                 = ,D14.5,
     +/5X,       38H  HIGH LET DOSE RATE                = ,D14.5)
C
C*********************************************************************
C      WRITE TIME CONTROL PARAMETERS
C*********************************************************************
C
      WRITE(6,250) ATOL,RTOL,TFINAL,TSTEP
250   FORMAT(5X,38H  ABSOLUTE TOLERANCE                 = ,D14.5,
     +/5X,       38H  RELATIVE TOLERANCE                 = ,D14.5,
     +/5X,       38H  FINAL TIME EVALUATION              = ,D14.5,
     +/5X,       38H  TIME STEP                          = ,D14.5)
C
C*********************************************************************
C      WRITE TIME STEP INCREMENT
C*********************************************************************
C
      WRITE(6,270)MULTIME
270   FORMAT(5x,38H  TIME MULTIPLE                      = ,D14.5,/)
      WRITE (6,290)
C
C*********************************************************************
C        READ THE SPECIES NAMES
C*********************************************************************
C
      READ (5,NML=NAMES)
130   FORMAT (A1)
140   FORMAT (3X,A8)
C
C*********************************************************************
C      INITIALIZE THE COEFFICIENT AND ORDER MATRICIES FOR THE FUNCTION
C      EVALUATION SEGMENT OF LSODES
C*********************************************************************
C
         DO 160 II=1,NEQ
            DO 160 IJ=1,NRTN
            KOEF(IJ,II)=0
            NJ(IJ,II)=0
160         CONTINUE
C
C*********************************************************************
C      CALL THE INPUT SUBROUTINE TO READ THE EQUATIONS AND SET UP
C      REACTION COEFFICIENTS AND ORDER MATRICIES AND INITIAL
C      CONDITIONS
C*********************************************************************
C
      CALL READIN(NEQ,Y)
      WRITE (6,290)
C
C*********************************************************************
C        SET UP REACTION ORDER INDICIES FOR FAST FUNCTION EVALUATION
C*********************************************************************
C
      DO 165, I = 1,NPTS+1
            Y(NPTS+I)=1.0D0
165   CONTINUE
      DO 180 I=1,NRTN
C
C*********************************************************************
```

```
C            INITIAL ALL REACTANTS TO ZERO ORDER
C*******************************************************************
C
        IN1(I)=NPTS+1
        IN2(I)=NPTS+1
        IN3(I)=NPTS+1
        IND=0
        IFLG=0
          DO 170 J=1,NEQ
C
C*******************************************************************
C            ESTABLISH ALL FIRST ORDER REACTANTS
C*******************************************************************
C
            IF(NJ(I,J).EQ.-1.AND.IND.EQ.0)THEN
            IN1(I)=J
            IFLG=IFLG+1
            ENDIF
            IF(NJ(I,J).EQ.-1.AND.IND.EQ.1)THEN
            IN2(I)=J
            IFLG=IFLG+1
            ENDIF
            IF(NJ(I,J).EQ.-1.AND.IND.EQ.2)THEN
            IN3(I)=J
            IFLG=IFLG+1
            ENDIF
C
C*******************************************************************
C        DETERMINE THE SECOND ORDER REACTANTS (EITHER FIRST TWO
C        OR LAST TWO) IND IS NUMBER OF REACTANTS CHOSEN SO FAR
C*******************************************************************
C
            IF(NJ(I,J).EQ.-2.AND.IND.EQ.0)THEN
            IN1(I)=J
            IN2(I)=J
            IFLG=IFLG+2
            ENDIF
            IF(NJ(I,J).EQ.-2.AND.IND.EQ.1)THEN
            IN2(I)=J
            IN3(I)=J
            IFLG=IFLG+2
            ENDIF
            IND=IND+IFLG
            IFLG=0
170         CONTINUE
180      CONTINUE
C
C*******************************************************************
C        INITIALIZE FOR LSODES
C*******************************************************************
C
        IOPT  = 0
        LSTEP = 0
        IND   = 0
        ITER  = 0
        T     = TOUT
C
C        READ THE NAMELIST LSIN IN FILE NLFILE TO GET PARAMETER CHANGES
C
        OPEN(1,FILE = NLFILE,STATUS = 'OLD')
```

180

```
          READ(1,NML = LSIN)
          CLOSE(1)
C
      WRITE (6,290)
      WRITE (6,291)

280       CONTINUE
C
C         CALL THE ODE SOLVER
C
C         CALL LSODES(FLT,NPTS,Y,T,TOUT,ITOL,RTOL,ATOL,ITASK,ISTATE,IOPT,
C        + RWORK,LRW,IWORK,LIW,JCS,MF)
C
C*******************************************************************************
C         PRINT RESULTS OF PREVIOUS TIME STEP AND INCREMENT
C         THE TIME OR STOP AND PRINT FINAL RESULTS
C*******************************************************************************
C         OPTIONAL ROUTINE TO CALL THE DERIVATIVE EVALUATION SUBROUTINE
C         ARRAY DKY MUST BE DIMENSIONED ABOVE
C*******************************************************************************
C
          IF(IDERV.GE.1) THEN
          LYH = IWORK(21)
          CALL INTDY (T,IDERV,RWORK(LYH),NEQ,DKY,FLAG)
          ENDIF
C
C*******************************************************************************
C         OUTPUT FOR DERIVATIVE EVALUATION (OPTIONAL)
C*******************************************************************************
C
          IF (IDERV.GE.1) THEN
          WRITE (6,350) IDERV,FLAG
 350      FORMAT (/15X,'TIME DERIVATIVES OF ORDER',I2,10X,7H FLAG =,I2/)
          WRITE (6,320) (ARS(I),DKY(I),I=1,NEQ)
          ENDIF
C
C*******************************************************************************
C         CALCULATE THE TOTAL DOSE (TDOSE), AND PRINT THE RESULTS OF
C         THE LAST ITERATION
C*******************************************************************************
C
          TDOSEL = DSRATE*T
          TDOSEH = DHRATE*T
      WRITE (6,290)
 291      FORMAT(37X,'OUTPUT')
 290      FORMAT (/80(1h_),/)
          WRITE (6,310) T
          DO 295 J = 1,MESHPTS,3
          K = NEQ*(J-1)
          WRITE (6,320) J, (ARS(I),Y(K+I),I=1,NEQ)
 295      CONTINUE
          WRITE (6,300) IWORK(11),TDOSEL,TDOSEH
 300      FORMAT (4X,13H NO. STEPS = ,I5,/5X,
         + 'LOW DOSE (RAD) =',D10.3,3X,'HIGH DOSE (RAD) =',D10.3)
 310      FORMAT (25X,'CONCENTRATIONS AT TIME = ',D11.4,/)
 320      FORMAT (5X,6HNODE =,I3,2(5X,A8,' = ',D15.6,' **'))
C
C*******************************************************************************
C optional:
C         THIS BEGINS A SEGMENT TO EVALUATE GAS PHASE SPECIES IN TERMS
```

```
C         OF ATMOSPHERES (SPECIES WITH "G" IN THERE NAME ARE ASSUMED TO
C         BE GASEOUS, ARRAY GAS MUST BE DIMENSIONED ABOVE
C****************************************************************************
C
C         IF (VG.LE.0.) GOTO 340
C            DO 340 I=1,NEQ
C            IC=0
C            R=0.08206D0
C            IC=INDEX(ARS(I),'G')
C            IF(IC.NE.0)THEN
C            GAS(ITER)=Y(I)*(VL/VG)*TEM*R
C            WRITE(6,330)ARS(I),GAS(ITER)
C330         FORMAT(/5X,A8,' = ',D11.3,' ATM')
C            ENDIF
C340         CONTINUE
C
C****************************************************************************
C         STORE RESULTS FOR OUTPUT TO PLOT FILE
C****************************************************************************
C
      IF(ISTATE.EQ.-1) THEN
      ISTATE=2
      GO TO 280
      ENDIF
C
C****************************************************************************
C         SAVE THE POINTS IN LOG-LOG (ILOGLOG=1, SEMI-LOG (ILOGLOG>1)
C         OR NORMAL (ILOGLOG=0) FORMAT FOR SPLINE EVALUATION
C****************************************************************************
C
         IF(ITER.EQ.0) GO TO 361
         IF(ILOGLOG.EQ.1) THEN
         TIM(ITER) = dlog10(T)
         ELSE
         TIM(ITER) = T
         ENDIF
         DO 360 J = 1,MESHPTS
         K = NEQ*(J-1)
         DO 360 I=1,NEQ
         IF(ILOGLOG.GE.1)THEN
          IF(Y(I).LE.0.d0) then
          yspec(iter,i) = 0.d0
          print *,' time = ',t,' yspec',i,' = zero'
          go to 360
          endif
          YSPEC(ITER,K+I) = dlog10(Y(K+I))
         ELSE
         YSPEC(ITER,K+I) = Y(K+I)
         ENDIF
360      CONTINUE
         IOPT = 0
361   continue
C
C****************************************************************************
C         DETERMINE ERROR CONDITION AND TAKE APPROPRIATE ACTION
C****************************************************************************
C
      IF (ISTATE.LT.0) GO TO 380
      IF (T.GE.TFINAL) GO TO 380
C
```

```
C           TSTEP<0.0        ADDITIVE        ISFLAG=1
C           TSTEP>= 0.0      MULTIPLICATIVE  ISFLAG = 0
C
      IF (ITER.EQ.0)THEN
        IF(TSTEP.LT.0.D0)THEN
         ISFLAG=1
         TSTEP=-TSTEP
         TOUT = 0.D0
        ELSE
         TOUT = TSTEP
         ISFLAG = 0
        ENDIF
        ENDIF
C
C*********************************************************************
C       INCREMENT THE TIME WITH ADDITIVE TERM OR MULTIPLICATIVE TERM
C*********************************************************************
C
        IF(ISFLAG.EQ.1) THEN
                TOUT = TOUT + MULTIME*TSTEP
        ELSE
                TOUT=TOUT*MULTIME
        ENDIF
370     ITER = ITER+1
C
C       DON'T OVERSHOOT FINAL TIME REQUESTED
C
        IF(TOUT.GT.TFINAL)TOUT=TFINAL
C
C       CONTINUE INTEGRATION
C
        GO TO 280
C
C*********************************************************************
C       PRINT THE FINAL RESULTS
C*********************************************************************
C
380     LENRW = IWORK(17)
        LENIW = IWORK(18)
        NST = IWORK(11)
        NFE = IWORK(12)
        NJE = IWORK(13)
        WRITE (6,290)
        WRITE (6,381)
381     FORMAT(29X,'RUN STATISTICS')
        WRITE (6,290)
        WRITE (6,390) LENRW,LENIW,NST,NFE,NJE
390     FORMAT(5X,/25H REQUIRED RWORK SIZE   = ,I5,
       +      5X,/25H IWORK SIZE            = ,I5,
       +      5X,/25H NUMBER OF STEPS       = ,I5,
       +      5X,/25H # OF FUNC.- EVALS.    = ,I5,
       +      5X,/25H # OF JACOB.- EVALS    = ,I5)
        WRITE (6,400) ISTATE
400     FORMAT (///22H ERROR HALT...ISTATE =,I3)
        CLOSE(6)
C
C*********************************************************************
C       WRITE TO DATA FILE TO BE READ BY RS/1
C*********************************************************************
C
```

```
        IF (NORSOUT.EQ.1) GO TO 470
        OPEN (8,FILE=PLOTFIL,STATUS='NEW')
        WRITE (8,410) NEQ, ITER, MESHPTS, ILOGLOG
410     FORMAT (1X,I3)
           DO 420 I1 = 1,NEQ
           WRITE (8,430) ARS(I1)
420        CONTINUE
430     FORMAT (1X,A8)
           DO 450 I2 = 1,ITER
           WRITE (8,460) TIM(I2)
              DO 440 I3 = 1,MESHPTS
                 K = NEQ*(I3-1)
                 DO 440 I4 = 1,NEQ
              WRITE (8,460) YSPEC(I2,K+I4)
440        CONTINUE
450        CONTINUE
460     FORMAT (1X,E21.14)
470     CLOSE (8)
        RETURN
        END

        SUBROUTINE PURE (NEQ,PLOTFIL,NLFILE)
C
C*********************************************************************
C       VERSION: MIT 1.0
C       DATE OF LAST CHANGE TO SUBROUTINE: 8/2/88
C       CODE CUSTODIAN: S. A. SIMONSON
C
C       PART OF THE MITIRAD CODE PACKAGE
C       THIS SUBROUTINE IS CALLED BY PROGRAM MITIRAD
C*********************************************************************
C       PURE IS THE WORKING ROUTINE FOR THE PROGRAM MITIRAD FOR PURE
C       REACTION PROBLEMS.  PURE WILL READ THE INPUT DATA, SET UP THE
C       NECESSARY WORKING ARRAYS, CALL THE ODE SOLVER AND FINALLY,
C       OUTPUT THE RESULTS TO PRINT AND PLOT FILES.
C*********************************************************************
C  SUBROUTINES CALLED:
C
C       INPUT
C       LSODES
C       INTDY
C*********************************************************************
C
        INCLUDE 'PARAMETER.BLK'
C
C*********************************************************************
C       DIMENSION BLOCK
C*********************************************************************
C
        INCLUDE 'DIMENSION.BLK'
        EXTERNAL FRO,JACL
        INTEGER FLAG
        DIMENSION RWORK(LRW),IWORK(LIW)
C
C       DIMENSION THE FOLLOWING OPTIONAL PARAMETERS WHEN NEEDED
C
C       DIMENSION GAS(ID2)
        DIMENSION YSPEC(ID2,ID2),TIM(ID2), Y(ID2),DKY(ID2)
C       DIMENSION B(ID2,ID2),C(ID2,ID2),D(ID2,ID2)
        REAL*8 MULTIME
```

```
      CHARACTER*35 PLOTFIL,NLFILE
      NAMELIST /LSIN/ IOPT,ITASK,RTOL,ITOL,ATOL,RWORK,IWORK
     +    ,RATOL ,IDERV,ISTATE, NPOT, NORSOUT, NPOINTS,MF,NORSDIA
      NAMELIST /STATE/ VL,VG,TEM,TEMR,DSRATE,DHRATE,TOUT,TFINAL
     +              ,TSTEP, MULTIME,IA, VEL
      NAMELIST /NAMES/ ARS
      DATA VL/0.0D0/, VG/0.0D0/, TEM/298./, TEMR/298./, TOUT/0.D0/
      DATA TSTEP/1.D-5/,TFINAL/100./, VEL/0./
      DATA ATOL/1.D-15/,MF/21/,ITOL/1/,ISTATE/1/,ITASK/1/,RTOL/1.D-5/
      DATA IDERV/0/,ILOGLOG/1/,NORSOUT/0/,MULTIME/3.16227/
C
C*******************************************************************
C     BEGIN READING THE INPUT DATA LUN=5, NAMELIST FOR STATE VARIABLES
C
C     READ THE LIQUID VOLUME, GAS VOLUME, EVALUATION TEMPERATURE,
C     AND REFERENCE TEMPERATURE FOR RATE CONSTANTS
C*******************************************************************
C
      READ (5,NML=STATE)
100   FORMAT (37X,D18.8)
      WRITE(6,110)VL,VG,TEM,TEMR
110   FORMAT( 5X,38H   LIQUID VOLUME                        = ,D14.5,
     +  /  5X,  38H   GAS VOLUME                           = ,D14.5,
     +  /  5X,  38H   TEMPERATURE                          = ,D14.5,
     +  /  5X,  38H   REFERENCE TEMPERATURE                = ,D14.5)
      WRITE (6,230) DSRATE,DHRATE
230   FORMAT (5X,38H   LOW LET DOSE RATE                    = ,D14.5,
     +/5X,        38H   HIGH LET DOSE RATE                   = ,D14.5)
C
C*******************************************************************
C     WRITE TIME CONTROL PARAMETERS
C*******************************************************************
C
      WRITE(6,250) ATOL,RTOL,TFINAL,TSTEP
250   FORMAT(5X,38H   ABSOLUTE TOLERANCE                   = ,D14.5,
     +/5X,        38H   RELATIVE TOLERANCE                   = ,D14.5,
     +/5X,        38H   FINAL TIME EVALUATION                = ,D14.5,
     +/5X,        38H   TIME STEP                            = ,D14.5)
C
C*******************************************************************
C     WRITE TIME STEP INCREMENT
C*******************************************************************
C
      WRITE(6,270)MULTIME
270   FORMAT(5x,38H   TIME MULTIPLE                        = ,D14.5,/)
      WRITE (6,290)
C
C*******************************************************************
C     READ THE SPECIES NAMES
C*******************************************************************
C
      READ (5,NML=NAMES)
130   FORMAT (A1)
140   FORMAT (3X,A8)
C
C*******************************************************************
C     INITIALIZE THE COEFFICIENT AND ORDER MATRICIES FOR THE FUNCTION
C     EVALUATION SEGMENT OF LSODES
C*******************************************************************
C
```

```
        DO 160 II=1,NEQ
          DO 160 IJ=1,NRTN
          KOEF(IJ,II)=0
          NJ(IJ,II)=0
160       CONTINUE
C
C*****************************************************************
C       CALL THE INPUT SUBROUTINE TO READ THE EQUATIONS AND SET UP
C       REACTION COEFFICIENTS AND ORDER MATRICIES
C*****************************************************************
C
      CALL READIN(NEQ,Y)
        WRITE (6,290)
C
C*****************************************************************
C       SET UP REACTION ORDER INDICIES FOR FAST FUNCTION EVALUATION
C*****************************************************************
C
      Y(NEQ+1)=1.0D0
        DO 180 I=1,NRTN
C
C*****************************************************************
C       INITIAL ALL REACTANTS TO ZERO ORDER
C*****************************************************************
C
        IN1(I)=NEQ+1
        IN2(I)=NEQ+1
        IN3(I)=NEQ+1
        IND=0
        IFLG=0
          DO 170 J=1,NEQ
C
C*****************************************************************
C       ESTABLISH ALL FIRST ORDER REACTANTS
C*****************************************************************
C
          IF(NJ(I,J).EQ.-1.AND.IND.EQ.0)THEN
          IN1(I)=J
          IFLG=IFLG+1
          ENDIF
          IF(NJ(I,J).EQ.-1.AND.IND.EQ.1)THEN
          IN2(I)=J
          IFLG=IFLG+1
          ENDIF
          IF(NJ(I,J).EQ.-1.AND.IND.EQ.2)THEN
          IN3(I)=J
          IFLG=IFLG+1
          ENDIF
C
C*****************************************************************
C       DETERMINE THE SECOND ORDER REACTANTS (EITHER FIRST TWO
C       OR LAST TWO) IND IS NUMBER OF REACTANTS CHOSEN SO FAR
C*****************************************************************
C
          IF(NJ(I,J).EQ.-2.AND.IND.EQ.0)THEN
          IN1(I)=J
          IN2(I)=J
          IFLG=IFLG+2
          ENDIF
          IF(NJ(I,J).EQ.-2.AND.IND.EQ.1)THEN
```

```
              IN2(I)=J
              IN3(I)=J
              IFLG=IFLG+2
              ENDIF
              IND=IND+IFLG
              IFLG=0
170           CONTINUE
180       CONTINUE
C
C***********************************************************************
C         INITIALIZE FOR LSODES
C***********************************************************************
C
      IOPT  = 0
      LSTEP = 0
      IND   = 0
      ITER  = 0
      T     = TOUT
         OPEN(1,FILE = NLFILE,STATUS = 'UNKNOWN')
         READ(1,NML = LSIN)
         CLOSE(1)
C
C***********************************************************************
C         CALL THE ODE SOLVER.  LSODES WILL CALL
C         THE ROUTINES FRO AND JAC INTERNALLY.
C         OTHER INPUT AND OUTPUT PARAMETERS ARE
C         DEFINED IN THE LSODES WRITE-UP.
C***********************************************************************
C
      WRITE (6,290)
      WRITE (6,291)

280       CONTINUE
C
      CALL LSODE(FRO,NEQ,Y,T,TOUT,ITOL,RTOL,ATOL,ITASK,ISTATE,IOPT,
     + RWORK,LRW,IWORK,LIW,JACL,MF)
C
C***********************************************************************
C         PRINT RESULTS OF PREVIOUS TIME STEP AND INCREMENT
C         THE TIME OR STOP AND PRINT FINAL RESULTS
C***********************************************************************
C         OPTIONAL ROUTINE TO CALL THE DERIVATIVE EVALUATION SUBROUTINE
C***********************************************************************
C
      IF(IDERV.GE.1) THEN
      LYH = IWORK(21)
      CALL INTDY (T,IDERV,RWORK(LYH),NEQ,DKY,FLAG)
      ENDIF
C
C***********************************************************************
C         OUTPUT FOR DERIVATIVE EVALUATION (OPTIONAL)
C***********************************************************************
C
      IF(IDERV.GE.1) THEN
      WRITE (6,350) IDERV,FLAG
350   FORMAT (/15X,'TIME DERIVATIVES OF ORDER',I2,10X,7H FLAG =,I2/)
      WRITE (6,320) (ARS(I),DKY(I),I=1,NEQ)
      ENDIF
C
C***********************************************************************
```

```
C           CALCULATE THE TOTAL DOSE (TDOSE), AND PRINT THE RESULTS OF
C           THE LAST ITERATION
C**********************************************************************
C
            TDOSEL=DSRATE*T
            TDOSEH=DHRATE*T
            WRITE (6,290)
291         FORMAT(37X,'OUTPUT')
290         FORMAT (/80(1h_),/)
            WRITE (6,310) T
            WRITE (6,320) (ARS(I),Y(I),I=1,NEQ)
            WRITE (6,300) IWORK(11),TDOSEL, TDOSEH
300         FORMAT (13H NO. STEPS = ,I5,/5X,'LOW DOSE RATE (RAD) = ',D10.3
     +           /5X,'HIGH DOSE RATE (RAD) = ',D10.3)
310         FORMAT (25X,'CONCENTRATIONS AT TIME = ',D11.4,/)
320         FORMAT (2(5X,A8,' = ',D15.6,' **'))
C
C**********************************************************************
C optional:
C       THIS BEGINS A SEGMENT TO EVALUATE GAS PHASE SPECIES IN TERMS
C       OF ATMOSPHERES (SPECIES WITH "G" IN THERE NAME ARE ASSUMED TO
C       BE GASEOUS
C**********************************************************************
C
            IF (VG.LE.0.) GOTO 340
            DO 340 I=1,NEQ
                IC=0
                R=0.08206D0
                IC=INDEX(ARS(I),'G')
            IF(IC.NE.0)THEN
                GAS=Y(I)*(VL/VG)*TEM*R
                WRITE(6,330)ARS(I),GAS
330             FORMAT(/5X,A8,' = ',D11.3,' ATM')
            ENDIF
340         CONTINUE
C
C**********************************************************************
C       STORE RESULTS FOR OUTPUT TO PLOT FILE
C**********************************************************************
C
            IF(ISTATE.EQ.-1) THEN
            ISTATE=2
            GO TO 280
            ENDIF
C
C**********************************************************************
C       SAVE THE INITIAL CONDITIONS ON THE FIRST CALL, OTHERWISE
C       SAVE THE POINTS IN LOG-LOG OR SEMI-LOG FORMAT FOR SPLINE
C       EVALUATION
C**********************************************************************
C
            IF(ITER.EQ.0) GO TO 361
            IF(ILOGLOG.EQ.1) THEN
            TIM(ITER) = dlog10(T)
            ELSE
            TIM(ITER) = T
            ENDIF
            DO 360 I=1,NEQ
            IF(ILOGLOG.GE.1)THEN
              if(Y(I).LE.0.d0) then
```

```
                yspec(iter,i) = 0.d0
                print *,' time = ',t,' yspec',i,' = zero'
                go to 360
                endif
                YSPEC(ITER,I) = dlog10(Y(I))
              ELSE
                YSPEC(ITER,I) = Y(I)
              ENDIF
360       CONTINUE
          IOPT = 0
361     continue
C
C*****************************************************************************
C       DETERMINE ERROR CONDITION AND TAKE APPROPRIATE ACTION
C*****************************************************************************
C
        IF (ISTATE.LT.0) GO TO 380
        IF (T.GE.TFINAL) GO TO 380
C
C       TSTEP < 0.0       ADDITIVE              ISFLAG = 1
C       TSTEP >= 0.0      MULTIPLICATIVE        ISFLAG = 0
C
        IF(ITER.EQ.0) THEN
        IF(TSTEP.LT.0.D0) THEN
                ISFLAG = 1
                TSTEP = -TSTEP
                TOUT = 0.D0
        ELSE
                TOUT = TSTEP
                ISFLAG = 0
        ENDIF
        ENDIF
C
C*****************************************************************************
C       INCREMENT THE TIME AND CONTINUE
C*****************************************************************************
C
        IF(ISFLAG.EQ.1) THEN
                TOUT = TOUT + MULTIME*TSTEP
        ELSE
                TOUT = TOUT*MULTIME
        ENDIF
370     ITER = ITER+1
        IF(TOUT.GT.TFINAL)TOUT=TFINAL
        GO TO 280
C
C*****************************************************************************
C       PRINT THE FINAL RESULTS
C*****************************************************************************
C
380     CONTINUE
          LENRW = IWORK(17)
        LENIW = IWORK(18)
        NST = IWORK(11)
        NFE = IWORK(12)
        NJE = IWORK(13)
          WRITE (6,290)
          WRITE (6,381)
381       FORMAT(29X,'RUN STATISTICS')
          WRITE (6,290)
```

```
      WRITE (6,390) LENRW,LENIW,NST,NFE,NJE
390   FORMAT(5X,/25H REQUIRED RWORK SIZE    = ,I5,
     +       5X,/25H IWORK SIZE             = ,I5,
     +       5X,/25H NUMBER OF STEPS        = ,I5,
     +       5X,/25H # OF FUNC.- EVALS.     = ,I5,
     +       5X,/25H # OF JACOB.- EVALS     = ,I5)
      WRITE (6,400) ISTATE
400   FORMAT (///22H ERROR HALT...ISTATE =,I3)
      CLOSE(6)
C
C*****************************************************************
C       WRITE TO DATA FILE TO BE READ BY RS/1
C*****************************************************************
C
      IF (NORSOUT.EQ.1) GO TO 470
      OPEN (8,FILE=PLOTFIL,STATUS='NEW')
      WRITE (8,410) NEQ, ITER, MESHPTS, ILOGLOG
410   FORMAT (1X,I3)
         DO 420 I1 = 1,NEQ
         WRITE (8,430) ARS(I1)
420      CONTINUE
430   FORMAT (1X,A8)
         DO 450 I2 = 1,ITER
         WRITE (8,460) TIM(I2)
            DO 440 I3 = 1,NEQ
            WRITE (8,460) YSPEC(I2,I3)
440         CONTINUE
450      CONTINUE
460   FORMAT (1X,E21.14)
470   CLOSE (8)
      RETURN
      END


      SUBROUTINE INPUT(NEQ,Y)
C
C*****************************************************************
C       VERSION: MIT 2.0
C       CODE CUSTODIAN: S. A. SIMONSON
C       DATE OF LAST CHANGE OF SUBROUTINE: 11/20/87
C       PART OF THE DYNEQL CODE PACKAGE
C       THIS SUBROUTINE IS CALLED BY THE SUBROUTINE SOLVER
C*****************************************************************
C       INPUT READS LOGICAL UNIT NUMBER 5 FOR THE REACTION MATRIX
C       AND REACTION RATE CONSTANTS.  REACTION RATE CONSTANTS ARE
C       ADJUSTED FOR TEMPERATURE USING AN ARRHENIUS TEMPERATURE
C       DEPENDENCE
C*****************************************************************
C
      INCLUDE 'PARAMETER.BLK'
      INCLUDE 'DIMENSION.BLK'
      CHARACTER*3 CJP
      DIMENSION Y(ID2), IR(ID1,3), IP(ID1,4), EA(ID2)
      ARS(0)='        '
      R=8.314D-3
      RU=0.08206D0
C
C*****************************************************************
C       READ THE REACTIONS ONE BY ONE AND SET UP THE COEFFICIENT
C       MATRICIES (KOEF), AND THE REACTION ORDER MATRIX (NJ)
C*****************************************************************
```

```
C
        IFLAG=0
        WRITE (6,10)
10      FORMAT (//10X,
       +59HCHEMICAL REACTIONS, RATE CONSTANTS, AND ACTIVATION ENERGIES//
       +26X,'REACTIONS'28X,'RATE',3X,'ACTIVATION',/62X,'CONSTANT',2X,
       +'ENERGIES')
            DO 140 I=1,NRTN
            READ(5,100)CJP,(IR(I,K),K=1,3),(IP(I,K),K=1,4),RC(I),EA(I)
C
C********************************************************************
C       IF THE FLAG INDICATES (IFLAG=1) THAT A PHASE EQUILIBRIUM
C       REACTION WAS DETECTED FOR THE LAST REACTION
C       (RC(I - 1) = -1), CALCULATE THE REACTION RATE USING A
C       HENRY'S LAW APPROACH FOR REACTION (I) GIVEN THAT
C       RC(I - 1) = 100.
C********************************************************************
C
        IF(IFLAG.EQ.1) THEN
        RC(I)=100.D0*VL/(VL+VG*(RC(I)/(RU*TEM)))
        IFLAG=0
        ELSE
C
C********************************************************************
C       CONVERT THE RATE CONSTANT USING AN ARRHENIUS EXPERSSION
C********************************************************************
C
        RC(I)=(RC(I)/DEXP(-EA(I)/(R*TEMR)))*DEXP(-EA(I)/(R*TEM))
        ENDIF
C
C********************************************************************
C       SET THE FIRST RATE CONSTANT OF A PHASE PARTITIONING TO 100.
C       (THIS SHOULD BE A LIQUID TO GAS PHASE REACTION FOLLOWED BY
C       A GAS TO LIQUID PHASE SO THAT THE PROGRAM PROPERLY ACCOUNTS
C       FOR THE PARTITIONING)
C********************************************************************
C
        IF(RC(I).LT.0) THEN
        RC(I)=100.D0
        IFLAG=1
        ENDIF
        WRITE(6,110)CJP,(ARS(JIABS(IR(I,K))),K=1,3),
       +    (ARS(IP(I,K)),K=1,4),RC(I),EA(I)
100     FORMAT(A3,3X,7I3,D18.8/D18.8)
110     FORMAT(A3,1X,3A8,'>',4A8,D9.2,1X,D9.2)
C
C********************************************************************
C       SET UP THE REACTION INDICIES IR AND THE PRODUCT INDICIES IP
C       FILL THE COEFFICIENT MATRIX KOEF
C
C       CHECK FIRST TO SEE IF ANY OF THE REACTANTS ARE PRESENT
C       IN A SECOND ORDER FASHION
C********************************************************************
C
        IF(((IR(I,1).EQ.IR(I,2)).OR.(IR(I,2).EQ.IR(I,3)))
       +    .AND.(IR(I,2).NE.0))THEN
        NJ(I,JIABS(IR(I,2)))=-2
        KOEF(I,JIABS(IR(I,2)))=-2
        ENDIF
C
```

```
C*******************************************************************
C       FILL UP THE MATRIX FOR THE REACTANTS WHICH ARE
C       FIRST ORDER
C*******************************************************************
C
        DO 120 K=1,3
        IF((IR(I,K).NE.0).AND.(NJ(I,JIABS(IR(I,K))).NE.-2))THEN
        NJ(I,JIABS(IR(I,K)))=-1
        KOEF(I,JIABS(IR(I,K)))=-1
        ENDIF
120     CONTINUE
C
C*******************************************************************
C       CHECK FOR SECOND ORDER PRODUCTS
C*******************************************************************
C
        IF(((IP(I,1).EQ.IP(I,2)).OR.(IP(I,2).EQ.IP(I,3)))
     +  .AND.(IP(I,2).NE.0))THEN
        KOEF(I,JIABS(IP(I,2)))=2
        ENDIF
        IF(((IP(I,2).EQ.IP(I,3)).OR.(IP(I,3).EQ.IP(I,4)))
     +  .AND.(IP(I,3).NE.0))THEN
        KOEF(I,JIABS(IP(I,3)))=2
        ENDIF
C
C*******************************************************************
C       FILL UP THE PRODUCTS MATRIX FOR FIRST ORDER PRODUCTS
C*******************************************************************
C
        DO 130 K=1,4
        IF((IP(I,K).NE.0).AND.(KOEF(I,IP(I,K)).NE.2))THEN
        KOEF(I,IP(I,K))=1
        ENDIF
130     CONTINUE
140     CONTINUE
C
C*******************************************************************
C       NORMALIZE CATALYTIC REACTANTS
C*******************************************************************
C
        DO 150 K=1,NEQ
        DO 150 I=1,NRTN
C
C*******************************************************************
C       ARE THERE PRODUCTS OF SPECIES K AS WELL AS REACTANTS OF
C       SPECIES K?
C*******************************************************************
C
        IF((KOEF(I,K).NE.NJ(I,K)).AND.(NJ(I,K).NE.0))
     +  KOEF(I,K)=KOEF(I,K)+NJ(I,K)
C
C*******************************************************************
C      ARE THERE ONLY PRODUCTS? (FILL NJ AFTER CHECKING FOR
C      CATALYTIC REACTIONS)
C*******************************************************************
C
        IF((KOEF(I,K).NE.NJ(I,K)).AND.(NJ(I,K).EQ.0))
     +  NJ(I,K)=KOEF(I,K)
150     CONTINUE
C
```

```
C***********************************************************************
C        READ INITIAL VALUES, G-VALUES, RELATIVE TOLERENCES, AND
C        CONVERT G-VALUES FROM # spec/100 ev  TO  moles/l-rad
C***********************************************************************
C
         READ (5,160) DUM
         READ (5,160) DUM
160      FORMAT (A1)
         WRITE (6,190)
190      FORMAT(//12X,7HLOW LET,4X,8HHIGH LET,3X,5HLOWER,3X,
        + /    5HUPPER,3X,9HDIFFUSION,
        + /    9X,8HG-VALUES,3X,8HG-VALUES,3X,5HBOUND,10X,
        +     5HBOUND,4X,11HCOEFFICIENTS)
         DO 200 I=1,NEQ
         READ (5,220) G(I),GH(I),BCL(I),BCR(I), DIF(I)
         WRITE (6,210) ARS(I),G(I),GH(I),BCL(I),BCR(I),DIF(I)
         GH(I)= GH(I)*1.033D-9
200      G(I) = G(I)*1.033D-9
210      FORMAT (/1X,A8,5(2X,D9.2))
220      FORMAT(5x,D10.3,4(/5x,D10.3))
         CALL YINITIAL(Y)
         CLOSE (5)
         RETURN
         END



         SUBROUTINE FRO (NEQ,T,Y,YDOT)
C
C***********************************************************************
C        VERSION: MIT 2.0
C        CODE CUSTODIAN: S. A. SIMONSON
C        DATE OF LAST CHANGE OF SUBROUTINE: 11/20/87
C        THIS SUBROUTINE IS PART OF THE DYNEQL CODE PACKAGE
C        FRO IS CALLED BY THE SUBROUTINE LSODE.
C***********************************************************************
C        FRO CALCULATES THE FUNCTION:
C                              SUM
C        DY/DT = G*DSRATE + _____  (RC(J)*KOEF(J)*REACTANTS(J))
C                           J REACTIONS
C***********************************************************************
C
         INCLUDE 'PARAMETER.BLK'
         INCLUDE 'DIMENSION.BLK'
         DIMENSION Y(ID2),YDOT(ID2)
C
C***********************************************************************
C        OUTER LOOP ITERATES THROUGH ALL OF THE ODES, AND THE INNER
C        LOOP ITERATES OVER THE APPLICABLE REACTIONS FOR EACH ODE.
C***********************************************************************
C
         DO 110 I=1,NEQ
         YDOT(I) = 0.0D0
           DO 100 J=1,NRTN
           IF (KOEF(J,I).EQ.0) GO TO 100
           YDOT(I) =YDOT(I)+RC(J)*DFLOTJ(KOEF(J,I))*Y(IN1(J))*Y(IN2(J))
        +    *Y(IN3(J))
100        CONTINUE
         YDOT(I) = G(I)*DSRATE+GH(I)*DHRATE+YDOT(I)
110      CONTINUE
         RETURN
```

```
          END

          SUBROUTINE JACL (NEQ,T,Y,ML,MU,PD,NROWPD)
C
C*****************************************************************************
C         VERSION: MIT 2.0
C         CODE CUSTODIAN: S. A. SIMONSON
C         DATE OF LAST CHANGE TO SUBROUTINE: 11/20/87
C         JACL IS PART OF THE DYNEQL CODE PACKAGE
C         JACL IS CALLED BY THE SUBROUTINE LSODE
C*****************************************************************************
C         JACL CALCULATES THE FULL JACOBIAN MATRIX
C*****************************************************************************
C
          INCLUDE 'PARAMETER.BLK'
          INCLUDE 'DIMENSION.BLK'
          DIMENSION PD(NROWPD,ID2), Y(ID2)
          DO 100 J=1,NEQ
          DO 100 I=1,NEQ
            DO 100 K=1,NRTN
            IF(KOEF(K,J).EQ.0.OR.KOEF(K,I).EQ.0) GOTO 100
            A = RC(K)*DFLOTJ(KOEF(K,I)*JIABS(KOEF(K,J)))
            IM = IN1(K)
            IN = IN2(K)
            IO = IN3(K)
C
C*****************************************************************************
C     CATCH SECOND ORDER REACTIONS
C*****************************************************************************
C
          IF((IM.EQ.IN.OR.IO.EQ.IN).AND.(IN.EQ.J))THEN
          PD(I,J) = PD(I,J)+A*Y(IM)*Y(IO)
          GO TO 100
          ENDIF
          IF (IM.EQ.J) THEN
          PD(I,J) = PD(I,J)+A*Y(IN)*Y(IO)
          GOTO 100
          ENDIF
C
C*****************************************************************************
C     FIRST ORDER REACTIONS
C*****************************************************************************
C
          IF (IO.EQ.J) THEN
          PD(I,J) = PD(I,J)+A*Y(IN)*Y(IM)
          GOTO 100
          ENDIF
          IF (IN.EQ.J) PD(I,J) = PD(I,J)+A*Y(IM)*Y(IO)
100       CONTINUE
          RETURN
          END
```

```
C
          DIMENSION RC(ID1), G(ID2), GH(ID2), DIF(ID2), BCL(ID2)
          DIMENSION BCR(ID2)
          dimension in1(id1),in2(id1),in3(id1),koef(id1,id2),
     +          nj(id1,id2)
          character*8 ars(id2)
C
```

```
c
      parameter(id1=200,id2=210,lrw=3000,liw=200)
      implicit double precision (a-h,o-z)
      COMMON /SINTGR/ NRTN, IN1, IN2, IN3,KOEF, NJ,MESHPTS
      COMMON /SCHAR/ ARS
      COMMON /SREAL/ RC, DSRATE, G, TEM, TEMR, VL, VG, GH, DHRATE
     + ,DIF,BCR,BCL,XINCSQ,XINC,VEL
c
```

```
c
        parameter(id1=200,id2=210,lrw=2000,liw=150)
        implicit double precision (a-h,o-z)
        COMMON /SINTGR/ NRTN, IN1, IN2, IN3,KOEF, NJ, MESHPTS
        COMMON /SCHAR/ ARS
        COMMON /SREAL/ RC, DSRATE, G, TEM, TEMR, VL, VG, GH, DHRATE
       +          , DIF,BCR,BCL,XINCSQ,XINC,VEL
c
```

```
C
        FUNCTION BC (Y,YDOT,T,I,M,BCN)
C
C*************************************************************************
C       VERSION: MIT 1.0
C       CODE CUSTODIAN: S. A. SIMONSON
C       DATE OF LAST CHANGE OF SUBROUTINE: 7/25/88
C       THIS SUBROUTINE IS PART OF THE SPATIAL_RADIOLYSIS CODE PACKAGE
C       BC IS CALLED BY THE SUBROUTINE FUNS.
C*************************************************************************
C
C       BOUNDARY CONDITION FUNCTION
C
C       BCN DETERMINES THE BOUNDARY (BCN<0. IS RIGHT, BCN>0. IS LEFT)
C       AND MAY UTILIZE PRESELECTED BOUNDARY CONDITIONS
C       |BCN|= 2.0      CONSTANT CONCENTRATION
C       BCN GREATER THAN OR EQUAL TO ZERO IS LEFT FLUX BOUNDARY
C       BCN = -1.0 IS ZERO FLUX ACROSS RIGHT BOUNDARY
C
C       THE BOUNDARY CONDITIONS CAN BE MODIFIED AS NEEDED
C
        INCLUDE 'PARAMETER.BLK'
        INCLUDE 'DIMENSION.BLK'
        DIMENSION Y(ID2),YDOT(ID2)
C
C       CHECK IF BCN IS RIGHT BOUNDARY CONDITION
C
        IF(BCN.EQ.-1.0.OR.BCN.EQ.-2.0) GO TO 100
C
C       BCN = 2.0 IS CONSTANT CONCENTRATION DC/DT=0.0 AT LEFT BOUNDARY
C
        IF(BCN.EQ.2.0)THEN
        BC = 0.0D0
        GO TO 110
        ENDIF
C
C       BCN = 0.0 IS NO FLUX AT LEFT BOUNDARY
C
        BC = YDOT(I) +
     +              2.0D0*DIF(I)*(Y(I+NEQ) - Y(I))
     +              + BCN*DIF(I)*XINC/2.0D0
        GO TO 110

100     CONTINUE
C
C       BCN = -1.0 IS NO FLUX AT RIGHT BOUNDARY
C
        IF(BCN.EQ.-1.0) BC = YDOT(I+M) +
     +              2.D0*DIF(I)*(Y(I+M-NEQ) - Y(I+M))
C
C       BCN = -2.0 IS CONSTANT CONCENTRATION DC/DT=0.0 AT RIGHT BOUNDARY
C
        IF(BCN.EQ.-2.0) BC = 0.0D0
110     CONTINUE
        RETURN
        END
```

```
C
C*****************************************************************************
c
c    generate the spline fits for the adjoint calculation
c
C*****************************************************************************
C
C        call splinem(neq,iter,yspec,tim,b,c,d)
C
C*****************************************************************************
C
C        PRINT THE RESULTS FOR RS/1
C
C*****************************************************************************
C
C        OPEN (8,FILE='spline.file',STATUS='NEW')
C        do 901 j = 1,neq
C          write(8,902)j,(b(i,j),c(i,j),d(i,j),i=1,iter)
C901      continue
C902      format(1x,i3/,3(3x,e18.8))
C        close (8)
C
C*****************************************************************************
c
c        SETUP FOR THE ADJOINT EVALUATION ROUTINE
C
C*****************************************************************************

         subroutine splinem (neq,iter,yspec,tim,b,c,d)
C
C*****************************************************************************
C
C        VERSION: MIT 2.0
C        CODE CUSTODIAN: S. A. SIMONSON
C        DATE OF LAST CHANGE OF SUBROUTINE: 10/23/87
C        PART OF THE DYNEQL CODE PACKAGE
C        THIS SUBROUTINE IS CALLED BY THE SUBROUTINE SOLVER
C
C*****************************************************************************
C
c    interfaces dyneql with spline, an error stop occurs
c    if any concentrations go to zero anywhere except for the initial
c    conditions
C
C*****************************************************************************
c
         include 'parameter.blk'
         dimension yspec(id2,id2),tim(id2)
         dimension b(id2,id2),c(id2,id2),d(id2,id2)
         dimension temb(id2),temc(id2),temd(id2), temy(id2)
         do 100 i = 1, neq
            do 10 j = 1, iter
              if(yspec(j,i).eq.-100.) then
          print *,' time = ',tim(j),' yspec',i,' = zero'
        print *,'yspec = ',yspec(j,i)
              print *,' Zero or Negative value detected in concentrations'
              print *,' Execution Terminated'
              stop
                 endif
              temy(j) = yspec(j,i)
```

```
10        continue
          call spline(iter,tim,temy,temb,temc,temd)
          do 20 j = 1, iter
              b(j,i) = temb(j)
              c(j,i) = temc(j)
              d(j,i) = temd(j)
20        continue
100   continue
      return
      end
```

```
        SUBROUTINE JCS (NPTS,T,Y,J,IAN,JAN,PDJ)
C
C***********************************************************************
C       VERSION: MIT 1.0
C       CODE CUSTODIAN: S. A. SIMONSON
C       DATE OF LAST CHANGE TO SUBROUTINE: 1/9/87
C       JCS IS PART OF THE SPATIAL RADIOLYSIS CODE PACKAGE
C       JCS IS CALLED BY THE SUBROUTINE LSODES
C***********************************************************************
C       JCS CALCULATES THE COLUMN VECTORS FOR THE JACOBIAN MATRIX
C       (d/dY(I))(dY(J)/dt)
C***********************************************************************
C
        INCLUDE 'PARAMETER.BLK'
        INCLUDE 'DIMENSION.BLK'
        DIMENSION PDJ(ID2), Y(ID2)
C
C***********************************************************************
C       CALCULATE THE SPATIAL (DIFFUSIONAL) PART OF THE JACOBIAN
C***********************************************************************
C
        ICHC = NPTS-NEQ
        KMAG = J/NEQ
        KMAG = NEQ*KMAG
        KMO  = JMOD(J-1,NEQ) + 1
C
C       DO BOUNDARY NODES
C
        IF(J.LE.NEQ) THEN
                CALL JBC(Y,PDJ,J,KMO,BCL(KMO))
                GO TO 10
                ENDIF
        IF(J.GT.ICHC) THEN
                CALL JBC(Y,PDJ,J,KMO,BCR(KMO))
                GO TO 10
                ENDIF
C
C       DO INNER NODES
C
        PDJ(J-NEQ)= DIF(I)
        PDJ(J+NEQ)= DIF(I)
        PDJ(J) = - 2*DIF(I)
10      CONTINUE
C
C***********************************************************************
C       CALCULATE THE CHEMICAL REACTION PORTION
C***********************************************************************
C
        DO 100 I=1,NEQ
          DO 100 K=1,NRTN
            IF(KOEF(K,KMO).EQ.0.OR.KOEF(K,I).EQ.0) GOTO 100
            A = RC(K)*DFLOTJ(KOEF(K,I)*JIABS(KOEF(K,KMO)))
            IM  = IN1(K)
            IMR = IM + KMAG
            IN  = IN2(K)
            INR = IN + KMAG
            IO  = IN3(K)
            IOR = IU + KMAG
```

```
C
C***********************************************************************
C      CATCH SECOND ORDER REACTIONS
C***********************************************************************
C
       IF((IM.EQ.IN.OR.IO.EQ.IN).AND.(IN.EQ.KMO))THEN
       PDJ(I) = PDJ(I)+A*Y(IMR)*Y(IOR)
       GO TO 100
       ENDIF
       IF (IM.EQ.KMO) THEN
       PDJ(I) = PDJ(I)+A*Y(INR)*Y(ICR)
       GOTO 100
       ENDIF
C
C***********************************************************************
C      FIRST ORDER REACTIONS
C***********************************************************************
C
       IF (IO.EQ.KMO) THEN
       PDJ(I) = PDJ(I)+A*Y(INR)*Y(IMR)
       GOTO 100
       ENDIF
       IF (IN.EQ.KMO) PDJ(I) = PDJ(I)+A*Y(IMR)*Y(IOR)
100    CONTINUE
       RETURN
       END
```

```
          SUBROUTINE FLT (NPTS,T,Y,YDOT)
C
C*****************************************************************************
C         VERSION: MIT 1.0
C         CODE CUSTODIAN: S. A. SIMONSON
C         DATE OF LAST CHANGE OF SUBROUTINE: 7/25/88
C         THIS SUBROUTINE IS PART OF THE SPATIAL RADIOLYSIS CODE PACKAGE
C         FLT IS CALLED BY THE SUBROUTINE LSODE OR LSODES.
C*****************************************************************************
C         FLT CALCULATES THE FUNCTION:
C
C
C DY(X,Y(J),T)/DT =
C
C (RADIATION)        G(LOW LET)*LOW LET DOSE RATE +
C                                G(HIGH LET)*HIGH LET DOSE RATE
C
C
C                           SUM
C (CHEMICAL        +    _____   (RC(J)*KOEF(J)*REACTANTS(J))
C REACTION)             J REACTIONS
C
C
C
C (DIFFUSION)      +   D(I)*(Y(X-1) - 2*Y(X) + Y(X+1))*(1/DX**2)
C
C (CONVECTION)     +   U*(Y(X)-Y(X-1))/DX
C
C         WITH BOUNDARY CONDITIONS SET BY FUNCTION BC
C*****************************************************************************
C
          INCLUDE 'PARAMETER.BLK'
          INCLUDE 'DIMENSION.BLK'
          DIMENSION Y(ID2),YDOT(ID2)

C
C*****************************************************************************
C         LOOP 110 CALCULATES INNER MESH POINTS, AND REACTIONS
C         FOR THE BOUNDARY NODES
C*****************************************************************************
C
          DO 110 M=1,MESHPTS
          K=NEQ*(M-1)
C
C*****************************************************************************
C         OUTER LOOP ITERATES THROUGH ALL OF THE SPECIES, AND THE INNER
C         LOOP ITERATES OVER THE APPLICABLE REACTIONS FOR EACH ODE.
C*****************************************************************************
C
          DO 110 I=1,NEQ
          YDOT(K+I) = 0.0D0
            DO 100 J=1,NRTN
            IF (KOEF(J,I).EQ.0) GO TO 100
            YDOT(K+I) = YDOT(K+I) + RC(J)*DFLOTJ(KOEF(J,I))*Y(K+IN1(J))
        +              *Y(K+IN2(J))*Y(K+IN3(J))
100       CONTINUE
          YDOT(K+I) = YDOT(K+I) + G(I)*DSRATE + GH(I)*DHRATE
C
C         TRANSPORT FOR BOUNDARIES IS ACCOUNTED FOR AT THE END
C
          IF(M.EQ.1.OR.M.EQ.MESHPTS) GO TO 110
C
```

```
C          COMMENT OUT DIFFUSION OR CONVECTION WHEN NOT USED
C
           YDOT(K+I) =  YDOT(K+I)
      +               + DIF(I)*(Y(I+K-NEQ) - 2.D0*Y(I+K) + Y(I+K+NEQ))
C     +               + VEL*(Y(I+K-NEQ) - Y(I+K))*XINC
110        CONTINUE
C
C***********************************************************************
C          BOUNDARY CONDITIONS AT FIRST MESH POINT, 0.0, CONSTANT,
C          OR DEFINE IN EXTERNAL FUNCTION BC, SKIP IF A PURE
C          REACTION PROBLEM IS BEING RUN (use MITIRAD WHEN a pure
C          reaction PROBLEM is being run)
C***********************************************************************
C
      DO 10 I=1,NEQ
         YDOT(I)=BC(Y,YDOT,T,I,0,BCL(I))
10         CONTINUE
C
C***********************************************************************
C          BOUNDARY CONDITIONS AT LAST MESH POINT, 0.0, CONSTANT,
C          OR DEFINE EXTERNAL FUNCTION FOR BCHIGH(I), SKIP FOR PURE
C          REACTION PROBLEM (use radiol for pure reaction)
C***********************************************************************
C
      M = NEQ*(MESHPTS - 1)
      DO 120 I=1,NEQ
      YDOT(M+I) = BC (Y,YDOT,T,I,M,BCR(I))
120        CONTINUE
           RETURN
           END
```

```
      subroutine spline (n,x,y,b,c,d)
      integer n
      double precision x(n),y(n),b(n),c(n),d(n), t
      integer nml,ib,i
c
c
c       the coefficients b(i), c(i), and d(i), i = 1,2,...,n are
c       computed for a cubic interpolating spline
c
c       s(x) = y(i) +b(i)*(x-x(i))+c(i)*(x-x(i))**2 + d(i)*(x-x(i))**3
c
c       for x(i).le. x .le. x(i+1)
c
c       input
c
c       n = the number of data points or knots (n.ge.2)
c       x = the abscissas of the knots in strictly increasing order
c       y = the ordinates of the knots
c
c       output
c
c       b, c, d = arrays of spline coefficients as defined above
c
c       using p to denote differentiation
c
c       y(i) = s(x(i))
c       b(i) = sp(x(i))
c       c(i) = spp(x(i))/2
c       d(i) = sppp(x(i))/6 (derivative from right)
c
c  the accompaning function subprogram seval can be used
c  to evaluate the spline
c
c
      nml=n-1
      if(n.lt.2) return
      if(n.lt.3) go to 50
c
c set up tridiagonal system
c
c b = diagonal, d = offdiagonal, c = right hand side
c
      d(1) = x(2)-x(1)
      c(2) = (y(2)-y(1))/d(1)
      do 10 i = 2, nml
            d(i) = x(i+1) - x(i)
            b(i) = 2.d0*(d(i-1)+d(i))
            c(i+1) = (y(i+1)-y(i))/d(i)
            c(i) = c(i+1) - c(i)
 10      continue
c
c  end conditions. third derivatives at x(1) and x(n)
c  obtained from divided differences
c
      b(1) = -d(1)
      b(n) = -d(n-1)
      c(1) = 0.d0
      c(n) = 0.d0
      if(n.eq.3) go to 15
      c(1) = c(3)/(x(4)-x(2))-c(2)/(x(3)-x(1))
      c(n) = c(n-1)/(x(n)-x(n-2)) - c(n-2)/(x(n-1) - x(n-3))
```

```fortran
        c(1) = c(1)*d(1)**2/(x(4)-x(1))
        c(n) = -c(n)*d(n-1)**2/(x(n)-x(n-3))
c
c  forward elimination
c
15      do 20 i = 2, n
            t = d(i-1)/b(i-1)
            b(i) = b(i) - t*d(i-1)
            c(i) = c(i)-t*c(i-1)
20      continue
c
c  back substitution
c
        c(n) = c(n)/b(n)
        do 30 ib = 1, nml
            i = n-ib
            c(i) = (c(i)-d(i)*c(i+1))/b(i)
30      continue
c
c  c(i) is now the sigma(i) of the text
c
c  compute polynomial coefficients
c
        b(n) = (y(n)-Y(nml))/d(nml) + d(nml)*(c(nml)+2.d0*c(n))
        do 40 i = 1, nml
            b(i) = (y(i+1)-y(i))/d(i) - d(i)*(c(i+1)+2.d0*c(i))
            d(i) = (c(i+1) - c(i))/d(i)
            c(i) = 3.d0*c(i)
40      continue
        c(n) = 3.d0*c(n)
        d(n) = d(n-1)
        return

50      b(1) = (y(2)-y(1))/(x(2)-x(1))
        c(1) = 0.d0
        d(1) = 0.d0
        b(2) = b(1)
        c(2) = 0.d0
        d(2) = 0.d0
        return
        end


        double precision function seval(n,u,x,y,b,c,d)
        integer n, i,j,k
        real*8 u,x(n),y(n),b(n),c(n),d(n), dx
c
c       this subroutine evaluates the cubic spline function
c
c  seval = y(i) + b(i)*(u-x(i)) + c(i)*(u-x(i))**2 + d(i)*(u-x(i))**3
c
c  where x(i).lt. u .lt. x(i+1), using horner's rule
c
c  if u .lt. x(1) then i = 1 is used
c  if u.ge. x(n) then i = n is used
c
c  input
c
c       n = the number of data points
c       u = the abscissa at which the spline is to be evaluated
```

```
c          x, y - the arrays of data abscissas and ordinates
c          b,c,d - arrays of spline coefficients computed by spline
c
c  if u is not i·. the same interval as the previous call, then a
c  binary searri is performed to determine the proper interval.
c
          data i/1/
C         COMMENTED OUT FOLLOWING STATEMENTS AND REQUIRE SEARCH ON
C         EACH CALL
C         if(i.ge.n) i - 1
C         if(u.lt.x(i)) go to 10
C         if(u.le.x(i+1)) go to 30
c
c  binary search
c
  10      i - 1
          j - n+1
  20      k - (i+j)/2
          if(u.lt.x(k)) j - k
          if(u.ge.x(k)) i - k
          if(j.gt.i+1) go to 20
c
c  evaluate spline
c
  30      dx - u - x(i)
          seval - y(i) + dx*(b(i) + dx*(c(i) + dx*d(i)))
          return
          end
```

```
        SUBROUTINE READIN(NEQ,Y)
C
C*********************************************************************
C       VERSION: MIT 1.0
C       CODE CUSTODIAN: S. A. SIMONSON
C       DATE OF LAST CHANGE OF SUBROUTINE: 1/9/87
C       PART OF THE SPATIAL_RADIOLYSIS CODE PACKAGE
C       THIS SUBROUTINE IS CALLED BY THE SUBROUTINE DRIVE
C*********************************************************************
C       INPUT READS LOGICAL UNIT NUMBER 5 FOR THE REACTION MATRIX
C       AND REACTION RATE CONSTANTS.  REACTION RATE CONSTANTS ARE
C       ADJUSTED FOR TEMPERATURE USING AN ARRHENIUS TEMPERATURE
C       DEPENDENCE
C*********************************************************************
C
        INCLUDE 'PARAMETER.BLK'
         INCLUDE 'DIMENSION.BLK'
        CHARACTER*3 CJP
        DIMENSION  EA(ID2), Y(ID2), IP(ID1,4), IR(ID1,3)
        DATA R/8.314D-3/, RU/0.08206D0/, IFLAG/0/
        ARS(0)='        '
C
C*********************************************************************
C       READ THE REACTIONS ONE BY ONE AND SET UP THE COEFFICIENT
C       MATRICIES (KOEF), AND THE REACTION ORDER MATRIX (NJ)
C*********************************************************************
C
        WRITE (6,10)
10      FORMAT (//10X,
       +59HCHEMICAL REACTIONS, RATE CONSTANTS, AND ACTIVATION ENERGIES//
       +26X,'REACTIONS'28X,'RATE',3X,'ACTIVATION',/62X,'CONSTANT',2X,
       +'ENERGIES')
        IFLAG=0
         DO 140 I=1,NRTN
         READ(5,100)CJP,(IR(I,K),K=1,3),(IP(I,K),K=1,4),RC(I),EA(I)
C
C*********************************************************************
C       IF THE FLAG INDICATES (IFLAG=1) THAT A PHASE EQUILIBRIUM
C       REACTION WAS DETECTED FOR THE LAST REACTION
C       (RC(I - 1) = -1), CALCULATE THE REACTION RATE USING A
C       HENRY'S LAW APPROACH FOR REACTION (I) GIVEN THAT
C       RC(I - 1) = 100.
C*********************************************************************
C
        PRINT *,I,RC(I)
        IF(IFLAG.EQ.0.AND.RC(I).GE.0.0) THEN
        RC(I)=(RC(I)/DEXP(-EA(I)/(R*TEMR)))*DEXP(-EA(I)/(R*TEM))
        ENDIF
        IF(IFLAG.GE.1) THEN
        IF(IFLAG.EQ.1) RC(I)=1.D6*VL/(VL+VG*(RC(I)/(RU*TEM)))
        IF(IFLAG.EQ.2) RC(I) = 10.D0**(10.0D0-RC(I))
         IFLAG=0
        ENDIF
C
C*********************************************************************
C       CONVERT THE RATE CONSTANT USING AN ARRHENIUS EXPERSSION
C*********************************************************************
C
C
```

```
C******************************************************************************
C       SET THE FIRST RATE CONSTANT OF A PHASE PARTITIONING TO 100.
C       (THIS SHOULD BE A LIQUID TO GAS PHASE REACTION FOLLOWED BY
C       A GAS TC LIQUID PHASE SO THAT THE PROGRAM PROPERLY ACCOUNTS
C       FOR THE PARTITIONING)
C******************************************************************************
C
              IF(RC(I).EQ.-1.0) THEN
              RC(I)=1.D+6
              IFLAG=1
              ENDIF
                    IF(RC(I).EQ.-2.0) THEN
                    RC(I) = 1.0D10
                    IFLAG = 2
                    ENDIF
              WRITE(6,110)CJP,(ARS(JIABS(IR(I,K))),K=1,3),
           +  (ARS(IP(I,K)),K=1,4),RC(I),EA(I)
100           FORMAT(A3,3X,7I3,D18.8/D18.8)
110           FORMAT(1X,A3,1X,3A8,'>',4A8,D9.2,1X,D9.2)
C
C******************************************************************************
C       SET UP THE REACTION INDICIES IR AND THE PRODUCT INDICIES IP
C       FILL THE COEFFICIENT MATRIX KOEF
C
C       CHECK FIRST TO SEE IF ANY OF THE REACTANTS ARE PRESENT
C       IN A SECOND ORDER FASHION
C******************************************************************************
C
              IF(((IR(I,1).EQ.IR(I,2)).OR.(IR(I,2).EQ.IR(I,3)))
           +  .AND.(IR(I,2).NE.0))THEN
              NJ(I,JIABS(IR(I,2)))=-2
              KOEF(I,JIABS(IR(I,2)))=-2
              ENDIF
C
C******************************************************************************
C       FILL UP THE MATRIX FOR THE REACTANTS WHICH ARE FIRST ORDER
C******************************************************************************
C
              DO 120 K=1,3
              IF((IR(I,K).NE.0).AND.(NJ(I,JIABS(IR(I,K))).NE.-2))THEN
              NJ(I,JIABS(IR(I,K)))=-1
              KOEF(I,JIABS(IR(I,K)))=-1
              ENDIF
120           CONTINUE
C
C******************************************************************************
C       CHECK FOR SECOND ORDER PRODUCTS
C******************************************************************************
C
              IF(((IP(I,1).EQ.IP(I,2)).OR.(IP(I,2).EQ.IP(I,3)))
           +  .AND.(IP(I,2).NE.0))THEN
              KOEF(I,JIABS(IP(I,2)))=2
              ENDIF
              IF(((IP(I,2).EQ.IP(I,3)).OR.(IP(I,3).EQ.IP(I,4)))
           +  .AND.(IP(I,3).NE.0))THEN
              KOEF(I,JIABS(IP(I,3)))=2
              ENDIF
C
C******************************************************************************
C       FILL UP THE PRODUCTS MATRIX FOR FIRST ORDER PRODUCTS
```

```
C***********************************************************************
C
            DO 130 K=1,4
            IF((IP(I,K).NE.0).AND.(KOEF(I,IP(I,K)).NE.2))THEN
            KOEF(I,IP(I,K))=1
            ENDIF
130         CONTINUE
140      CONTINUE
C
C***********************************************************************
C      NORMALIZE CATALYTIC REACTANTS
C***********************************************************************
C
         DO 150 K=1,NEQ
            DO 150 I=1,NRTN
C
C***********************************************************************
C      ARE THERE PRODUCTS OF SPECIES K AS WELL AS REACTANTS OF
C      SPECIES K?
C***********************************************************************
C
            IF((KOEF(I,K).NE.NJ(I,K)).AND.(NJ(I,K).NE.0))
     +      KOEF(I,K)=KOEF(I,K)+NJ(I,K)
C
C***********************************************************************
C      ARE THERE ONLY PRODUCTS? (FILL NJ AFTER CHECKING FOR
C      CATALYTIC REACTIONS)
C***********************************************************************
C
            IF((KOEF(I,K).NE.NJ(I,K)).AND.(NJ(I,K).EQ.0))
     +      NJ(I,K)=KOEF(I,K)
150         CONTINUE
C
C***********************************************************************
C      READ INITIAL VALUES, G-VALUES, RELATIVE TOLERENCES, AND
C      CONVERT G-VALUES FROM # spec/100 ev  TO  moles/l-rad
C***********************************************************************
C
      READ (5,160) DUM
      READ (5,160) DUM
160   FORMAT (A1)
      WRITE (6,190)
190   FORMAT(//18X,7HLOW LET,4X,8HHIGH LET,3X,5HLOWER,3X,
     +          5HUPPER,3X,9HDIFFUSION,
     + /        14X,3X,8HG-VALUES,3X,8HG-VALUES,3X,5HBOUND,3X,
     +          5HBOUND,2X,11HCOEFFICIENT)
         DO 200 I=1,NEQ
         READ (5,220)  G(I), GH(I), BCL(I), BCR(I), DIF(I)
         WRITE (6,210) ARS(I), G(I), GH(I), BCL(I), BCR(I),
     +                 DIF(I)
         IF(MESHPTS.GT.0) DIF(I) = DIF(I)/XINCSQ
         GH(I)= GH(I)*1.033D-9
200      G(I) = G(I)*1.033D-9
210   FORMAT (/1X,A8,5(2X,D9.2))
220   FORMAT(8X,D10.3,4(/8X,D10.3))
         CALL YINITIAL(Y)
      DO 240 K=1,NEQ
      WRITE(6,230) ARS(K), Y(K)
230   FORMAT(1X,A8,5(2X,D9.2)/11X)
240      CONTINUE
```

```
CLOSE (5)
RETURN
END
```

```
        PROGRAM FILEMAKER
C
C*********************************************************************
C       VERSION: MIT 1.0
C       CODE CUSTODIAN: S. A. SIMONSON
C       DATE OF LAST CHANGE OF SUBROUTINE: 1/9/87
C       PART OF THE SPATIAL_RADIOLYSIS CODE PACKAGE
C       THIS PROGRAM IS CALLED PRIOR TO RUNNING THE CODE
C*********************************************************************
C       FILEMAKER GENERATES THE NECESSARY INPUT FILES TO RUN MITIRAD
C       REACTION RATE CONSTANTS ARE
C       ADJUSTED FOR TEMPERATURE USING AN ARRHENIUS TEMPERATURE
C       DEPENDENCE, UNLESS OTHER DATA ARE INPUT
C*********************************************************************
C
        INCLUDE 'PARAMETER.BLK'
        INCLUDE 'DIMENSION.BLK'
        DIMENSION GN(ID2),GHN(ID2),DIFN(ID2),IBFLAG(ID2)
        DIMENSION IAFLAG(ID2)
        DIMENSION  EA(ID2), YI(ID2), IP(ID1,4), IR(ID1,3)
        DIMENSION ISETS(30)
        REAL*8 MULTIME
        CHARACTER*3 CJP(ID1)
        CHARACTER*8 ARSN(ID2)
        CHARACTER*40 TNAME,DFILNAME(30),NLNAME
        NAMELIST /NAMESN/ ARSN,GN,GHN,DIFN
        NAMELIST /NAMES/ ARS
        NAMELIST /LSIN/ IOPT, ITASK, RTOL, ITOL, ATOL, RWORK
     +                  ,IDERV, ISTATE, NPOT, NORSOUT, NPOINTS
     +                  ,MF, NORSDIA, RATOL
        NAMELIST /STATE/ VL, VG, TEM, TEMR, DSRATE, DHRATE,
     +                  TOUT, TFINAL, TSTEP, MULTIME, IA, VEL
        NAMELIST /SIZE/ NEQ, NRTN, MESHPTS,XHIGH, XLOW
        NAMELIST /VALUES/ YI
C*********************************************************************
C       INITIAL ALL DEFAULT VALUES
C*********************************************************************
        DATA DFILNAME/'WATER.298','FE.298','CU.298','NI.298',
     +   'CL.298','BR.298','SO4.298','AIR.298','CO2.298',
     +   'FEEQ.298','FEEQ.363','NIEQ.298','NIEQ.363','CUEQ.298',
     +   'CUEQ.363','AIREQ.298','H2OEQ.298','H2OEQ.363',
     +   'CESIUM.298','DIF.298'/

        DATA R/8.314D-3/, RU/0.08206D0/, IFLAG/0/, ARS/ID2*'        '/

        DATA TEM/298./,TEMR/298./,TOUT/0.0/,TFINAL/1000./,DHRATE/0.0/

        DATA DSRATE/0.0/,TSTEP/1.D-3/,MULTIME/3.16227/,VL/0.0/,VG/0.0/

        DATA MF/121/,ITOL/1/,ISTATE/1/,ITASK/1/,IDERV/0/,NORSOUT/0/

        DATA NORSDIA/1/,ILOGLOG/1/,RTOL/1.D-5/,ATOL/1.D-16/,IA/1/

        DATA MESHPTS/0/,XLOW/0.0/,XHIGH/1.0/, VEL/0.0D0/

        DATA YI/ID2*0.0/, NPOINTS/100/, RATOL/1.D-4/, NPOT/1/

C*********************************************************************
C       START PROCESSING
C*********************************************************************
```

```
        IAFLAG(0) = 0
        NRTN = 0
        NRTNN = 0
        NEQ = 0
        NEQN = 0
        PRINT *,' '
        PRINT *,'        FILEMAKER GENERATES THE NECESSARY INPUT FILES'
        PRINT *,'        FOR RUNNING MITIRAD.  DEFAULT VALUES ARE PUT'
        PRINT *,'        INTO THE FILE "SET.FIL" FOR THE OUTPUT FILES'
        PRINT *,'        THIS FILE SHOULD BE EDITED TO MAKE THESE'
        PRINT *,'        OUTPUT FILE NAMES UNIQUE'
        PRINT *,' '
        PRINT *,'        IT IS NOT NECESSARY TO RUN FILEMAKER, THE'
        PRINT *,'        NECESSARY FILES CAN BE GENERATED USING THE'
        PRINT *,'        EDITOR AND THE INSTRUCTIONS IN THE MITIRAD'
        PRINT *,'        USERS MANUAL'
        PRINT *,' '
C****************************************************************************
C       WRITE FIRST FILE, 'SET.FIL'
C****************************************************************************
        OPEN(8,FILE='SET.FIL',STATUS='UNKNOWN')
        PRINT *,' '
        PRINT *,' INPUT A FILE NAME FOR THE REACTION INPUT FILE, E.G.
     +  TEST.DATA'
        PRINT *,' '
        ACCEPT 2,TNAME
        PRINT *,' '
        PRINT *,' FILE NAME FOR  NUMERICAL PARAMETERS?'
        PRINT *,' '
        ACCEPT 2,NLNAME
        PRINT *,' '
        WRITE (8,2) TNAME,'MITIRAD.OUTPUT','MITIRAD.PLOT',NLNAME,
     +              'MITIAD.OUTPUT','MITIAD.PLOT','MITIAD.SPLINE',
     +              'LINTY.OUTPUT','DIAGNOSTICS.FILE'
        CLOSE(8)
C****************************************************************************
C       BEGIN PROCESSING NON-DEFAULT VALUES
C****************************************************************************
        PRINT *,'        CHANGE THE DEFAULT # OF MESHPOINTS? [0]'
        PRINT *,'        ( 1 = YES, 0 = NO) '
        PRINT *,' '
        ACCEPT 22,IL
        IF (IL.NE.1)
     +                                          GO TO 71
        PRINT *,' '
        PRINT *,'        HOW MANY MESH POINTS WILL THERE BE? '
        PRINT *,' '
        ACCEPT *,MESHPTS
        IF(MESHPTS.NE.0) THEN
        PRINT *,' '
        PRINT *,'        WHAT IS THE LOWER BOUND: [0.0]'
        PRINT *,' '
        ACCEPT *,XLOW
        PRINT *,' '
        PRINT *,'        WHAT IS THE UPPER BOUND: [1.0]'
        PRINT *,' '
        ACCEPT *,XHIGH
        PRINT *,' '
        ENDIF
71      PRINT *,'        CHANGE THE DEFAULT GAS VOLUME? [0.0]'
```

```
      PRINT *,'         ( 1 - YES, 0 - NO) '
      PRINT *,' '
      ACCEPT 22,IL
      IF (IL.NE.1)
    +                                              GO TO 72
      PRINT *,' '
      PRINT *,'         WHAT IS THE VOLUME OF GAS, ml: '
      ACCEPT *,VG
72    PRINT *,'         CHANGE THE DEFAULT LIQUID VOLUME? [0.0]'
      PRINT *,'         ( 1 - YES, 0 - NO) '
      PRINT *,' '
      ACCEPT 22,IL
      IF (IL.NE.1)
    +         GO TO 73
      PRINT *,' '
      PRINT *,'         WHAT IS THE LIQUID VOLUME, ml:
    +         [0.0 IF GAS IS 0.0]'
      PRINT *,' '
      ACCEPT *,VL
73    PRINT *,' '
      PRINT *,'         CHANGE THE DEFAULT TEMPERATURE? [298.0]'
      PRINT *,'         ( 1 - YES, 0 - NO) '
      PRINT *,' '
      ACCEPT 22,IL
      PRINT *,' '
      IF (IL.NE.1)
    +         GO TO 74
      PRINT *,'         WHAT IS THE TEMPERATURE, K: '
      PRINT *,' '
      ACCEPT *,TEM
74    PRINT *,' '
      PRINT *,'         CHANGE THE FINAL TIME? [1000.0]'
      PRINT *,'         ( 1 - YES, 0 - NO) '
      PRINT *,' '
      ACCEPT 22,IL
      PRINT *,' '
      IF (IL.NE.1)
    +         GO TO 75
      PRINT *,'         WHAT IS THE FINAL TIME, S:'
      PRINT *,' '
      ACCEPT *,TFINAL
75    PRINT *,' '
      PRINT *,'         CHANGE THE TIME STEP? [1.0D-3]'
      PRINT *,'         ( 1 - YES, 0 - NO) '
      PRINT *,' '
      ACCEPT 22,IL
      PRINT *,' '
      IF (IL.NE.1)
    +         GO TO 76
      PRINT *,'         WHAT TIME STEP DO YOU WANT:'
      PRINT *,' '
      ACCEPT *,TSTEP
      PRINT *,' '
76    PRINT *,' '
      PRINT *,'         CHANGE THE TIME MULTIPLE? [3.1622..]'
      PRINT *,'         ( 1 - YES, 0 - NO) '
      PRINT *,' '
      ACCEPT 22,IL
      PRINT *,' '
      IF (IL.NE.1)
```

```
     +       GO TO 77
             PRINT *,' '
             PRINT *,'         INPUT THE TIME MULTIPLE:'
             PRINT *,' '
             ACCEPT *,MULTIME
             PRINT *,' '
77           PRINT *,' '
             PRINT *,'         CHANGE THE FIRST DOSE RATE? [0.0 RAD/S]'
             PRINT *,'         ( 1 = YES, 0 = NO) '
             PRINT *,' '
             ACCEPT 22,IL
             PRINT *,' '
             IF (IL.NE.1)
     +       GO TO 78
             PRINT *,' '
             PRINT *,'         WHAT IS THE DOSE RATE, RAD/S: '
             PRINT *,' '
             ACCEPT *,DSRATE
78           PRINT *,' '
             PRINT *,'         CHANGE THE SECOND DOSE RATE? [0.0 RAD/S]'
             PRINT *,'         ( 1 = YES, 0 = NO) '
             PRINT *,' '
             ACCEPT 22,IL
             PRINT *,' '
             IF (IL.NE.1)
     +       GO TO 79
             PRINT *,' '
             PRINT *,'         WHAT IS THE DOSE RATE, RAD/S: '
             PRINT *,' '
             ACCEPT *,DHRATE
             PRINT *,' '
C******************************************************************
C        GET REQUESTED DATA SETS
C******************************************************************
79           PRINT *,' '
             PRINT *,'         CHOOSE THE REACTION SETS DESIRED,'
             PRINT *,'         BE SURE THE TEMPERATURE OF THE DATA SET'
             PRINT *,'         IS MATCHED TO THE TEMPERATURE GIVEN ABOVE'
             PRINT *,' '
             DO 10 I = 1,20
             PRINT *,'THE FOLLOWING DATA SETS ARE AVAILABLE:'
             PRINT *,' '
             PRINT *,'         RADICAL              EQUILIBRIUM '
             PRINT *,'         REACTIONS              REACTIONS'
             PRINT *,' '
             PRINT *,'1) WATER 25-300 C    10) FE   EQUIL. 25 C'
             PRINT *,'2) IRON  25-300 C    11) FE   EQUIL. 90 C'
             PRINT *,'3) CU    25-300 C    12) NI   EQUIL. 25 C'
             PRINT *,'4) NI    25-300 C    13) NI   EQUIL. 90 C'
             PRINT *,'5) CL    25-300 C    14) CU   EQUIL. 25 C'
             PRINT *,'6) BR    25-300 C    15) CU   EQUIL. 90 C'
             PRINT *,'7) SO4   25-300 C    16) AIR  EQUIL. 25 C'
             PRINT *,'8) AIR   25-300 C    17) H2O  EQUIL. 25 C'
             PRINT *,'9) CO2   25-300 C    18) H2O  EQUIL. 90 C'
             PRINT *,'                     19) FLARE EXAMPLE'
             PRINT *,'                     20) DIFFUSION EXAMPLE'
             PRINT *,' '
             PRINT *,' INPUT A DATA SET NUMBER: '
             PRINT *,' (  0  TO STOP ADDING DATA SETS,'
             PRINT *,'    -1  TO QUIT PROGRAM)'
```

```
          ACCEPT *,ISETS(I)
          IF(ISETS(I).LT.0) GO TO 999
          IF(ISETS(I).EQ.0) THEN
          IISETS = I-1
          GO TO 20
          ENDIF
10        CONTINUE
C****************************************************************************
C         BEGIN OUTER LOOP TO PROCESS ALL DATA SETS
C****************************************************************************
20        DO 30 ILJ = 1, IISETS
          PRINT *,'PROCESSING DATA FROM FILE: ',DFILNAME(ISETS(ILJ))
          OPEN (5,FILE=DFILNAME(ISETS(ILJ)),STATUS='OLD')
          READ (5,3) NEQN,NRTNN
          IF(ILJ.EQ.1) GO TO 221
          DO 221 I = 1,NEQ
          ARSN(I) = '          '
          GHN(I) = 0.
          GN(I) = 0.
          DIFN(I) = 0.
          IAFLAG(I)=0
          IBFLAG(I)=0
221       CONTINUE
          READ (5,NML=NAMESN)
C****************************************************************************
C         SKIP SOME OF THE WORK FOR THE FIRST DATA SET
C****************************************************************************
          IF(ILJ.EQ.1) THEN
          NEQ = NEQN
          DO 21 I = 1,NEQ
          ARS(I) = ARSN(I)
          ARSN(I) = '          '
          G(I) = GN(I)
          GN(I) = 0.
          GH(I) = GHN(I)
          GHN(I) = 0.
          DIF(I) = DIFN(I)
          DIFN(I) = 0.
21        CONTINUE
          GO TO 26
          ENDIF
C****************************************************************************
C         AVOID DUPLICATE SPECIES WITH THE NEXT SEGMENT
C****************************************************************************
          NEQI = 0
          DO 229 J = 1,NEQN
                    DO 228 K = 1,NEQ
                              IF(ARS(K).EQ.ARSN(J)) THEN
                              NEQI = NEQI + 1
                              IAFLAG(J) = K
                              GO TO 229
                    ELSE
                              IAFLAG(J)=0
                              IBFLAG(J-NEQI)=J
                    ENDIF
228       CONTINUE
229       CONTINUE
          DO 24 K = 1,NEQN - NEQI
                    ARS(NEQ+K) = ARSN(IBFLAG(K))
                    G(NEQ+K) = GN(IBFLAG(K))
```

```
                              GH(NEQ+K) = GHN(IBFLAG(K))
                              DIF(NEQ+K) = DIFN(IBFLAG(K))
24            CONTINUE
C             NEQ = NEQ + NEQN - NEQI
C*************************************************************************
C             READ IN THE REACTIONS AND RESEQUENCE
C*************************************************************************
26            DO 140 I=NRTN+1,NRTN+NRTNN
              READ(5,100)CJP(I),(IR(I,K),K=1,3),(IP(I,K),K=1,4),RC(I),EA(I)
C*************************************************************************
C             IF THE FLAG INDICATES (IFLAG=1) THAT A PHASE EQUILIBRIUM
C             REACTION WAS DETECTED FOR THE LAST REACTION
C             (RC(I - 1) = -1), CALCULATE THE REACTION RATE USING A
C             HENRY'S LAW APPROACH FOR REACTION (I) GIVEN THAT
C             RC(I - 1) = 100.
C*************************************************************************
C             IF(IFLAG.EQ.1) THEN
C             RC(I)=100.D0*VL/(VL+VG*(RC(I)/(RU*TEM)))
C             IFLAG=0
C             ENDIF
C*************************************************************************
C             SET THE FIRST RATE CONSTANT OF A PHASE PARTITIONING TO 100.
C             (THIS SHOULD BE A LIQUID TO GAS PHASE REACTION FOLLOWED BY
C             A GAS TO LIQUID PHASE SO THAT THE PROGRAM PROPERLY ACCOUNTS
C             FOR THE PARTITIONING)
C*************************************************************************
C             IF(RC(I).LT.0) THEN
C             RC(I)=100.D0
C             IFLAG=1
C             ENDIF
C*************************************************************************
C             PERFORM RESEQUENCING OF REACTANTS
C*************************************************************************
              IF(ILJ.EQ.1) GO TO 140
              DO 112 K = 1,3
                      IF(IR(I,K).EQ.0) GO TO 112
                      IF(IAFLAG(IR(I,K)).NE.0) THEN
                              IR(I,K)=IAFLAG(IR(I,K))
                      ELSE
                      IZB = 1
                      DO 111 JT =1,NEQN-NEQI
                      IF(IBFLAG(JT).NE.IR(I,K)) IZB = IZB + 1
                      IF(IBFLAG(JT).EQ.IR(I,K)) THEN
                      IR(I,K)=NEQ+IZB
                      GO TO 112
                      ENDIF
111                   CONTINUE
                      ENDIF
112           CONTINUE
              DO 114 K = 1,4
                      IF(IP(I,K).EQ.0) GO TO 114
                      IF(IAFLAG(IP(I,K)).NE.0) THEN
                              IP(I,K)=IAFLAG(IP(I,K))
                      ELSE
                      IZB = 1
                      DO 1111 JT =1,NEQN-NEQI
                      IF(IBFLAG(JT).NE.IP(I,K)) IZB = IZB + 1
                      IF(IBFLAG(JT).EQ.IP(I,K))THEN
                              IP(I,K)=NEQ+IZB
                              GO TO 114
```

```fortran
                 ENDIF
1111                CONTINUE
                 ENDIF
114     CONTINUE
140       CONTINUE
C*****************************************************************************
C       END OF DATA SET, GO BACK FOR MORE OR END
C*****************************************************************************
        NRTN = NRTN + NRTNN
        IF(ILJ.NE.1) NEQ = NEQ + NEQN - NEQI
        CLOSE(5)
30      CONTINUE
C*****************************************************************************
C       WRITE THE FIRST DATA FILE
C*****************************************************************************
        OPEN(6,FILE=TNAME,STATUS='NEW')
        WRITE(6,NML=SIZE)
C*****************************************************************************
C       GET INITIAL CONDITIONS
C*****************************************************************************
        PRINT *,' '
        PRINT *,'  THE FOLLOWING IS A LIST OF THE SPECIES.  NOTE'
        PRINT *,'  THE NUMBER OF THE SPECIES TO BE USED FOR '
        PRINT *,'  SENSITIVITY CALCULATIONS'
        PRINT *,' '
        DO 40 I = 1,NEQ
        PRINT *,'CHANGE THE DEFAULT FOR SPECIES ',I,'  ',ARS(I)
        PRINT *,'         [ 1 = YES, 0 = NO] '
        PRINT *,' '
        ACCEPT 22,IL
        IF (IL.NE.1)
     +                                          GO TO 40
        PRINT *,' '
        IF(MESHPTS.NE.0) THEN
        PRINT *,' '
        PRINT *,'  LOWER BOUNDARY CONDITION FOR ABOVE SPECIES:[0.0]'
        PRINT *,' '
        ACCEPT *,BCL(I)
        PRINT *,' '
        PRINT *,'  UPPER BOUNDARY CONDITION FOR ABOVE SPECIES:[0.0]'
        PRINT *,' '
        ACCEPT *,BCR(I)
        PRINT *,' '
        ENDIF
        PRINT *,' '
        PRINT *,'  INITIAL CONDITION FOR ABOVE SPECIES: [0.0]'
        PRINT *,' '
        ACCEPT *,YI(I)
40      PRINT *,' '
        CONTINUE
        PRINT *,' '
        PRINT *,'         WHAT IS THE NUMBER OF THE VARIABLE TO'
        PRINT *,'         BE USE FOR SENSITIVITY CALCULATION:'
        PRINT *,' '
        ACCEPT *,IA
        PRINT *,' '
C*****************************************************************************
        WRITE(6,NML=STATE)
        WRITE(6,NML=NAMES)
        DO 50 I = 1,NRTN
```

```
            WRITE(6,101)CJP(I),(IR(I,K),K=1,3),(IP(I,K),K=1,4),RC(I),EA(I)
    50      CONTINUE
            WRITE (6,2)
            WRITE(6,2)
            DO 60 I = 1,NEQ
            WRITE (6,220) ARS(I), G(I), GH(I), BCL(I), BCR(I), DIF(I)
    60      CONTINUE
            WRITE(6,NML=VALUES)
            CLOSE(6)
            PRINT *,' CREATE A NEW NAMELIST FILE?'
            PRINT *,'          ( 1 = YES, 0 = NO} '
            PRINT *,' '
            ACCEPT 22,IL
            IF (IL.EQ.1) THEN
            OPEN(8, FILE = NLNAME,STATUS='UNKNOWN')
            WRITE(8,NML=LSIN)
            CLOSE(8)
            ENDIF
            PRINT *,' '
            PRINT *,' '
            PRINT *,'         THE FILES ARE COMPLETE, REMEMBER TO EDIT'
            PRINT *,'         "SET.FIL" TO MAKE THE OUTPUT FILES UNIQUE.'
            PRINT *,' '
            PRINT *,'         THE MITIRAD CODE CAN BE RUN BY USING THE'
            PRINT *,'         FOLLOWING COMMAND:'
            PRINT *,' '
            PRINT *,'         $SUBMIT MITIRAD.COM'
            PRINT *,' '
    2       FORMAT(A40)
    3       FORMAT (I4)
    22      FORMAT(I1)
    100      FORMAT(A3,3X,7I3,D18.8/D18.8)
    101      FORMAT(A3,3X,7I3,E15.6/E15.6)
    220     FORMAT(A8,F10.4,4(/8X,D10.3))
    999     STOP
            END
```

```
        PROGRAM MITIAD
C**********************************************************************
C       VERSION: 1.0
C       DATE: 7/27/88
C       ADJOINT EVALUATION DRIVER:
C
C       EVALUATES THE ADJOINTS FOR A FORWARD CALCULATION
C       CALCULATES dJ = x(adjoint)*dx
C**********************************************************************
C       MITIAD IS A POST PROCESSOR TO THE CODE MITIRAD
C
C       THE FILES REQUIRED ARE THE INPUT FILE TO MITIRAD AND
C       THE OUTPUT FILE WRITTEN BY MITIRAD FOR RS/1.  MITIAD PRODUCES
C       TWO OUTPUT FILES, ONE WRITTEN FOR HUMAN CONSUMPTION, AND THE
C       OTHER WRITTEN FOR READING BY RS/1 (AD_TO.RS1) PROCEDURE BPOST_PRE
C**********************************************************************
C       MANY OF THE VARIABLES USED BY MITIAD ARE THE SAME AS DESCRIBED
C       BY THE WRITEUP OF MITIRAD.  NEW VARIABLES ARE BELOW:
C
C       B,C,D   = TWO DIMENSIONAL ARRAYS THAT ARE USED TO CALCULATE
C                 CUBIC SPLINE FITS TO THE DATA PRODUCED IN THE
C                 FORWARD CALCULATION
C
C       BI,CI,DI  TEMPORARY VECTORS FOR PARTS OF B,C,D
C
C       YASPEC(ITER,NEQ) = ARRAY OF ADJOINTS AT TIMES ITER FOR
C                          SPECIES NEQ
C
C       DKY     = TEMPORARY VECTOR FOR DERIVATIVE EVALUATION
C
C  FILES:
C       "SET.FIL"       = CONTAINS NAMES OF INFILE,DUMMY,PLOTFILE,OUTFILE
C       RSFILE          = OUTPUT FILE FOR RS/1 USE
C       SPFILE          = OUTPUT FILE OF SPLINE FITS FOR RS/1
C       INFILE          = INPUT FILE FROM FORWARD CALCULATION
C       OUTFILE         = OUTPUT FILE FOR HUMAN CONSUMPTION
C       PLOTFILE        = INPUT FILE GENERATED BY DYNEQL
C
C  SUBROUTINES:
C
C       ADFUN           = CALCULATES ADJOINTS AT SPECIFIED TIMES
C       JAC             = FULL JACOBIAN MATRIX OF FORWARD CALCULATION
C       SPLINEM         = INTERFACE BETWEEN MITIAD AND SPLINE
C       SPLINE          = ROUTINE DESCRIBED BY FORSYTHE, MALCOLM, AND MOLER
C                         FOR GENERATING CUBIC SPLINE FITS TO DATA
C       READIN          = SAME AS FOR MITIRAD
C       LSODES          = SEE LSODES WRITEUP
C       INTDY           = SEE LSODES WRITEUP
C       ASEVAL          = EVALUATES SPLINE FITS GENERATED BY SPLINE
C
C  FUNCTIONS:
C
C       CINTY           = INTEGRAND EVALUATION FUNCTION FOR DQUANC8
C       RESP            = RESPONSE FUNCTION FOR ADJOINT CALCULATIONS
C       SEVAL           = ROUTINE DESCRIBED BY FORSYTHE, MALCOLM AND MOLER
C                         FOR EVALUATION SPLINE FITS GENERATED BY SPLINE
C**********************************************************************
C   SAS 12/19/88
        INCLUDE 'PARAMETER.BLK'
        COMMON/CRESP/ IA
```

```
      COMMON/ADVAL/ YI, TIM, YSPEC, ITER, B, C, D
    EXTERNAL ADFUN
      INCLUDE 'DIMENSION.BLK'
C23456789012345678901234567890123456789012345678901234567890123456789012
      DIMENSION DKY(ID2),   PDJ(ID2),
   +  YASPEC(ID1,ID1),     YI(ID2), B(ID2,ID2), C(ID2,ID2), D(ID2,ID2),
   +   YSPEC(ID2,ID2),      Y(ID2),    YF(ID2), RWORK(LRW), IWORK(LIW),
   +        TIM(ID2),  ATIM(ID1), RRTOL(ID2)
      CHARACTER*35 INFILE,OUTFILE,PLOTFILE,RSFILE,SPFILE, NLFILE
      REAL*8 MULTIME
      CHARACTER*8 CHRESP, DUM
      CHARACTER*9 TDATE
      INTEGER FLAG
      NAMELIST /LSIN/IOPT, ITASK, RTOL, ITOL, ATOL, RWORK, IWORK
   +                , IDERV, ISTATE, NPOT, NORSOUT, ICADRE, NORSDIA
   +                , RITOL,  RATOL, NPOINTS, MF
      NAMELIST /STATE/ VL, VG, TEM, TEMR, DSRATE, DHRATE, TOUT,
   +                  TFINAL, TSTEP, MULTIME, IA, VEL
      NAMELIST /SIZE/ NEQ, NRTN, MESHPTS, XHIGH, XLOW
      NAMELIST /NAMES/ ARS
     DATA ATOL/1.0D-10/,    MF/222/,   ITOL/3/, ISTATE/1/,   ITASK/1/,
   +      RTOL/1.D-4/, CHRESP/'O'/,    IA/0/,  IOPT /0/,      IND/0/,
   +      IATER/0/,       IDERV/0/,NPOT /1/,NORSOUT/0/,NPOINTS/100/,
   +      RATOL/1.D-4/, TEM/298./,TEMR/298./
C**********************************************************************
C     READ THE FORWARD CALCULATION INFORMATION
C**********************************************************************
      OPEN(5,FILE = 'SET.FIL', STATUS = 'OLD')
      READ (5,100) INFILE, OUTFILE, PLOTFILE, NLFILE, OUTFILE,
   +     RSFILE, SPFILE
100   FORMAT(A35)
      CLOSE(5)
      OPEN(5, FILE = INFILE, STATUS = 'OLD')
      OPEN(6, FILE = OUTFILE, STATUS = 'NEW')
      READ (5,SIZE)
      READ (5,NML=STATE)
      NPTS=NEQ*MESHPTS
      XINC = (XHIGH-XLOW)/DFLOAT(MESHPTS-1)
      XINCSQ = XINC**2.
130   FORMAT(A1)
      READ (5,NML=NAMES)
      DO 150 II = 1,NEQ
           DO 150 IJ = 1,NRTN
                 KOEF(IJ,II) = 0
                 NJ (IJ,II) = 0
150   CONTINUE
C**********************************************************************
C     READ THE NAMELIST FILE NLFILE FOR NAMELIST LSIN
C**********************************************************************
      OPEN(1,FILE=NLFILE,STATUS = 'OLD')
      READ(1, NML = LSIN)
      CLOSE(1)
      IF(IA.EQ.0) THEN
      PRINT *,' NO MATCHING RESPONSE VARIABLE, PROGRAM TERMINATED'
      STOP
      ENDIF
      CALL DATE(TDATE)
      CALL TIME(DUM)
      WRITE(6,278) TDATE,DUM,ARS(IA),ATOL,RTOL,MF,NPOT,NPOINTS,
   +  INFILE,
```

```
      +    OUTFILE,RSFILE,NLFILE,SPFILE,DHRATE,DSRATE,VL,VG,TEM                      :
  278      FORMAT(//20X,'SOLADJ RUN RESULTS',5X,A9,5X,A8//,
      +    5X,'RESPONSE IS CALCULATED WITH RESPECT TO ',A8,//
      +    5X,'RUN PARAMETERS:',/
      +    5X,'ABSOLUTE TOLERANCE = ',1X,D15.7/,
      +    5X,'RELATIVE TOLERANCE = ',1X,D15.7/,
      +    5X,'METHOD FLAG        = ',1X,I3/,
      +    5X,'TIME STEP OPTION   = ',1X,I3/,
      +    5X,'NUMBER INTEGRAL PTS= ',1X,I3/,
      +    5X,'INPUT FILE         = ',1X,A35/,
      +    5X,'OUTPUT FILE        = ',1X,A35/,
      +    5X,'TRANSFER FILE      = ',1X,A35/,
      +    5X,'NAMELIST FILE      = ',1X,A35/,
      +    5X,'SPLINE DATA FILE   = ',1X,A35/,
      +    5X,'HIGH-LET DOSE RATE = ',1X,D15.7/,
      +    5X,'LOW-LET DOSE RATE  = ',1X,D15.7/,
      +    5X,'LIQUID VOLUME      = ',1X,D15.7/,
      +    5X,'GAS VOLUME         = ',1X,D15.7/,
      +    5X,'TEMPERATURE (K)    = ',1X,F7.2/)
           CALL READIN(NEQ,Y)
           DO 151 I = 1,NEQ
  151         YI(I) = Y(I)
C***********************************************************************
C         REPROCESS ALL INDEXING VARIABLES
C***********************************************************************
           YF(NEQ+1) = 1.0D0
           DO 170 I = 1,NRTN
                   IN1(I) = NEQ+1
                   IN2(I) = NEQ+1
                   IN3(I) = NEQ+1
                   IND = 0
                   IFLG = 0
                   DO 160 J = 1, NEQ
                   IF(NJ(I,J).EQ.-1.AND.IND.EQ.0) THEN
                   IN1(I) = J
                   IFLG = IFLG + 1
                   ENDIF
                   IF(NJ(I,J).EQ.-1.AND.IND.EQ.1) THEN
                   IN2(I) = J
                   IFLG = IFLG + 1
                   ENDIF
                   IF(NJ(I,J).EQ.-1.AND.IND.EQ.2) THEN
                   IN3(I) = J
                   IFLG = IFLG + 1
                   ENDIF
                   IF(NJ(I,J).EQ.-2.AND.IND.EQ.0) THEN
                   IN1(I) = J
                   IN2(I) = J
                   IFLG = IFLG + 2
                   ENDIF
                   IF(NJ(I,J).EQ.-2.AND.IND.EQ.1) THEN
                   IN2(I) = J
                   IN3(I) = J
                   IFLG = IFLG + 2
                   ENDIF
                   IND = IND + IFLG
                   IFLG = 0
  160              CONTINUE
  170      CONTINUE
C***********************************************************************
```

```
C          READ THE RESULTS FROM THE FORWARD CALCULATION AND THE
C          SPLINE FITS
C**********************************************************************
          OPEN (8, FILE - PLOTFILE, STATUS - 'OLD')
          READ (8, 201) NEQ, ITER, MESHPTS, ILOGLOG
          DO 190 I - 1,NEQ
                READ(8, 202) ARS(I)
190       CONTINUE
          IF(MESHPTS.NE.0) THEN
          DO 200 I - 1,ITER
                READ(8,203) TIM(I)
                DO 200 L - 1,MESHPTS
                      K - NEQ*(L-1)
                      DO 200 J - 1,NEQ
                            IF(L.EQ.1)THEN
                                     READ (8, 203) YSPEC(I,J)
                            ELSE
                                     READ (8, 203) YDUM
                            ENDIF
200       CONTINUE
          ELSE
          DO 204 I - 1,ITER
                READ(8,203) TIM(I)
                DO 204 J - 1,NEQ
                      READ (8, 203) YSPEC(I,J)
204       CONTINUE
          ENDIF
201       FORMAT(1X,I3)
202       FORMAT(1X,A8)
203       FORMAT(1X,E21.14)
          CLOSE(8)
C**********************************************************************
C          FIT THE FORWARD CALCULATIONS TO CUBIC SPLINES
C**********************************************************************
          IF(ILOGLOG.NE.1) THEN
          PRINT *,' DATA FROM MITIRAD IS NOT CORRECT FOR CURRENT'
          PRINT *,' VERSION OF MITIAD ILOGLOG - ',ILOGLOG
          STOP
          ENDIF
C          CALL SPLINEM(NEQ,ITER,YSPEC,TIM,B,C,D)
C**********************************************************************
C          PRINT SPLINE RESULTS
C**********************************************************************
C          OPEN (9, FILE - SPFILE, STATUS - 'NEW')
C          DO 210 J - 1,NEQ
C                WRITE(9, 211) J,(B(I,J),C(I,J),D(I,J), I-1,ITER)
C210       CONTINUE
C211       FORMAT(1X,I3/,3(3X,E18.8))
C          CLOSE(9)
       WRITE (6,290)
       T - 10.D0**TIM(ITER)
       TM2 - T/DFLOTJ(NPOINTS)
       TOUT - T
C**********************************************************************
c adjoints initial conditions always zero, SET UP RELATIVE TOLERANCE
C**********************************************************************
       DO 277 I - 1, NEQ
          Y(I) - 0.0D0
          RRTOL(I) - RTOL
                IF(MESHPTS.EQ.0) BCL(I) - 2.D0
```

```
    277  CONTINUE
         RRTOL(IA) = RATOL
C***************************************************************************
c  start the reverse loop
C***************************************************************************
         DO 810 I = NPOINTS, 0, -1
C***************************************************************************
C  start an inner loop USING INDEX 281 FOR EXCESSIVE WORK
C***************************************************************************
C        DO NOT LET THE SOLVER OVERSHOOT THE DATA
C
C        NPOT = 0   MEANS NO CONTROL ON ALLOWABLE STEP, ITASK = 1
C        NPOT = 1   STEP ONLY TO NEXT TIME POINT, ITASK = 4
C        NPOT = 2   DO NOT STEP PAST T = 0.0D0, ITASK = 4
C***************************************************************************
         IF(NPOT.EQ.1) THEN
         ITASK = 4
         RWORK(1) = TOUT
         ENDIF
         IF(NPOT.EQ.2) THEN
         ITASK = 4
         RWORK(1) = 0.0D0
         ENDIF
C***************************************************************************
C        CALL THE ODE SOLVER.  LSODES WILL CALL
C        THE ROUTINES ADFUN AND JACT.
C        OTHER INPUT AND OUTPUT PARAMETERS ARE
C        DEFINED IN THE LSODES WRITE-UP.
C***************************************************************************
         IATER = IATER+1
         IF(I.EQ.0) THEN
         T=0.0
         GO TO 282
         ENDIF
    281  CALL LSODES(ADFUN,NEQ,Y,T,TOUT,ITOL,RRTOL,ATOL,ITASK,ISTATE,IOPT,
        + RWORK,LRW,IWORK,LIW,JACBLANKET,MF)
C***************************************************************************
C        PRINT RESULTS OF PREVIOUS TIME STEP AND INCREMENT
C        THE TIME OR STOP AND PRINT FINAL RESULTS
C***************************************************************************
C        OPTIONAL ROUTINE TO CALL THE DERIVATIVE EVALUATION SUBROUTINE
C        IDERV = DERIVATIVE TO BE CALCULATED, USUALLY 1 OR 2.
C***************************************************************************
    282  CONTINUE
         IF(IDERV.GE.1) THEN
         LYH = IWORK(21)
         CALL INTDY (T,IDERV,RWORK(LYH),NEQ,DKY,FLAG)
         ENDIF
C***************************************************************************
C        PRINT THE RESULTS OF THE LAST ITERATION
C***************************************************************************
         WRITE (6,290)
    291  FORMAT(37X,'OUTPUT')
    290  FORMAT (/80(1H_),/)
         WRITE (6,310) T
         WRITE (6,320) (ARS(K),Y(K),K=1,NEQ)
         WRITE (6,300) IWORK(11)
    300  FORMAT (13H NO. STEPS = ,I5)
    310  FORMAT (25X,'ADJOINTS AT TIME = ',D11.4,/)
    320  FORMAT (2(5X,A8,' = ',D15.6,' **'))
```

```
C************************************************************************
C        OUTPUT FOR DERIVATIVE EVALUATION (OPTIONAL)
C************************************************************************
        IF(IDERV.GE.1) THEN
        WRITE (6,350) IDERV, FLAG
350     FORMAT (/28X,'DERIVATIVES OF ORDER',I1,10X,I2/)
        WRITE (6,320) (ARS(K),DKY(K),K=1,NEQ)
        ENDIF
C************************************************************************
C        ISTATE=-1 IS TOO MUCH WORK (OVER 500 STEPS DURING LAST CALL
C        TO LSODE) THE CODE ASSUMES THAT THE CORRECT SOLUTION WILL BE
C        DETERMINED EVENTUALLY AND SENDS CONTROL BACK TO THE SOLVER
C        FOR MORE WORK
C************************************************************************
        IF(ISTATE.EQ.-1) THEN
        PRINT *,ISTATE, ' WORKING HARD !'
          PRINT *, ' '
        ISTATE=2
        GO TO 281
        ENDIF
C************************************************************************
C        STORE RESULTS FOR OUTPUT TO PLOT FILE
C************************************************************************
        ATIM(IATER) = T
          DO 360 K=1,NEQ
          YASPEC(IATER,K) = Y(K)
360       CONTINUE
C************************************************************************
C        IF ISTATE LT 0 AT THIS POINT, MAJOR ERROR HAS OCCURED AND
C        THE ROUTINE TERMINATES
C************************************************************************
        IF (ISTATE.LT.0) GO TO 380

C************************************************************************
C        INCREMENT THE TIME AND CONTINUE
C************************************************************************
C        IF(I.EQ.1) THEN
C        TOUT = 0.0d0
C        ELSE
        TOUT = DFLOTJ(I-1)*TM2
C        ENDIF
        PRINT *,TOUT
810     CONTINUE
C************************************************************************
C        PRINT THE ADJOINT CALCULATION FINAL RESULTS
C************************************************************************
380     LENRW = IWORK(17)
        LENIW = IWORK(18)
        NST = IWORK(11)
        NFE = IWORK(12)
        NJE = IWORK(13)
          WRITE (6,290)
          WRITE (6,381)
381       FORMAT(29X,'RUN STATISTICS')
          WRITE (6,290)
        WRITE (6,390) LENRW,LENIW,NST,NFE,NJE
390     FORMAT(5X,/25H REQUIRED RWORK SIZE   = ,I5,
     +        5X,/25H IWORK SIZE            = ,I5,
     +        5X,/25H NUMBER OF STEPS       = ,I5,
     +        5X,/25H # OF FUNC.- EVALS.    = ,I5,
```

```
     +          5X,/25H # OF JACOB.- EVALS    = ,I5)
            WRITE (6,400) ISTATE
 400    FORMAT (///22H ERROR HALT...ISTATE =,I3)
C*********************************************************************
C       PREPARE A FILE FOR RS/1, NORSOUT=0 (DEFAULT)
C*********************************************************************
            IF(NORSOUT.EQ.1) GO TO 470
            OPEN(8,FILE = RSFILE,STATUS = 'UNKNOWN')
            WRITE (8,410) NEQ, IATER,MESHPTS, ILOGLOG
 410        FORMAT(1X,I3)
            DO 420 I1 = 1,NEQ
                WRITE(8,430) ARS(I1)
 420        CONTINUE
 430        FORMAT(1X,A8)
            DO 450 I2 = 1,IATER
                WRITE(8,460) ATIM(I2)
                DO 440 I3 = 1,NEQ
                    WRITE(8,460) YASPEC(I2,I3)
 440        CONTINUE
 450        CONTINUE
 460        FORMAT(1X,E21.14)
            CLOSE(8)
 470        CONTINUE
            STOP
            END

        SUBROUTINE ADFUN(NEQ,T,Y,DY)
C*********************************************************************
c   function to compute adjoints to the forward equations
C*********************************************************************
        INCLUDE 'PARAMETER.BLK'
        COMMON/CRESP/ IA
            COMMON/ADVAL/ YI, TIM, YSPEC, ITER, B, C, D
            INCLUDE 'DIMENSION.BLK'
        DIMENSION PDJ(ID2), Y(ID2), DY(ID2), YF(ID2), YI(ID2)
            DIMENSION YSPEC(ID2,ID2)
        IF (T.GT.0.0D0)THEN
        AT = DLOG10(T)
        ELSE
        AT = 0.0D0
        ENDIF
        YF(NEQ+1) = 1.0D0
C*********************************************************************
C       STORE FORWARD VARIABLES IN YF
C*********************************************************************
        DO 10 I = 1,NEQ
            YF(I) =  ASEVAL(AT,I,INC)
 10     CONTINUE
C*********************************************************************
C       LOOP THROUGH ALL ADJOINT EQUATIONS
C*********************************************************************
        DO 110 I = 1, NEQ
C*********************************************************************
C       RESP GIVES THE RESPONSE, JCSA CALLS THE JACOBIAN MATRIX
C       COLUMNWISE
C*********************************************************************
C          DY(I) = -RESP(I,NEQ,T,YF,YI)
            CALL JCSA(NEQ,T,YF,I,PDJ)
            DY(I) = -PDJ(IA)
C*********************************************************************
```

```
C         INNER LOOP SUMS JACOBIAN TERMS * ADJOINT + dL/dk
C*********************************************************************
          DO 100 K = 1, NEQ
              DY(I) = DY(I) - PDJ(K)*Y(K)
100       CONTINUE
110   CONTINUE
      RETURN
      END


C      REAL*8 FUNCTION RESP(K,NEQ,T,YF,YI)
C*********************************************************************
c    generate the appropriate response function
C*********************************************************************
C      include 'PARAMETER.blk'
C       INCLUDE 'DIMENSION.BLK'
C      dimension pdj(id2),yf(id2),yi(id2),y(id2)
C      common/cresp/ia
C      call JAC(neq,t,yf,k,pdj)
C      resp = pdj(IA)
C      return
C      end

       DOUBLE PRECISION FUNCTION ASEVAL(U,IS,INC)
C*********************************************************************
c  evaluate the forward values of the functions for time 'u'
C*********************************************************************
       INCLUDE 'PARAMETER.blk'
       common /adval/ yi(id2),x,q,n,b,c,d
C       COMMON IL
       INCLUDE 'DIMENSION.BLK'
       dimension x(id2),q(id2,id2),b(id2,id2),c(id2,id2),d(id2,id2)
C*********************************************************************
c       this subroutine evaluates the cubic spline function
c
c   seval = y(i) + b(i)*(u-x(i)) + c(i)*(u-x(i))**2 + d(i)*(u-x(i))**3
c
c   where x(i).lt. u .lt. x(i+1), using horner's rule
c
c   if u .lt. x(1) then i = 1 is used
c   if u.ge. x(n) then i = n is used
c
c   input
c        n = the number of data points
c        u = the abscissa at which the spline is to be evaluated
c        x, y = the arrays of data abscissas and ordinates
c        b,c,d = arrays of spline coefficients computed by spline
c
c   A binary search is performed to determine the proper interval.
C*********************************************************************
       data I/1/
       if (u.le.0.d0) go to 40
C*********************************************************************
c  binary search, USUALLY SEQUENTIAL BUT SLIP BACK ONE INTERVAL ANYWAY
C*********************************************************************
C       I = 1
       IF(IS.GT.1) THEN
       I=INC
       GO TO 30
       ENDIF
```

```
   10      i = 1
           j = n+1
   20      k = (i+j)/2
           if(u.lt.x(k)) j = k
           if(u.ge.x(k)) i = k
           if(j.gt.i+1) go to 20
           INC = I
C*****************************************************************
c   evaluate spline
C*****************************************************************
   30      CONTINUE
           DX = U - X(I)
C          aseval = q(i,is) + dx*(b(i,is) + dx*(c(i,is) + dx*d(i,is)))
C          aseval = 10.d0**aseval
           IF(I.EQ.ITER) THEN
           ASEVAL = Q(I,IS)
           GO TO 35
           ENDIF
           ASEVAL = Q(I,IS) + DX*(Q(I+1,IS)-Q(I,IS))/(X(I+1)-X(I))
   35      ASEVAL = 10.D0**ASEVAL
           return
   40      Aseval = yi(is)
           return
           end


           SUBROUTINE JCSA (NEQ,T,Y,J,PDJ)
C*****************************************************************
C          VERSION: MIT 2.0
C          CODE CUSTODIAN: S. A. SIMONSON
C          DATE OF LAST CHANGE TO SUBROUTINE: 11/17/86
C          JAC IS PART OF THE SOLADJ CODE
C          JAC IS CALLED BY THE SUBROUTINE ADFUN
C*****************************************************************
C          JAC CALCULATES THE COLUMN VECTORS FOR THE JACOBIAN MATRIX
C*****************************************************************
           include 'PARAMETER.blk'
           INCLUDE 'DIMENSION.BLK'
           dimension  PDJ(ID2), Y(ID2)
           DO 100 I=1,NEQ
           IF(I.EQ.J.AND.BCL(I).NE.2.0D0) THEN
           PDJ(I) = -2.D0*DIF(I)
           ELSE
           PDJ(I) = 0.0D0
           ENDIF
             DO 100 K=1,NRTN
             IF(KOEF(K,J).EQ.0.OR.KCEF(K,I).EQ.0) GOTO 100
             A = RC(K)*DFLOTJ(KOEF(K,I)*JIABS(KOEF(K,J)))
             IM = IN1(K)
             IN = IN2(K)
             IO = IN3(K)
C*****************************************************************
C      CATCH SECOND ORDER REACTIONS
C*****************************************************************
             IF((IM.EQ.IN.OR.IO.EQ.IN).AND.(IN.EQ.J))THEN
             PDJ(I) = PDJ(I)+A*Y(IM)*Y(IO)
             GO TO 100
             ENDIF
             IF (IM.EQ.J) THEN
             PDJ(I) = PDJ(I)+A*Y(IN)*Y(IO)
             GOTO 100
```

```
            ENDIF
C*********************************************************************************
C       FIRST ORDER REACTIONS
C*********************************************************************************
            IF (IO.EQ.J) THEN
            PDJ(I) = PDJ(I)+A*Y(IN)*Y(IM)
            GOTO 100
            ENDIF
            IF (IN.EQ.J) PDJ(I) = PDJ(I)+A*Y(IM)*Y(IO)
100     CONTINUE
        RETURN
        END
        SUBROUTINE JACL (NEQ,T,Y,ML,MU,PD,NROWPD)
C
C*********************************************************************************
C       VERSION: MIT 2.0
C       CODE CUSTODIAN: S. A. SIMONSON
C       DATE OF LAST CHANGE TO SUBROUTINE: 11/20/87
C       JACL IS PART OF THE DYNEQL CODE PACKAGE
C       JACL IS CALLED BY THE SUBROUTINE LSODE
C*********************************************************************************
C       JACL CALCULATES THE FULL JACOBIAN MATRIX
C*********************************************************************************
C
        INCLUDE 'PARAMETER.BLK'
        INCLUDE 'DIMENSION.BLK'
        DIMENSION PD(NROWPD,ID2), Y(ID2)
        DO 100 J=1,NEQ
        DO 100 I=1,NEQ
          DO 100 K=1,NRTN
          IF(KOEF(K,J).EQ.0.OR.KOEF(K,I).EQ.0) GOTO 100
          A = RC(K)*DFLOTJ(KOEF(K,I)*JIABS(KOEF(K,J)))
          IM = IN1(K)
          IN = IN2(K)
          IO = IN3(K)
C
C*********************************************************************************
C       CATCH SECOND ORDER REACTIONS
C*********************************************************************************
C
            IF((IM.EQ.IN.OR.IO.EQ.IN).AND.(IN.EQ.J))THEN
            PD(I,J) = PD(I,J)+A*Y(IM)*Y(IO)
            GO TO 100
            ENDIF
            IF (IM.EQ.J) THEN
            PD(I,J) = PD(I,J)+A*Y(IN)*Y(IO)
            GOTO 100
            ENDIF
C
C*********************************************************************************
C       FIRST ORDER REACTIONS
C*********************************************************************************
C
            IF (IO.EQ.J) THEN
            PD(I,J) = PD(I,J)+A*Y(IN)*Y(IM)
            GOTO 100
            ENDIF
            IF (IN.EQ.J) PD(I,J) = PD(I,J)+A*Y(IM)*Y(IO)
100     CONTINUE
        RETURN
```

END

```
      PROGRAM LINTY
C******************************************************************
C      IMPORTANCE INTEGRATOR:
C
C      CALCULATES dJ/dk(i) where k(i) is the rate constant of the i-th
C      reaction rate.
C******************************************************************
C      LINTY IS A POST PROCESSOR TO THE CODE MITIAD
C
C      THE FILES REQUIRED ARE THE INPUT FILE TO MITIRAD AND
C      THE OUTPUT FILE WRITTEN BY MITIRAD FOR RS/1.  MITIAD PRODUCES
C      TWO OUTPUT FILES, ONE WRITTEN FOR HUMAN CONSUMPTION, AND THE
C      OTHER WRITTEN FOR READING BY RS/1 PROCEDURE LDIAGNOSTICS
C******************************************************************
C      MANY OF THE VARIABLES USED BY MITIAD ARE THE SAME AS DESCRIBED
C      BY THE WRITEUP OF MITIRAD.  NEW VARIABLES ARE BELOW:
C
C      B,C,D    - TWO DIMENSIONAL ARRAYS THAT ARE USED TO CALCULATE
C                 CUBIC SPLINE FITS TO THE DATA PRODUCED IN THE
C                 FORWARD CALCULATION
C      BI,CI,DI   TEMPORARY VECTORS FOR PARTS OF B,C,D
C      YASPEC(ITER,NEQ) -  ARRAY OF ADJOINTS AT TIMES ITER FOR
C                          SPECIES NEQ
C      DKY      - TEMPORARY VECTOR FOR DERIVATIVE EVALUATION AND
C                 TO HOLD INTEGRANDS FOR SPLINE FITS
C
C  FILES:
C      "SET.FIL"     - CONTAINS NAMES OF INFILE,DUMMY,PLOTFILE,OUTFILE
C      "AD_TO.RS1"   - OUTPUT FILE FOR RS/1 USE
C      "SPLINE.FILE" - OUTPUT FILE OF SPLINE FITS FOR RS/1
C      "ADIA.RS1"    - DIAGNOSTIC FILE FOR RS/1
C      INFILE        - INPUT FILE FROM FORWARD CALCULATION
C      OUTFILE       - OUTPUT FILE FOR HUMAN CONSUMPTION
C      PLOTFILE      - INPUT FILE GENERATED BY DYNEQL
C
C  SUBROUTINES:
C
C      ADFUN       - CALCULATES ADJOINTS AT SPECIFIED TIMES
C      JAC         - FULL JACOBIAN MATRIX OF FORWARD CALCULATION
C      SPLINEM     - INTERFACE BETWEEN SOLADJ AND SPLINE
C      SPLINE      - ROUTINE DESCRIBED BY FORSYTHE, MALCOLM, AND MOLER
C                    FOR GENERATING CUBIC SPLINE FITS TO DATA
C      INPUT       - SAME AS FOR DYNEQL
C      LSODE       - SEE LSODE WRITEUP
C      INTDY       - SEE LSODE WRITEUP
C      ASEVAL      - EVALUATES SPLINE FITS GENERATED BY SPLINE
C
C  FUNCTIONS:
C
C      CINTY       - INTEGRAND EVALUATION FUNCTION FOR DQUANC8
C      RESP        - RESPONSE FUNCTION FOR ADJOINT CALCULATIONS
C      SEVAL       - ROUTINE DESCRIBED BY FORSYTHE, MALCOLM AND MOLER
C                    FOR EVALUATION SPLINE FITS GENERATED BY SPLINE
C******************************************************************
C   SAS 12/19/88
      INCLUDE 'PARAMETER.BLK'
C      COMMON/INTER/ BI,CI,DI,DKY,TALE(ID1)
C      COMMON/CRESP/ IA
      COMMON/ADVAL/ YI, TIM, YSPEC, ITER, B, C, D, IATER
      COMMON/PASSER/IR,IP,EA
```

```
C       EXTERNAL  CINTY
        INCLUDE 'DIMENSION.BLK'
C2345678901234567890123456789012345678901234567890123456789012
        DIMENSION     DKY(ID1),    PDJ(ID2), IR(ID1,3), IP(ID1,4),
     +  YASPEC(ID1,ID2),    YI(ID2), B(ID2,ID2), C(ID2,ID2), D(ID2,ID2),
     +  YSPEC(ID2,ID2),     Y(ID2),    YF(ID2), CAD(8),CMAD(8),
     +         TIM(ID2),   ATIM(ID1),   TALE(ID1), FIA(8,ID1),IPLK(4)
C    +  BI(ID2),     CI(ID2),     DI(ID2),RWORK(LRW), IWORK(LIW),
        CHARACTER*35 INFILE,OUTFILE,PLOTFILE,NLFILE,RSFILE,SPFILE,
     +                DIAGFILE
        REAL*8 MULTIME
        CHARACTER*8 CHRESP, DUM
        CHARACTER*9 TDATE
        INTEGER FLAG
        NAMELIST /LSIN/IOPT, ITASK, RTOL, ITOL, ATOL, RWORK, IWORK
     +               , IDERV, ISTATE, NPOT, NORSOUT, ICADRE, NORSDIA
     +               , RITOL,  RATOL, NPOINTS, MF
        NAMELIST /STATE/ VL, VG, TEM, TEMR, DSRATE, DIIRATE, TOUT,
     +                TFINAL, TSTEP, MULTIME, IA, VEL
        NAMELIST /SIZE/ NEQ, NRTN, MESHPTS, XHIGH, XLOW
        NAMELIST /NAMES/ ARS
        DATA ATOL/1.0D-15/,     MF/222/,   ITOL/3/, ISTATE/1/,   ITASK/1/,
     +       RTOL/1.D-5/, CHRESP/'O'/,     IA/0/,   IOPT /0/,      IND/0/,
     +         IATER/0/,      IDERV/0/,NPOT /1/,NORSOUT/0/,NPOINTS/100/,
     +       RATOL/1.D-5/, TEM/298./,TEMR/298./,NORSDIA/1/,ITPRIME/0/
C******************************************************************
C       READ THE FORWARD CALCULATION INFORMATION
C******************************************************************
        OPEN(5,FILE = 'SET.FIL', STATUS = 'OLD')
        READ (5,100) INFILE, OUTFILE, PLOTFILE, NLFILE, OUTFILE,
     +       RSFILE, SPFILE, OUTFILE, DIAGFILE
100     FORMAT(A35)
        CLOSE(5)
        OPEN(5, FILE = INFILE, STATUS = 'OLD')
        OPEN(6, FILE = OUTFILE, STATUS = 'NEW')
        READ (5,SIZE)
        READ (5,NML=STATE)
        NPTS=NEQ*MESHPTS
        XINC = (XHIGH-XLOW)/DFLOAT(MESHPTS-1)
        XINCSQ = XINC**2.
130     FORMAT(A1)
        READ (5,NML=NAMES)
        DO 150 II = 1,NEQ+1
             DO 150 IJ = 1,NRTN
                   KOEF(IJ,II) = 0
                   NJ (IJ,II) = 0
150     CONTINUE
C******************************************************************
C       READ THE NAMELIST FILE NLFILE FOR NAMELIST LSIN
C******************************************************************
        OPEN(1,FILE=NLFILE,STATUS = 'OLD')
        READ(1, NML = LSIN)
        CLOSE(1)
        IF(IA.EQ.0) THEN
        PRINT *,' NO MATCHING RESPONSE VARIABLE, PROGRAM TERMINATED'
        STOP
        ENDIF         q
        CALL DATE(TDATE)
        CALL TIME(DUM)
        WRITE(6,278) TDATE,DUM,ARS(IA),ATOL,RTOL,MF,NPOT,NPOINTS,
```

```
      +   INFILE,
      +   OUTFILE,RSFILE,NLFILE,SPFILE,DHRATE,DSRATE,VL,VG,TEM
 278      FORMAT(//20X,'LINTY RUN RESULTS',5X,A9,5X,A8//,
      +   5X,'RESPONSE IS CALCULATED WITH RESPECT TO ',A8,//
      +   5X,'RUN PARAMETCRS:',/
      +   5X,'ABSOLUTE TOLERANCE = ',1X,D15.7/,
      +   5X,'RELATIVE TOLERANCE = ',1X,D15.7/,
      +   5X,'METHOD FLAG        = ',1X,I3/,
      +   5X,'TIME STEP OPTION   = ',1X,I3/,
      +   5X,'NUMBER INTEGRAL PTS= ',1X,I3/,
      +   5X,'INPUT FILE         = ',1X,A35/,
      +   5X,'OUTPUT FILE        = ',1X,A35/,
      +   5X,'TRANSFER FILE      = ',1X,A35/,
      +   5X,'NAMELIST FILE      = ',1X,A35/,
      +   5X,'SPLINE DATA FILE   = ',1X,A35/,
      +   5X,'HIGH-LET DOSE RATE = ',1X,D15.7/,
      +   5X,'LOW-LET DOSE RATE  = ',1X,D15.7/,
      +   5X,'LIQUID VOLUME      = ',1X,D15.7/,
      +   5X,'GAS VOLUME         = ',1X,D15.7/,
      +   5X,'TEMPERATURE (K)    = ',1X,F7.2/)
          CALL READIN(NEQ,Y)
          DO 151 I = 1,NEQ
 151          YI(I) = Y(I)
C**************************************************************************
C         REPROCESS ALL INDEXING VARIABLES
C**************************************************************************
          YF(NEQ+1) = 1.0D0
          DO 170 I = 1,NRTN
                  IN1(I) = NEQ+1
                  IN2(I) = NEQ+1
                  IN3(I) = NEQ+1
                  IND = 0
                  IFLG = 0
                  DO 160 J = 1, NEQ
                  IF(NJ(I,J).EQ.-1.AND.IND.EQ.0) THEN
                  IN1(I) = J
                  IFLG = IFLG + 1
                  ENDIF
                  IF(NJ(I,J).EQ.-1.AND.IND.EQ.1) THEN
                  IN2(I) = J
                  IFLG = IFLG + 1
                  ENDIF
                  IF(NJ(I,J).EQ.-1.AND.IND.EQ.2) THEN
                  IN3(I) = J
                  IFLG = IFLG + 1
                  ENDIF
                  IF(NJ(I,J).EQ.-2.AND.IND.EQ.0) THEN
                  IN1(I) = J
                  IN2(I) = J
                  IFLG = IFLG + 2
                  ENDIF
                  IF(NJ(I,J).EQ.-2.AND.IND.EQ.1) THEN
                  IN2(I) = J
                  IN3(I) = J
                  IFLG = IFLG + 2
                  ENDIF
                  IND = IND + IFLG
                  IFLG = 0
 160              CONTINUE
 170      CONTINUE
```

```
C********************************************************************************
C        READ THE RESULTS FROM THE FORWARD CALCULATION
C********************************************************************************
         OPEN (8, FILE = PLOTFILE, STATUS = 'OLD')
         READ (8, 201) NEQ, ITER, MESHPTS, ILOGLOG
         DO 190 I = 1,NEQ
                 READ(8, 202) ARS(I)
190      CONTINUE
         IF(MESHPTS.NE.0) THEN
         DO 200 I = 1,ITER
                 READ(8,203) TIM(I)
                 DO 200 L = 1,MESHPTS
                         K = NEQ*(L-1)
                         DO 200 J = 1,NEQ
                                 IF(L.EQ.1)THEN
                                         READ (8, 203) YSPEC(I,J)
                                 ELSE
                                         READ (8, 203) YDUM
                                 ENDIF
200      CONTINUE
         ELSE
         DO 204 I = 1,ITER
                 READ(8,203) TIM(I)
                 DO 204 J = 1,NEQ
                         READ (8, 203) YSPEC(I,J)
204      CONTINUE
         ENDIF
201      FORMAT(1X,I3)
202      FORMAT(1X,A8)
203      FORMAT(1X,E21.14)
         CLOSE(8)
C********************************************************************************
C        FIT THE FORWARD CALCULATIONS TO CUBIC SPLINES
C********************************************************************************
         IF(ILOGLOG.NE.1) THEN
         PRINT *,' DATA FROM DYNEQL IS NOT CORRECT FOR CURRENT'
         PRINT *,' VERSION OF SOLADJ  ILOGLOG = ',ILOGLOG
         STOP
         ENDIF
C        CALL SPLINEM(NEQ,ITER,YSPEC,TIM,B,C,D)
C********************************************************************************
C        READ THE RESULTS FROM THE ADJOINT CALCULATION
C********************************************************************************
         OPEN (8, FILE = RSFILE, STATUS = 'OLD')
         READ (8, 201) NEQ, IATER,MESHPTS, ILOGLOG
         DO 1190 I = 1,NEQ
                 READ(8, 202) ARS(I)
1190     CONTINUE
         DO 1200 I = 1,IATER
                 READ(8,203) ATIM(I)
                 DO 1200 J = 1,NEQ
                         READ (8, 203) YASPEC(I,J)
1200     CONTINUE
         CLOSE(8)
C&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
C        BEGIN THE FINAL INTEGRATIONS
C&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
C        CYCLE THROUGH ALL RATE CONSTANTS, G-VALUES, AND DOSE RATES
C         FOR TOTAL SENSITIVITIES
C********************************************************************************
```

```
            ITESTLAST=1
            TPRIME = 0.0D0
            WRITE(6,1201)
1201        FORMAT(5X,'***TOLERANCE TEST ON THE ADJOINT OF THE RESPONSE***')
            WRITE(6,*)
            DO 1210 IK = 1,IATER
                    IKY = IATER-IK+1
                    TOLTEST = 1.0D0+YASPEC(IKY,IA)
                    IF(TOLTEST.GT.1.0E-3) THEN
                            ITEST = 1
                    ELSE
                            ITEST = 0
                    ENDIF
                    IF(ITEST.NE.ITESTLAST) THEN
                            ITESTLAST=ITEST
                            IF(ITEST.EQ.0) THEN
                                    WRITE(6,661) ATIM(IKY)
                                    TPRIME = ATIM(IKY)
                            ENDIF
                            IF(ITEST.EQ.1) THEN
                            WRITE(6,662) ATIM(IKY),YASPEC(IKY,IA)
                                    TPRIME = ATIM(IKY)-TPRIME
                                    ITPRIME = IK
                            ENDIF
                    ENDIF
1210        CONTINUE
661         FORMAT(5X,' 1 + ADJOINT(T) = 0.; FROM:'E10.3)
662         FORMAT(5X,' 1 + ADJOINT(T) = 0.;   TO:'E10.3,1X,E10.3/5X,'***')
C           DO 1511 LK = 1,NEQ
C           DO 1501 I = 1, IATER
C***************************************************************************
C           REVERSE THE ORDER OF THE ADJOINT STORAGE WITH THE
C           FOLLOWING INDEX, REMEMBER WE CALCULATED BACKWARDS
C***************************************************************************
C           IK = IATER - I + 1
C***************************************************************************
C           put the actual (SCALAR) times in array TALE
C***************************************************************************
C           TALE(I) = ATIM(IK)
C           IF(I.EQ.1) then
C           T = ATIM(IK)
C           ELSE
C           T = DLOG10(ATIM(IK))
C           ENDIF
C***************************************************************************
C           store the actual forward variables in array y
C***************************************************************************
C           Y(NEQ+1) = 1.0D0
C           DO 1500 J = 1,NEQ
C                   Y(J) = ASEVAL(T,J,INC)
C1500       CONTINUE
C           YA = 10.D0**YSPEC(ITER,IA)
C           YJ = 10.D0**YSPEC(ITER,IA)
C           CALL JCSA(NEQ,T,Y,LK,PDJ)
C           DKY(I) = PDJ(IA)
C           DO 1502 K = 1,NEQ
C                   DKY(I) = DKY(I) + YASPEC(IK,K)*PDJ(K)
C1502       CONTINUE
C1501       CONTINUE
C           CADC = 0.0D0
```

```
C          DO 1512 L = 1,IATER-1
C          CADC = CADC +
C      +      (TALE(L+1)-TALE(L))*(DKY(L+1)+DKY(L))/2.D0
C1512     CONTINUE
C          CADC = CADC
C          WRITE(6,1516) ARS(IA),ARS(LK),CADC
C1515     FORMAT(5X,'SENSITIVITY OF ',A8,' TO   ',A8,' WAS < 1.E-3')
C1516     FORMAT(5X,'SENSITIVITY OF ',A8,' TO   ',A8,' IS ',E21.14)
C1511     CONTINUE

          WRITE(6,471)
471       FORMAT(/20X,'TOTAL SENSITIVITY WITH RESPECT TO RATE CONSTANTS')
          WRITE(6,*)
C*************************************************************************
C          INITIAL YA TO THE FINAL CONCENTRATION OF THE SPECIES OF INTEREST
C*************************************************************************
          YA = 10.D0**YSPEC(ITER,IA)
C*************************************************************************
C          OPEN FILE FOR DIAGNOSTICS, IF REQUESTED THROUGH THE NAMELIST
C          OPTION: NORSDIA  (1=NO DIAGNOSTICS, DEFAULT)
C*************************************************************************
          IF(NORSDIA.NE.1) THEN
          OPEN(8,FILE=DIAGFILE, STATUS = 'UNKNOWN')
          WRITE(8,410) IATER
410       FORMAT(1X,I3)
          ENDIF
C*************************************************************************
C          LOOP OVER ALL REACTION RATES, LK
C*************************************************************************
          DO 7000 LK = 1,NRTN
C          ICJP=0
C          PRINT *,'EXAMINE RATE CONSTANT :',LK,'  [1=Y/0=N]'
C          ACCEPT *,ICJP
C          IF(ICJP.NE.1) GO TO 7000
C
C          FIND THE PRODUCTS
C
          IPX = 0
          DO 419 JK = 1,4
419               IPLK(JK) = 0.0
          DO 420 JK = 1,NEQ
          IF(IPX.EQ.4) GO TO 420
                  IF(KOEF(LK,JK).GT.0) THEN
                          IF(KOEF(LK,JK).EQ.1) THEN
                          IPX=IPX+1
                          IPLK(IPX)=JK
                          ENDIF
                          IF(KOEF(LK,JK).EQ.2) THEN
                          IPX = IPX + 1
                          IPLK(IPX) = JK
                          IPX = IPX + 1
                          IPLK(IPX) = JK
                          ENDIF
                  ENDIF
420       CONTINUE
C*************************************************************************
C          SET UP A LOOP TO FILL THE INTEGRAND DO A SIMPLE TRAPAZODIAL
C          INTEGRATION FOR EACH REACTION RATE
C*************************************************************************
          DO 700 I = 1, IATER
```

```
C*********************************************************************
C        REVERSE THE ORDER OF THE ADJOINT STORAGE WITH THE
C        FOLLOWING INDEX, REMEMBER WE CALCULATED BACKWARDS
C*********************************************************************
         IK = IATER - I + 1
C*********************************************************************
C        put the actual (SCALAR) times in array TALE
C*********************************************************************
         TALE(I) = ATIM(IK)
         IF(I.EQ.1) then
         T = ATIM(IK)
         ELSE
         T = DLOG10(ATIM(IK))
         ENDIF
C*********************************************************************
C        store the actual forward variables in array y
C*********************************************************************
         Y(NEQ+1) = 1.0D0
         DO 500 J = 1,NEQ
            Y(J) = ASEVAL(T,J,INC)
500      CONTINUE
         Q = Y(IN1(LK))*Y(IN2(LK))*Y(IN3(LK))
C*********************************************************************
C        EVALUATE dF/da
C        dF(ia)/da = dL/da, lk is the rate constant being evaluated
C*********************************************************************
         DO 505 J = 1,8
               FIA(J,I) = 0.0D0
505      CONTINUE
         FIA(1,I) = DFLOTJ(KOEF(LK,IA))*Q
         IF(IN1(LK).NE.NEQ+1)
     +      FIA(2,I) = -1.0D0*YASPEC(IK,IN1(LK))*Q
         IF(IN2(LK).NE.NEQ+1)
     +      FIA(3,I) = -1.0D0*YASPEC(IK,IN2(LK))*Q
         IF(IN3(LK).NE.NEQ+1)
     +      FIA(4,I) = -1.0D0*YASPEC(IK,IN3(LK))*Q
         IF(IPX.EQ.0) GO TO 700
         DO 510 J = 1, IPX
            FIA(J+4,I) = YASPEC(IK,IPLK(J))*Q
510      CONTINUE
700   CONTINUE
         IF(NORSDIA.NE.1.AND.LK.EQ.1) THEN
         DO 701 I = 1,IATER
            WRITE(8,702) TALE(I).
701      CONTINUE
702      FORMAT(E21.14)
         ENDIF
C*********************************************************************
C        perform the integration
C*********************************************************************
         RESULT2 = 0.0D0
         RESULT1 = 0.0D0
         DO 2113 J = 1,4+IPX
         CAD(J) = 0.0D0
         DO 2112 L = 1,IATER-1
            CAD(J) = CAD(J) +
     +         (TALE(L+1)-TALE(L))*(FIA(J,L+1)+FIA(J,L))/2.D0
C        IF(L+1.EQ.ITPRIME) CMAD(J) = CAD(J)
2112     CONTINUE
         CAD(J) = (CAD(J)/YA)*RC(LK)
```

```
C          CMAD(J) = (CAD(J)-CMAD(J))*RC(LK)/YA
           RESULT2 = RESULT2 + CAD(J)
2113       CONTINUE
C***********************************************************************
C          PRINT DIAGNOSTICS WHEN REQUESTED
C***********************************************************************
           IF(NORSDIA.NE.1) THEN
           WRITE(8,1455) LK
           DO 1450 I2 = 1,IATER
             WRITE(8,1460)   DKY(I2)
1450       CONTINUE
1455       FORMAT(I3)
1460       FORMAT(4E21.14)
           ENDIF
C***********************************************************************
C          PRINT FINAL RESULTS
C***********************************************************************
           IF(DABS(RESULT2).LT.1.D-3) THEN
           WRITE(6,1802) ARS(IA),LK
           GO TO 7000
           ENDIF
1802       FORMAT(/5X,'SENSITIVITY OF ',A8,' TO RXN ',I3,' WAS < 1.E-3')
             WRITE(6,1810)LK,(ARS(ABS(IR(LK,K))),K=1,3),
     +     (ARS(IP(LK,K)),K=1,4),RC(LK)
1810         FORMAT(/1X,I3,1X,3A8,'>',4A8,D9.2,1X,D9.2)
C          IF(DABS(RESULT2).GT.1.5D0) THEN
           WRITE(6,1803)ARS(IA),LK,RESULT2,(J,CAD(J), J=1,4+IPX)
C            WRITE(6,1803)ARS(IA),LK,RESULT2,(J,CAD(J),CMAD(J),J=1,4+IPX)
C          GO TO 7000
C          ENDIF
C          WRITE (6,802) ARS(IA),LK,RESULT2
7000       CONTINUE
1803       FORMAT(5X,'TOTAL SENSITIVITY OF SPECIES : ',A8,/
     +     5x,'WITH RESPECT TO RATE CONSTANT: ',I5,/
     +     5X,'RELATIVE SENSITIVITY ****** : ',E21.14,
     +     /1X,'*THE FOLLOWING WERE SUMMED TO OBTAIN THE SENSITIVITY*',
     +/8(5X,I2,5X,E21.14/))
802        FORMAT(5X,'TOTAL SENSITIVITY OF SPECIES : ',A8,/
     +     5x,'WITH RESPECT TO RATE CONSTANT: ',I5,/
     +     5X,'RELATIVE SENSITIVITY ****** : ',E21.14/)
C***********************************************************************
C          SENSITIVITY WITH RESPECT TO LOW LET G-VALUES
C***********************************************************************
           WRITE(6,8033)
8033       FORMAT(/5X,'TOTAL SENSITIVITY WITH RESPECT TO PRODUCTION RATES')
           DO 900 NK = 1,NEQ
           IF(G(NK).EQ.0.0) GO TO 900
           DO 960 I = 1,IATER
C***********************************************************************
C          REVERSE THE ORDER OF THE ADJOINT STORAGE WITH THE
C          FOLLOWING INDEX, REMEMBER WE CALCULATED BACKWARDS
C***********************************************************************
           IK = IATER - I + 1
C***********************************************************************
C          store the integrands in array DKY(I) WHERE I IS FOR A PARTICULAR
C          TIME STEP.
C***********************************************************************
           DO 960 NGK = 1,NEQ
           IF(NGK.EQ.IA.AND.ATIM(IK).LT.TPRIME) GO TO 960
                 DKY(I)=YASPEC(IK,NGK)
```

```
960       CONTINUE
C******************************************************************
C         PERFORM INTEGRATION
C******************************************************************
C         ADD THE EFFECT OF A NON-ZERO G-VALUE FOR THE RESPONSE VARIABLE
C         WHEN EVALUATING THE RESPONSE DUE TO THIS G-VALUE
C******************************************************************
          CADL = 0.0D0
          DO 2213 L = 1,IATER-1
2213         CADL = CADL + (TALE(L+1)-TALE(L))*(DKY(L+1)+DKY(L))/2.D0
          IF(IA.EQ.NK)
     +              CADL=CADL+(TALE(IATER)-TPRIME)
          RESULT2 = (CADL/YA)*G(NK)*DSRATE
C******************************************************************
C         PRINT DIAGNOSTICS FOR IMPERFECT RUNS
C******************************************************************
          IF(NORSDIA.NE.1) THEN
          WRITE(8,1455) -NK
          DO 1950 I2 = 1,IATER
             WRITE(8,1460)    DKY(I2)
1950      CONTINUE
          ENDIF
C******************************************************************
C         PRINT FINAL RESULTS
C******************************************************************
          IF(DABS(RESULT2).LT.1.D-3) THEN
          WRITE(6,1902) ARS(IA),NK
          GO TO 900
          ENDIF
1902      FORMAT(5X,'SENSITIVITY OF ',A8,' TO G(',I3,')*D WAS < 1.E-3')
          WRITE(6,902)  ARS(IA), NK, CADL, RESULT2
900       CONTINUE
902       FORMAT(5X,'TOTAL SENSITIVITY OF SPECIES : ',A8,/
     +         5X,'WITH RESPECT TO G*D  (FIRST) : ',I5,/
     +         5X,'(dJ/dk) = **************** : ',E21.14,/
     +         5X,'RELATIVE SENSITIVITY ****** : ',E21.14,//)
C******************************************************************
C         HIGH LET G-VALUES
C******************************************************************
       DO 1000 NK = 1,NEQ
       IF(GH(NK).EQ.0.0) GO TO 1000
          DO 1060 I = 1,IATER
C******************************************************************
C         REVERSE THE ORDER OF THE ADJOINT STORAGE
C******************************************************************
          IK = IATER - I + 1
C******************************************************************
C      store the integrands in array DKY(I) WHERE I IS FOR A PARTICULAR
C      TIME STEP.
C******************************************************************
          DKY(I) = YASPEC(IK,NK)
1060      CONTINUE
C******************************************************************
C         PERFORM INTEGRATION
C******************************************************************
          CADH = 0.0D0
          DO 2114 L = 1,IATER-1
2114         CADH = CADH + (TALE(L+1)-TALE(L))*(DKY(L+1)+DKY(L))/2.D0
C******************************************************************
C         ADD THE EFFECT OF A NON-ZERO G-VALUE FOR THE RESPONSE VARIABLE
```

```
C         WHEN EVALUATING THE RESPONSE DUE TO THIS G-VALUE
C*****************************************************************************
          IF(GH(IA).NE.0..AND.IA.EQ.NK)
       +             CADH=CADH+(TALE(IATER)-TPRIME)
          RESULT2 = (CADH/YA)*GH(NK)*DHRATE
C*****************************************************************************
C         PRINT DIAGNOSTICS FOR IMPERFECT RUNS
C*****************************************************************************
          IF(NORSDIA.NE.1) THEN
          WRITE(8,1455) -NK
          DO 1050 I2 = 1,IATER
            WRITE(8,1460)   DKY(I2)
1050      CONTINUE
          ENDIF
C*****************************************************************************
C         PRINT FINAL RESULTS
C*****************************************************************************
          WRITE(6,*)
          WRITE(6,903) ARS(IA), NK,   CADH, RESULT2
          WRITE(6,*)
C
1000      CONTINUE
903    FORMAT(5X,'TOTAL SENSITIVITY OF SPECIES : ',A8,/
       +        5X,'WITH RESPECT TO G*D (SECOND) : ',I5,/
       +        5X,'(dJ/dk) = ***************** : ',E21.14,/
       +        5X,'RELATIVE SENSITIVITY ****** : ',E21.14,//)
C
C         DOSE RATES
C
          IF(DSRATE.EQ.0.0D0) GO TO 9999
          DO 1160 I = 1,IATER
C*****************************************************************************
C         REVERSE THE ORDER OF THE ADJOINT CALCULATION WITH THE
C         FOLLOWING INDEX
C*****************************************************************************
          IK = IATER - I + 1
C*****************************************************************************
C         store the integrands in array DKY(I) WHERE I IS FOR A PARTICULAR
C         TIME STEP.
C*****************************************************************************
          TOLTEST=1.D0+YASPEC(IK,IA)
          DKY(I)=0.0
          IF(TOLTEST.GT.RTOL) DKY(I)=TOLTEST*G(IA)
       DO 1160 K = 1,NEQ
          IF(K.EQ.IA) GO TO 1160
             DKY(I) = DKY(I) + G(K)*YASPEC(IK,K)
1160      CONTINUE
C*****************************************************************************
C         PERFORM INTEGRATION
C*****************************************************************************
          CADD = 0.0D0
          DO 2115 L = 1,IATER-1
2115         CADD = CADD + (TALE(L+1)-TALE(L))*(DKY(L+1)+DKY(L))/2.D0
          RESULT2 = (CADD/YA)*DSRATE
C*****************************************************************************
C         PRINT DIAGNOSTICS FOR IMPERFECT RUNS
C*****************************************************************************
          IF(NORSDIA.NE.1) THEN
          WRITE(8,1455) 0
          DO 1150 I2 = 1,IATER
```

```
            WRITE(8,1460)   DKY(I2)
1150        CONTINUE
            ENDIF
C***************************************************************************
C       PRINT FINAL RESULTS
C***************************************************************************
        WRITE(6,*)
        WRITE(6,904)  ARS(IA),   CADD, RESULT2
        WRITE(6,*)
C
904     FORMAT(//5X,'TOTAL SENSITIVITY OF SPECIES : ',A8,/
       +        5x,'WITH RESPECT TO DOSE RATES   ',/
       +        5X,'(dJ/dk LO) = ************** : ',E21.14,/
       +        5X,'RELATIVE SENSITIVITY(LO) *** : ',E21.14,//)
9999    CLOSE (6)
        STOP
        END

        DOUBLE PRECISION FUNCTION ASEVAL(U,IS,INC)
C***************************************************************************
c  evaluate the forward values of the functions for time 'u'
C***************************************************************************
        INCLUDE 'PARAMETER.blk'
        common /adval/ yi(id2),x,q,n,b,c,d, IATER
C       COMMON IL
        INCLUDE 'DIMENSION.BLK'
        dimension x(id2),q(id2,id2),b(id2,id2),c(id2,id2),d(id2,id2)
C***************************************************************************
c       this subroutine evaluates the cubic spline function
c
c    seval = y(i) + b(i)*(u-x(i)) + c(i)*(u-x(i))**2 + d(i)*(u-x(i))**3
c
c    where x(i).lt. u .lt. x(i+1), using horner's rule
c
c    if u .lt. x(1) then i = 1 is used
c    if u.ge. x(n) then i = n is used
c
c  input
c       n = the number of data points
c       u = the abscissa at which the spline is to be evaluated
c       x, y = the arrays of data abscissas and ordinates
c       b,c,d = arrays of spline coefficients computed by spline
c
c  A binary search is performed to determine the proper interval.
C***************************************************************************
C       data I/1/
        if (u.le.0.d0) go to 40
C***************************************************************************
c  binary search, USUALLY SEQUENTIAL BUT SLIP BACK ONE INTERVAL ANYWAY
C***************************************************************************
        IF (IS.GT.1) THEN
        I = INC
        GO TO 30
        ENDIF
 10     i = 1
        j = n+1
 20     k = (i+j)/2
        if(u.lt.x(k)) j = k
        if(u.ge.x(k)) i = k
        if(j.gt.i+1) go to 20
```

```
        INC = I
C****************************************************************************
c  evaluate spline
C****************************************************************************
 30        CONTINUE
           dx = u - x(i)
C          aseval = q(i,is) + dx*(b(i,is) + dx*(c(i,is) + dx*d(i,is)))
           IF(I.EQ.ITER) THEN
           ASEVAL = Q(I,IS)
           GO TO 35
           ENDIF
           ASEVAL = Q(I,IS) + DX*((Q(I+1,IS)-Q(I,IS))/(X(I+1)-X(I)))
 35        aseval = 10.d0**aseval
           return
 40        Aseval = yi(is)
           return
           end


        SUBROUTINE JCSA (NEQ,T,Y,J,PDJ)
C****************************************************************************
C        VERSION: MIT 2.0
C        CODE CUSTODIAN: S. A. SIMONSON
C        DATE OF LAST CHANGE TO SUBROUTINE: 11/17/86
C        JAC IS PART OF THE SOLADJ CODE
C        JAC IS CALLED BY THE SUBROUTINE ADFUN
C****************************************************************************
C        JAC CALCULATES THE COLUMN VECTORS FOR THE JACOBIAN MATRIX
C****************************************************************************
        include 'PARAMETER.blk'
        INCLUDE 'DIMENSION.BLK'
        dimension  PDJ(ID2), Y(ID2)
        DO 100 I=1,NEQ
        IF(I.EQ.J.AND.BCL(I).NE.2.0D0.AND.MESHPTS.NE.0) THEN
        PDJ(I) = -2.D0*DIF(I)
        ELSE
        PDJ(I) = 0.0D0
        ENDIF
           DO 100 K=1,NRTN
           IF(KOEF(K,J).EQ.0.OR.KOEF(K,I).EQ.0) GOTO 100
           A = RC(K)*DFLOTJ(KOEF(K,I)*JIABS(KOEF(K,J)))
           IM = IN1(K)
           IN = IN2(K)
           IO = IN3(K)
C****************************************************************************
C        CATCH SECOND ORDER REACTIONS
C****************************************************************************
           IF((IM.EQ.IN.OR.IO.EQ.IN).AND.(IN.EQ.J))THEN
           PDJ(I) = PDJ(I)+A*Y(IM)*Y(IO)
           GO TO 100
           ENDIF
           IF (IM.EQ.J) THEN
           PDJ(I) = PDJ(I)+A*Y(IN)*Y(IO)
           GOTO 100
           ENDIF
C****************************************************************************
C        FIRST ORDER REACTIONS
C****************************************************************************
           IF (IO.EQ.J) THEN
           PDJ(I) = PDJ(I)+A*Y(IN)*Y(IM)
           GOTO 100
```

```
              ENDIF
            ·IF (IN.EQ.J) PDJ(I) = PDJ(I)+A*Y(IM)*Y(IO)
      100   CONTINUE
            RETURN
            END

            SUBROUTINE READIN(NEQ,Y)
      C
      C*********************************************************************
      C         VERSION: MIT 1.0
      C         CODE CUSTODIAN: S. A. SIMONSON
      C         DATE OF LAST CHANGE OF SUBROUTINE: 1/9/87
      C         PART OF THE SPATIAL RADIOLYSIS CODE PACKAGE
      C         THIS SUBROUTINE IS CALLED BY THE SUBROUTINE DRIVE
      C*********************************************************************
      C         INPUT READS LOGICAL UNIT NUMBER 5 FOR THE REACTION MATRIX
      C         AND REACTION RATE CONSTANTS.  REACTION RATE CONSTANTS ARE
      C         ADJUSTED FOR TEMPERATURE USING AN ARRHENIUS TEMPERATURE
      C         DEPENDENCE
      C*********************************************************************
      C
            INCLUDE 'PARAMETER.BLK'
              INCLUDE 'DIMENSION.BLK'
            CHARACTER*3 CJP
            COMMON/PASSER/IR,IP,EA
          DIMENSION  EA(ID2), Y(ID2), IP(ID1,4), IR(ID1,3)
          DATA R/8.314D-3/, RU/0.08206D0/, IFLAG/0/
          ARS(0)='        '
      C
      C*********************************************************************
      C         READ THE REACTIONS ONE BY ONE AND SET UP THE COEFFICIENT
      C         MATRICIES (KOEF), AND THE REACTION ORDER MATRIX (NJ)
      C*********************************************************************
      C
      C       WRITE (6,10)
      10      FORMAT (//10X,
          +59HCHEMICAL REACTIONS, RATE CONSTANTS, AND ACTIVATION ENERGIES//
          +26X,'REACTIONS'28X,'RATE',3X,'ACTIVATION',/62X,'CONSTANT',2X,
          +'ENERGIES')
              IFLAG=0
              DO 140 I=1,NRTN
              READ(5,100)CJP,(IR(I,K),K=1,3),(IP(I,K),K=1,4),RC(I),EA(I)
      C
      C*********************************************************************
      C         IF THE FLAG INDICATES (IFLAG=1) THAT A PHASE EQUILIBRIUM
      C         REACTION WAS DETECTED FOR THE LAST REACTION
      C         (RC(I - 1) = -1), CALCULATE THE REACTION RATE USING A
      C         HENRY'S LAW APPROACH FOR REACTION (I) GIVEN THAT
      C         RC(I - 1) = 100.
      C*********************************************************************
      C
            IF(IFLAG.EQ.0.AND.RC(I).GE.0.0) THEN
            RC(I)=(RC(I)/DEXP(-EA(I)/(R*TEMR)))*DEXP(-EA(I)/(R*TEM))
            ENDIF
            IF(IFLAG.GE.1) THEN
            IF(IFLAG.EQ.1) RC(I)=1000.D3*VL/(VL+VG*(RC(I)/(RU*TEM)))
            IF(IFLAG.EQ.2) RC(I) = 10.D0**(10.0D0-RC(I))
             IFLAG=0
            ENDIF
      C
```

```
C*****************************************************************
C      CONVERT THE PATE CONSTANT USING AN ARRHENIUS EXPERSSION
C*****************************************************************
C
C
C*****************************************************************
C      SET THE FIRST RATE CONSTANT OF A PHASE PARTITIONING TO 100.
C      (THIS SHOULD BE A LIQUID TO GAS PHASE REACTION FOLLOWED BY
C      A GAS TO LIQUID PHASE SO THAT THE PROGRAM PROPERLY ACCOUNTS
C      FOR THE PARTITIONING)
C*****************************************************************
C
         IF(RC(I).EQ.-1.0) THEN
         RC(I)=1000.D3
         IFLAG=1
         ENDIF
              IF(RC(I).EQ.-2.0) THEN
              RC(I) = 1.0D10
              IFLAG = 2
              ENDIF
C        WRITE(6,110)CJP,(ARS(JIABS(IR(I,K))),K=1,3),
C      +  (ARS(IP(I,K)),K=1,4),RC(I),EA(I)
100      FORMAT(A3,3X,7I3,D18.8/D18.8)
110      FORMAT(1X,A3,1X,3A8,'>',4A8,D9.2,1X,D9.2)
C
C*****************************************************************
C      SET UP THE REACTION INDICIES IR AND THE PRODUCT INDICIES IP
C      FILL THE COEFFICIENT MATRIX KOEF
C
C      CHECK FIRST TO SEE IF ANY OF THE REACTANTS ARE PRESENT
C      IN A SECOND ORDER FASHION
C*****************************************************************
C
         IF(((IR(I,1).EQ.IR(I,2)).OR.(IR(I,2).EQ.IR(I,3)))
      +  .AND.(IR(I,2).NE.0))THEN
         NJ(I,JIABS(IR(I,2)))=-2
         KOEF(I,JIABS(IR(I,2)))=-2
         ENDIF
C
C*****************************************************************
C      FILL UP THE MATRIX FOR THE REACTANTS WHICH ARE FIRST ORDER
C*****************************************************************
C
         DO 120 K=1,3
         IF((IR(I,K).NE.0).AND.(NJ(I,JIABS(IR(I,K))).NE.-2))THEN
         NJ(I,JIABS(IR(I,K)))=-1
         KOEF(I,JIABS(IR(I,K)))=-1
         ENDIF
120      CONTINUE
C
C*****************************************************************
C      CHECK FOR SECOND ORDER PRODUCTS
C*****************************************************************
C
         IF(((IP(I,1).EQ.IP(I,2)).OR.(IP(I,2).EQ.IP(I,3)))
      +  .AND.(IP(I,2).NE.0))THEN
         KOEF(I,JIABS(IP(I,2)))=2
         ENDIF
         IF(((IP(I,2).EQ.IP(I,3)).OR.(IP(I,3).EQ.IP(I,4)))
      +  .AND.(IP(I,3).NE.0))THEN
```

```
            KOEF(I,JIABS(IP(I,3)))=2
            ENDIF
C
C**********************************************************************
C         FILL UP THE PRODUCTS MATRIX FOR FIRST ORDER PRODUCTS
C**********************************************************************
C
            DO 130 K=1,4
            IF((IP(I,K).NE.0).AND.(KOEF(I,IP(I,K)).NE.2))THEN
            KOEF(I,IP(I,K))=1
            ENDIF
130         CONTINUE
140      CONTINUE
C
C**********************************************************************
C         NORMALIZE CATALYTIC REACTANTS
C**********************************************************************
C
            DO 150 K=1,NEQ
            DO 150 I=1,NRTN
C
C**********************************************************************
C         ARE THERE PRODUCTS OF SPECIES K AS WELL AS REACTANTS OF
C         SPECIES K?
C**********************************************************************
C
            IF((KOEF(I,K).NE.NJ(I,K)).AND.(NJ(I,K).NE.0))
     +      KOEF(I,K)=KOEF(I,K)+NJ(I,K)
C
C**********************************************************************
C         ARE THERE ONLY PRODUCTS? (FILL NJ AFTER CHECKING FOR
C         CATALYTIC REACTIONS)
C**********************************************************************
C
            IF((KOEF(I,K).NE.NJ(I,K)).AND.(NJ(I,K).EQ.0))
     +      NJ(I,K)=KOEF(I,K)
150         CONTINUE
C
C**********************************************************************
C         READ INITIAL VALUES, G-VALUES, RELATIVE TOLERENCES, AND
C         CONVERT G-VALUES FROM # spec/100 ev  TO  moles/1-rad
C**********************************************************************
C
      READ (5,160) DUM
      READ (5,160) DUM
160   FORMAT (A1)
C     WRITE (6,190)
190   FORMAT(//18X,7HLOW LET,4X,8HHIGH LET,3X,5HLOWER,3X,
     +           5HUPPER,3X,9HDIFFUSION,
     + /         14X,3X,8HG-VALUES,3X,8HG-VALUES,3X,5HBOUND,3X,
     +           5HBOUND,2X,11HCOEFFICIENT)
      DO 200 I=1,NEQ
      READ (5,220)  G(I), GH(I), BCL(I), BCR(I), DIF(I)
C     WRITE (6,210) ARS(I), G(I), GH(I), BCL(I), BCR(I),
C     +             DIF(I)
      IF(MESHPTS.GT.0) DIF(I) = DIF(I)/XINCSQ
      GH(I)= GH(I)*1.033D-9
200   G(I) = G(I)*1.033D-9
210   FORMAT (/1X,A8,5(2X,D9.2))
220   FORMAT(8X,D10.3,4(/8X,D10.3))
```

```
            CALL YINITIAL(Y)
C          DO 240 K=1,NEQ
C          WRITE(6,230) ARS(K), Y(K)
C230    FORMAT(1X,A8,5(2X,D9.2)/11X)
C240    CONTINUE
        CLOSE (5).
        RETURN
        END


        SUBROUTINE YINITIAL(Y)
C
C
C*********************************************************************
C       VERSION: MIT 1.0
C       CODE CUSTODIAN: S. A. SIMONSON
C       DATE OF LAST CHANGE OF SUBROUTINE: 7/27/88
C       PART OF THE SPATIAL RADIOLYSIS CODE PACKAGE
C       THIS SUBROUTINE IS CALLED BY THE SUBROUTINE READIN
C*********************************************************************
C       YINITIAL SETS THE INITIAL VALUES OF ARRAY Y AT NODES 1 TO
C       MESHPTS
C
C
C*********************************************************************
C
C       ASSIGN INITIAL VALUES TO ARRAY Y
C
        INCLUDE 'PARAMETER.BLK'
        INCLUDE 'DIMENSION.BLK'
        NAMELIST /VALUES/ YI
        DIMENSION Y(ID2),YI(ID2)
C
C       SET-UP FOR HINDMARSH PROBLEM
C
C       DO 30 I = 1, MESHPTS
C               Z = 30.D0 + DFLOAT(I-1)*XINC
C               Z1 = 0.1D0*Z - 4.D0
C               Z1 = Z1**2
C               GAMZ = 1. - Z1 + 0.5D0*Z1**2
C               Y(2*I-1) = 1.0D6*GAMZ
C               Y(2*I) = 1.D12*GAMZ
C30     CONTINUE
        READ(5,NML=VALUES)
        DO 10 I=1,ID2
        Y(I)=YI(I)
10      CONTINUE
        RETURN
        END
```

APPENDIX C
LSODE WRITEUP

```
C-------------------------------------------------------------------
C THIS IS THE AUGUST 13, 1981 VERSION OF
C LSODE.. LIVERMORE SOLVER FOR ORDINARY DIFFERENTIAL EQUATIONS.
C THIS VERSION IS IN SINGLE PRECISION.
C
C LSODE SOLVES THE INITIAL VALUE PROBLEM FOR STIFF OR NONSTIFF
C SYSTEMS OF FIRST ORDER ODE-S,
C      DY/DT = F(T,Y) ,   OR, IN COMPONENT FORM,
C      DY(I)/DT = F(I) = F(I,T,Y(1),Y(2),...,Y(NEQ)) (I = 1,...,NEQ).
C LSODE IS A PACKAGE BASED ON THE GEAR AND GEARB PACKAGES, AND ON THE
C OCTOBER 23, 1978 VERSION OF THE TENTATIVE ODEPACK USER INTERFACE
C STANDARD, WITH MINOR MODIFICATIONS.
C-------------------------------------------------------------------
C REFERENCE..
C      ALAN C. HINDMARSH,  LSODE AND LSODI, TWO NEW INITIAL VALUE
C      ORDINARY DIFFERENTIAL EQUATION SOLVERS,
C      ACM-SIGNUM NEWSLETTER, VOL. 15, NO. 4 (1980), PP. 10-11.
C-------------------------------------------------------------------
C AUTHOR AND CONTACT.. ALAN C. HINDMARSH,
C                      MATHEMATICS AND STATISTICS DIVISION, L-316
C                      LAWRENCE LIVERMORE NATIONAL LABORATORY
C                      LIVERMORE, CA 94550.
C-------------------------------------------------------------------
C SUMMARY OF USAGE.
C
C COMMUNICATION BETWEEN THE USER AND THE LSODE 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
C B. NEXT DETERMINE (OR GUESS) WHETHER OR NOT THE PROBLEM IS STIFF.
C STIFFNESS OCCURS WHEN THE JACOBIAN MATRIX DF/DY HAS AN EIGENVALUE
C WHOSE REAL PART IS NEGATIVE AND LARGE IN MAGNITUDE, COMPARED TO THE
C RECIPROCAL OF THE T SPAN OF INTEREST.  IF THE PROBLEM IS NONSTIFF,
C USE A METHOD FLAG MF = 10.  IF IT IS STIFF, THERE ARE FOUR STANDARD
C CHOICES FOR MF, AND LSODE REQUIRES THE JACOBIAN 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, LSODE REQUIRES TWO
C HALF-BANDWIDTH PARAMETERS ML AND MU.  THESE ARE, RESPECTIVELY, THE
C 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
C I-ML .LE. J .LE. I+MU, AND THE FULL BANDWIDTH IS ML+MU+1.
C
C C. IF THE PROBLEM IS STIFF, YOU ARE ENCOURAGED TO SUPPLY THE JACOBIAN
C DIRECTLY (MF = 21 OR 24), BUT IF THIS IS NOT FEASIBLE, LSODE WILL
C COMPUTE IT INTERNALLY BY DIFFERENCE QUOTIENTS (MF = 22 OR 25).
C IF YOU ARE SUPPLYING THE JACOBIAN, PROVIDE A SUBROUTINE OF THE FORM..
C                SUBROUTINE JAC (NEQ, T, Y, ML, MU, PD, NROWPD)
C                DIMENSION Y(NEQ), PD(NROWPD,NEQ)
C WHICH SUPPLIES DF/DY BY LOADING PD AS FOLLOWS..
C      FOR A FULL JACOBIAN (MF = 21), LOAD PD(I,J) WITH DF(I)/DY(J),
C THE PARTIAL DERIVATIVE OF F(I) WITH RESPECT TO Y(J).  (IGNORE THE
C ML AND MU ARGUMENTS IN THIS CASE.)
```

```
C       FOR A BANDED JACOBIAN (MF = 24), LOAD PD(I-J+MU+1,J) WITH
C DF(I)/DY(J), I.E. LOAD THE DIAGONAL LINES OF DF/DY INTO THE ROWS OF
C PD FROM THE TOP DOWN.
C       IN EITHER CASE, ONLY NONZERO ELEMENTS NEED BE LOADED.
C
C D. WRITE A MAIN PROGRAM WHICH CALLS SUBROUTINE LSODE ONCE FOR
C 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 LSODE.  ON THE FIRST CALL TO LSODE, 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 (.NE. T).
C ITOL    = 1 OR 2 ACCORDING AS ATOL (BELOW) IS A SCALAR OR ARRAY.
C RTOL    = RELATIVE TOLERANCE PARAMETER (SCALAR).
C ATOL    = ABSOLUTE TOLERANCE PARAMETER (SCALAR OR ARRAY).
C           THE ESTIMATED LOCAL ERROR IN Y(I) WILL BE CONTROLLED SO AS
C           TO BE ROUGHLY LESS (IN MAGNITUDE) THAN
C              EWT(I) = RTOL*ABS(Y(I)) + ATOL      IF ITOL = 1, OR
C              EWT(I) = RTOL*ABS(Y(I)) + ATOL(I)   IF ITOL = 2.
C           THUS THE LOCAL ERROR TEST PASSES IF, IN EACH COMPONENT,
C           EITHER THE ABSOLUTE ERROR IS LESS THAN ATOL (OR ATOL(I)),
C           OR THE RELATIVE ERROR IS LESS THAN RTOL.
C           USE RTOL = 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
C           LOCAL TOLERANCES, SO CHOOSE THEM CONSERVATIVELY.
C ITASK   = 1 FOR NORMAL COMPUTATION OF OUTPUT VALUES OF Y AT T = TOUT.
C ISTATE  = INTEGER FLAG (INPUT AND OUTPUT).  SET ISTATE = 1.
C IOPT    = 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 LRW     = DECLARED LENGTH OF RWORK (IN USER-S DIMENSION).
C IWORK   = INTEGER WORK ARRAY OF LENGTH AT LEAST..
C              20         FOR MF = 10,
C              20 + NEQ   FOR MF = 21, 22, 24, OR 25.
C           IF MF = 24 OR 25, INPUT IN IWORK(1),IWORK(2) THE LOWER
C           AND UPPER HALF-BANDWIDTHS ML,MU.
C LIW     = DECLARED LENGTH OF IWORK (IN USER-S DIMENSION).
C JAC     = NAME OF SUBROUTINE FOR JACOBIAN MATRIX (MF = 21 OR 24).
C           IF USED, THIS NAME MUST BE DECLARED EXTERNAL IN CALLING
C           PROGRAM.  IF NOT USED, PASS A DUMMY NAME.
C MF      = METHOD FLAG.  STANDARD VALUES ARE..
C           10 FOR NONSTIFF (ADAMS) METHOD, NO JACOBIAN USED.
C           21 FOR STIFF (BDF) METHOD, USER-SUPPLIED FULL JACOBIAN.
C           22 FOR STIFF METHOD, INTERNALLY GENERATED FULL JACOBIAN.
C           24 FOR STIFF METHOD, USER-SUPPLIED BANDED JACOBIAN.
C           25 FOR STIFF METHOD, INTERNALLY GENERATED BANDED JACOBIAN.
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      Y = ARRAY OF COMPUTED VALUES OF Y(T) VECTOR.
C      T = CORRESPONDING VALUE OF INDEPENDENT VARIABLE (NORMALLY TOUT).
C ISTATE = 2  IF LSODE WAS SUCCESSFUL, NEGATIVE OTHERWISE.
C           -1 MEANS EXCESS WORK DONE ON THIS CALL (PERHAPS WRONG MF).
```

```
C              -2 MEANS EXCESS ACCURACY REQUESTED (TOLERANCES TOO SMALL).
C              -3 MEANS ILLEGAL INPUT DETECTED (SEE PRINTED MESSAGE).
C              -4 MEANS REPEATED ERROR TEST FAILURES (CHECK ALL INPUTS).
C              -5 MEANS REPEATED CONVERGENCE FAILURES (PERHAPS BAD JACOBIAN
C                 SUPPLIED OR WRONG CHOICE OF MF OR TOLERANCES).
C              -6 MEANS ERROR WEIGHT BECAME ZERO DURING PROBLEM. (SOLUTION
C                 COMPONENT I VANISHED, AND ATOL OR ATOL(I) = 0.)
C
C F. TO CONTINUE THE INTEGRATION AFTER A SUCCESSFUL RETURN, SIMPLY
C RESET TOUT AND CALL LSODE AGAIN.  NO OTHER PARAMETERS NEED BE RESET.
C
C------------------------------------------------------------------------
C EXAMPLE PROBLEM.
C
C THE FOLLOWING IS A SIMPLE EXAMPLE PROBLEM, WITH THE CODING
C NEEDED FOR ITS SOLUTION BY LSODE.  THE PROBLEM IS FROM CHEMICAL
C KINETICS, AND CONSISTS OF THE FOLLOWING THREE RATE EQUATIONS..
C      DY1/DT = -.04*Y1 + 1.E4*Y2*Y3
C      DY2/DT = .04*Y1 - 1.E4*Y2*Y3 - 3.E7*Y2**2
C      DY3/DT = 3.E7*Y2**2
C ON THE INTERVAL FROM T = 0.0 TO T = 4.E10, WITH INITIAL CONDITIONS
C Y1 = 1.0, Y2 = Y3 = 0.  THE PROBLEM IS STIFF.
C
C THE FOLLOWING CODING SOLVES THIS PROBLEM WITH LSODE, USING MF = 21
C AND PRINTING RESULTS AT T = .4, 4., ..., 4.E10.  IT USES
C ITOL = 2 AND ATOL MUCH SMALLER FOR Y2 THAN 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
C      EXTERNAL FEX, JEX
C      DIMENSION Y(3), ATOL(3), RWORK(58), IWORK(23)
C      NEQ = 3
C      Y(1) = 1.
C      Y(2) = 0.
C      Y(3) = 0.
C      T = 0.
C      TOUT = .4
C      ITOL = 2
C      RTOL = 1.E-4
C      ATOL(1) = 1.E-6
C      ATOL(2) = 1.E-10
C      ATOL(3) = 1.E-6
C      ITASK = 1
C      ISTATE = 1
C      IOPT = 0
C      LRW = 58
C      LIW = 23
C      MF = 21
C      DO 40 IOUT = 1,12
C        CALL LSODE(FEX,NEQ,Y,T,TOUT,ITOL,RTOL,ATOL,ITASK,ISTATE,
C       1   IOPT,RWORK,LRW,IWORK,LIW,JEX,MF)
C        WRITE(6,20)T,Y(1),Y(2),Y(3)
C 20     FORMAT(7H AT T =,E12.4,6H   Y =,3E14.6)
C        IF (ISTATE .LT. 0) GO TO 80
C 40     TOUT = TOUT*10.
C      WRITE(6,60)IWORK(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(6,90)ISTATE
```

```
C 90   FORMAT(///22H ERROR HALT.. ISTATE =,I3)
C      STOP
C      END
C
C      SUBROUTINE FEX (NEQ, T, Y, YDOT)
C      DIMENSION Y(3), YDOT(3)
C      YDOT(1) = -.04*Y(1) + 1.E4*Y(2)*Y(3)
C      YDOT(3) = 3.E7*Y(2)*Y(2)
C      YDOT(2) = -YDOT(1) - YDOT(3)
C      RETURN
C      END
C
C      SUBROUTINE JEX (NEQ, T, Y, ML, MU, PD, NRPD)
C      DIMENSION Y(3), PD(NRPD,3)
C      PD(1,1) = -.04
C      PD(1,2) = 1.E4*Y(3)
C      PD(1,3) = 1.E4*Y(2)
C      PD(2,1) = .04
C      PD(2,3) = -PD(1,3)
C      PD(3,2) = 6.E7*Y(2)
C      PD(2,2) = -PD(1,2) - PD(3,2)
C      RETURN
C      END
C
C THE OUTPUT OF THIS PROGRAM (ON A CDC-7600 IN SINGLE PRECISION)
C IS AS FOLLOWS..
C
C   AT T = 4.0000E-01   Y =  9.851726E-01  3.386406E-05  1.479357E-02
C   AT T = 4.0000E+00   Y =  9.055142E-01  2.240418E-05  9.446344E-02
C   AT T = 4.0000E+01   Y =  7.158050E-01  9.184616E-06  2.841858E-01
C   AT T = 4.0000E+02   Y =  4.504846E-01  3.222434E-06  5.495122E-01
C   AT T = 4.0000E+03   Y =  1.831701E-01  8.940379E-07  8.168290E-01
C   AT T = 4.0000E+04   Y =  3.897016E-02  1.621193E-07  9.610297E-01
C   AT T = 4.0000E+05   Y =  4.935213E-03  1.983756E-08  9.950648E-01
C   AT T = 4.0000E+06   Y =  5.159269E-04  2.064759E-09  9.994841E-01
C   AT T = 4.0000E+07   Y =  5.306413E-05  2.122677E-10  9.999469E-01
C   AT T = 4.0000E+08   Y =  5.494529E-06  2.197824E-11  9.999945E-01
C   AT T = 4.0000E+09   Y =  5.129458E-07  2.051784E-12  9.999995E-01
C   AT T = 4.0000E+10   Y = -7.170586E-08 -2.868234E-13  1.000000E+00
C
C   NO. STEPS = 330   NO. F-S = 405   NO. J-S =  69
C-----------------------------------------------------------------------
C FULL DESCRIPTION OF USER INTERFACE TO LSODE.
C
C THE USER INTERFACE TO LSODE CONSISTS OF THE FOLLOWING PARTS.
C
C I.    THE CALL SEQUENCE TO SUBROUTINE LSODE, WHICH IS A DRIVER
C       ROUTINE FOR THE SOLVER.  THIS INCLUDES DESCRIPTIONS OF BOTH
C       THE CALL SEQUENCE ARGUMENTS AND OF USER-SUPPLIED ROUTINES.
C       FOLLOWING THESE DESCRIPTIONS IS A DESCRIPTION OF
C       OPTIONAL INPUTS AVAILABLE THROUGH THE CALL SEQUENCE, AND THEN
C       A DESCRIPTION OF OPTIONAL OUTPUTS (IN THE WORK ARRAYS).
C
C II.   DESCRIPTIONS OF OTHER ROUTINES IN THE LSODE PACKAGE THAT MAY BE
C       (OPTIONALLY) CALLED BY THE USER.  THESE PROVIDE THE ABILITY TO
C       ALTER ERROR MESSAGE HANDLING, SAVE AND RESTORE THE INTERNAL
C       COMMON, AND OBTAIN SPECIFIED DERIVATIVES OF THE SOLUTION Y(T).
C
C III.  DESCRIPTIONS OF COMMON BLOCKS TO BE DECLARED IN OVERLAY
C       OR SIMILAR ENVIRONMENTS, OR TO BE SAVED WHEN DOING AN INTERRUPT
```

```
C        OF THE PROBLEM AND CONTINUED SOLUTION LATER.
C
C IV.  DESCRIPTION OF TWO SUBROUTINES IN THE LSODE PACKAGE, EITHER OF
C      WHICH THE USER MAY REPLACE WITH HIS OWN VERSION, IF DESIRED.
C      THESE RELATE TO THE MEASUREMENT OF ERRORS.
C
C-----------------------------------------------------------------------
C PART I.  CALL SEQUENCE.
C
C THE CALL SEQUENCE PARAMETERS USED FOR INPUT ONLY ARE
C      F, NEQ, TOUT, ITOL, RTOL, ATOL, ITASK, IOPT, LRW, LIW, JAC, MF,
C AND THOSE USED FOR BOTH INPUT AND OUTPUT ARE
C      Y, T, ISTATE.
C THE WORK ARRAYS RWORK AND IWORK ARE ALSO USED FOR CONDITIONAL AND
C OPTIONAL INPUTS AND OPTIONAL OUTPUTS.  (THE TERM OUTPUT HERE REFERS
C TO THE RETURN FROM SUBROUTINE LSODE TO THE USER-S CALLING PROGRAM.)
C
C THE LEGALITY OF INPUT PARAMETERS WILL BE THOROUGHLY CHECKED ON THE
C INITIAL CALL FOR THE PROBLEM, BUT NOT CHECKED THEREAFTER UNLESS A
C CHANGE IN INPUT PARAMETERS IS FLAGGED BY ISTATE = 3 ON INPUT.
C
C THE DESCRIPTIONS OF THE CALL ARGUMENTS ARE AS FOLLOWS.
C
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
C          SUBROUTINE F MAY ACCESS USER-DEFINED QUANTITIES IN
C          NEQ(2),... AND Y(NEQ(1)+1),... IF NEQ IS AN ARRAY
C          (DIMENSIONED IN F) AND Y HAS LENGTH EXCEEDING NEQ(1).
C          SEE THE DESCRIPTIONS OF NEQ AND Y BELOW.
C
C NEQ    = THE SIZE OF THE ODE SYSTEM (NUMBER OF FIRST ORDER
C          ORDINARY DIFFERENTIAL EQUATIONS).  USED ONLY FOR INPUT.
C          NEQ MAY BE DECREASED, BUT NOT INCREASED, DURING THE PROBLEM.
C          IF NEQ IS DECREASED (WITH ISTATE = 3 ON INPUT), THE
C          REMAINING COMPONENTS OF Y SHOULD BE LEFT UNDISTURBED, IF
C          THESE ARE TO BE ACCESSED IN F AND/OR JAC.
C
C          NORMALLY, NEQ IS A SCALAR, AND IT IS GENERALLY REFERRED TO
C          AS A SCALAR IN THIS USER INTERFACE DESCRIPTION.  HOWEVER,
C          NEQ MAY BE AN ARRAY, WITH NEQ(1) SET TO THE SYSTEM SIZE.
C          (THE LSODE PACKAGE ACCESSES ONLY NEQ(1).)  IN EITHER CASE,
C          THIS PARAMETER IS PASSED AS THE NEQ ARGUMENT IN ALL CALLS
C          TO F AND JAC.  HENCE, IF IT IS AN ARRAY, LOCATIONS
C          NEQ(2),... MAY BE USED TO STORE OTHER INTEGER DATA AND PASS
C          IT TO F AND/OR JAC.  SUBROUTINES F AND/OR JAC MUST INCLUDE
C          NEQ IN A DIMENSION STATEMENT IN THAT CASE.
C
C Y      = A REAL ARRAY FOR THE VECTOR OF DEPENDENT VARIABLES, OF
```

```
C         LENGTH NEQ OR MORE.  USED FOR BOTH INPUT AND OUTPUT ON THE
C         FIRST CALL (ISTATE = 1), AND ONLY FOR OUTPUT ON OTHER CALLS.
C         ON THE FIRST CALL, Y MUST CONTAIN THE VECTOR OF INITIAL
C         VALUES.  ON OUTPUT, Y CONTAINS THE COMPUTED SOLUTION VECTOR,
C         EVALUATED AT T.  IF DESIRED, THE Y ARRAY MAY BE USED
C         FOR OTHER PURPOSES BETWEEN CALLS TO THE SOLVER.
C
C         THIS ARRAY IS PASSED AS THE Y ARGUMENT IN ALL CALLS TO
C         F AND JAC.  HENCE ITS LENGTH MAY EXCEED NEQ, AND LOCATIONS
C         Y(NEQ+1),... MAY BE USED TO STORE OTHER REAL DATA AND
C         PASS IT TO F AND/OR JAC.  (THE LSODE PACKAGE ACCESSES ONLY
C         Y(1),...,Y(NEQ).)
C
C T     = THE INDEPENDENT VARIABLE.  ON INPUT, T IS USED ONLY ON THE
C         FIRST CALL, AS THE INITIAL POINT OF THE INTEGRATION.
C         ON OUTPUT, AFTER EACH CALL, T IS THE VALUE AT WHICH A
C         COMPUTED SOLUTION Y IS EVALUATED (USUALLY THE SAME AS TOUT).
C         ON AN ERROR RETURN, T IS THE FARTHEST POINT REACHED.
C
C TOUT  = THE NEXT VALUE OF T AT WHICH A COMPUTED SOLUTION IS DESIRED.
C         USED ONLY FOR INPUT.
C
C         WHEN STARTING THE PROBLEM (ISTATE = 1), TOUT MAY BE EQUAL
C         TO T FOR ONE CALL, THEN SHOULD .NE. T FOR THE NEXT CALL.
C         FOR THE INITIAL T, AN INPUT VALUE OF TOUT .NE. T IS USED
C         IN ORDER TO DETERMINE THE DIRECTION OF THE INTEGRATION
C         (I.E. THE ALGEBRAIC SIGN OF THE STEP SIZES) AND THE ROUGH
C         SCALE OF THE PROBLEM.  INTEGRATION IN EITHER DIRECTION
C         (FORWARD OR BACKWARD IN T) IS PERMITTED.
C
C         IF ITASK = 2 OR 5 (ONE-STEP MODES), TOUT IS IGNORED AFTER
C         THE FIRST CALL (I.E. THE FIRST CALL WITH TOUT .NE. T).
C         OTHERWISE, TOUT IS REQUIRED ON EVERY CALL.
C
C         IF ITASK = 1, 3, OR 4, THE VALUES OF TOUT NEED NOT BE
C         MONOTONE, BUT A VALUE OF TOUT WHICH BACKS UP IS LIMITED
C         TO THE CURRENT INTERNAL T INTERVAL, WHOSE ENDPOINTS ARE
C         TCUR - HU AND TCUR (SEE OPTIONAL OUTPUTS, BELOW, FOR
C         TCUR AND HU).
C
C ITOL  = AN INDICATOR FOR THE TYPE OF ERROR CONTROL.  SEE
C         DESCRIPTION BELOW UNDER ATOL.  USED ONLY FOR INPUT.
C
C RTOL  = A RELATIVE ERROR TOLERANCE PARAMETER, EITHER A SCALAR OR
C         AN ARRAY OF LENGTH NEQ.  SEE DESCRIPTION BELOW UNDER ATOL.
C         INPUT ONLY.
C
C ATOL  = AN ABSOLUTE ERROR TOLERANCE PARAMETER, EITHER A SCALAR OR
C         AN ARRAY OF LENGTH NEQ.  INPUT ONLY.
C
C         THE INPUT PARAMETERS ITOL, RTOL, AND ATOL DETERMINE
C         THE ERROR CONTROL PERFORMED BY THE SOLVER.  THE SOLVER WILL
C         CONTROL THE VECTOR E = (E(I)) OF ESTIMATED LOCAL ERRORS
C         IN Y, ACCORDING TO AN INEQUALITY OF THE FORM
C                     RMS-NORM OF ( E(I)/EWT(I) )   .LE.   1,
C         WHERE       EWT(I) = RTOL(I)*ABS(Y(I)) + ATOL(I),
C         AND THE RMS-NORM (ROOT-MEAN-SQUARE NORM) HERE IS
C         RMS-NORM(V) = SQRT(SUM V(I)**2 / NEQ).  HERE EWT = (EWT(I))
C         IS A VECTOR OF WEIGHTS WHICH MUST ALWAYS BE POSITIVE, AND
C         THE VALUES OF RTOL AND ATOL SHOULD ALL BE NON-NEGATIVE.
```

```
C            THE FOLLOWING TABLE GIVES THE TYPES (SCALAR/ARRAY) OF
C            RTOL AND ATOL, AND THE CORRESPONDING FORM OF EWT(I).
C
C            ITOL    RTOL       ATOL          EWT(I)
C             1      SCALAR     SCALAR     RTOL*ABS(Y(I)) + ATOL
C             2      SCALAR     ARRAY      RTOL*ABS(Y(I)) + ATOL(I)
C             3      ARRAY      SCALAR     RTOL(I)*ABS(Y(I)) + ATOL
C             4      ARRAY      ARRAY      RTOL(I)*ABS(Y(I)) + ATOL(I)
C
C            WHEN EITHER OF THESE PARAMETERS IS A SCALAR, IT NEED NOT
C            BE DIMENSIONED IN THE USER-S CALLING PROGRAM.
C
C            IF NONE OF THE ABOVE CHOICES (WITH ITOL, RTOL, AND ATOL
C            FIXED THROUGHOUT THE PROBLEM) IS SUITABLE, MORE GENERAL
C            ERROR CONTROLS CAN BE OBTAINED BY SUBSTITUTING
C            USER-SUPPLIED ROUTINES FOR THE SETTING OF EWT AND/OR FOR
C            THE NORM CALCULATION.  SEE PART IV BELOW.
C
C            IF GLOBAL ERRORS ARE TO BE ESTIMATED BY MAKING A REPEATED
C            RUN ON THE SAME PROBLEM WITH SMALLER TOLERANCES, THEN ALL
C            COMPONENTS OF RTOL AND ATOL (I.E. OF EWT) SHOULD BE SCALED
C            DOWN UNIFORMLY.
C
C ITASK   = AN INDEX SPECIFYING THE TASK TO BE PERFORMED.
C            INPUT ONLY.  ITASK HAS THE FOLLOWING VALUES AND MEANINGS.
C            1   MEANS NORMAL COMPUTATION OF OUTPUT VALUES OF Y(T) AT
C                T = TOUT (BY OVERSHOOTING AND INTERPOLATING).
C            2   MEANS TAKE ONE STEP ONLY AND RETURN.
C            3   MEANS STOP AT THE FIRST INTERNAL MESH POINT AT OR
C                BEYOND T = TOUT AND RETURN.
C            4   MEANS NORMAL COMPUTATION OF OUTPUT VALUES OF Y(T) AT
C                T = TOUT BUT WITHOUT OVERSHOOTING T = TCRIT.
C                TCRIT MUST BE INPUT AS RWORK(1).  TCRIT MAY BE EQUAL TO
C                OR BEYOND TOUT, BUT NOT BEHIND IT IN THE DIRECTION OF
C                INTEGRATION.  THIS OPTION IS USEFUL IF THE PROBLEM
C                HAS A SINGULARITY AT OR BEYOND T = TCRIT.
C            5   MEANS TAKE ONE STEP, WITHOUT PASSING TCRIT, AND RETURN.
C                TCRIT MUST BE INPUT AS RWORK(1).
C
C            NOTE..  IF ITASK = 4 OR 5 AND THE SOLVER REACHES TCRIT
C            (WITHIN ROUNDOFF), IT WILL RETURN T = TCRIT (EXACTLY) TO
C            INDICATE THIS (UNLESS ITASK = 4 AND TOUT COMES BEFORE TCRIT,
C            IN WHICH CASE ANSWERS AT T = TOUT ARE RETURNED FIRST).
C
C ISTATE  = AN INDEX USED FOR INPUT AND OUTPUT TO SPECIFY THE
C            THE STATE OF THE CALCULATION.
C
C            ON INPUT, THE VALUES OF ISTATE ARE AS FOLLOWS.
C            1   MEANS THIS IS THE FIRST CALL FOR THE PROBLEM
C                (INITIALIZATIONS WILL BE DONE).  SEE NOTE BELOW.
C            2   MEANS THIS IS NOT THE FIRST CALL, AND THE CALCULATION
C                IS TO CONTINUE NORMALLY, WITH NO CHANGE IN ANY INPUT
C                PARAMETERS EXCEPT POSSIBLY TOUT AND ITASK.
C                (IF ITOL, RTOL, AND/OR ATOL ARE CHANGED BETWEEN CALLS
C                WITH ISTATE = 2, THE NEW VALUES WILL BE USED BUT NOT
C                TESTED FOR LEGALITY.)
C            3   MEANS THIS IS NOT THE FIRST CALL, AND THE
C                CALCULATION IS TO CONTINUE NORMALLY, BUT WITH
C                A CHANGE IN INPUT PARAMETERS OTHER THAN
C                TOUT AND ITASK.  CHANGES ARE ALLOWED IN
```

```
C          NEQ, ITOL, RTOL, ATOL, IOPT, LRW, LIW, MF, ML, MU,
C          AND ANY OF THE OPTIONAL INPUTS EXCEPT H0.
C          (SEE IWORK DESCRIPTION FOR ML AND MU.)
C     NOTE..   A PRELIMINARY CALL WITH TOUT = T IS NOT COUNTED
C     AS A FIRST CALL HERE, AS NO INITIALIZATION OR CHECKING OF
C     INPUT IS DONE.  (SUCH A CALL IS SOMETIMES USEFUL FOR THE
C     PURPOSE OF OUTPUTTING THE INITIAL CONDITIONS.)
C     THUS THE FIRST CALL FOR WHICH TOUT .NE. T REQUIRES
C     ISTATE = 1 ON INPUT.
C
C     ON OUTPUT, ISTATE HAS THE FOLLOWING VALUES AND MEANINGS.
C      1  MEANS NOTHING WAS DONE, AS TOUT WAS EQUAL TO T WITH
C         ISTATE = 1 ON INPUT.  (HOWEVER, AN INTERNAL COUNTER WAS
C         SET TO DETECT AND PREVENT REPEATED CALLS OF THIS TYPE.)
C      2  MEANS THE INTEGRATION WAS PERFORMED SUCCESSFULLY.
C     -1  MEANS AN EXCESSIVE AMOUNT OF WORK (MORE THAN MXSTEP
C         STEPS) WAS DONE ON THIS CALL, BEFORE COMPLETING THE
C         REQUESTED TASK, BUT THE INTEGRATION WAS OTHERWISE
C         SUCCESSFUL AS FAR AS T.   (MXSTEP IS AN OPTIONAL INPUT
C         AND IS NORMALLY 500.)  TO CONTINUE, THE USER MAY
C         SIMPLY RESET ISTATE TO A VALUE .GT. 1 AND CALL AGAIN
C         (THE EXCESS WORK STEP COUNTER WILL BE RESET TO 0).
C         IN ADDITION, THE USER MAY INCREASE MXSTEP TO AVOID
C         THIS ERROR RETURN (SEE BELOW ON OPTIONAL INPUTS).
C     -2  MEANS TOO MUCH ACCURACY WAS REQUESTED FOR THE PRECISION
C         OF THE MACHINE BEING USED.   THIS WAS DETECTED BEFORE
C         COMPLETING THE REQUESTED TASK, BUT THE INTEGRATION
C         WAS SUCCESSFUL AS FAR AS T.   TO CONTINUE, THE TOLERANCE
C         PARAMETERS MUST BE RESET, AND ISTATE MUST BE SET
C         TO 3.   THE OPTIONAL OUTPUT TOLSF MAY BE USED FOR THIS
C         PURPOSE.  (NOTE.. IF THIS CONDITION IS DETECTED BEFORE
C         TAKING ANY STEPS, THEN AN ILLEGAL INPUT RETURN
C         (ISTATE = -3) OCCURS INSTEAD.)
C     -3  MEANS ILLEGAL INPUT WAS DETECTED, BEFORE TAKING ANY
C         INTEGRATION STEPS.   SEE WRITTEN MESSAGE FOR DETAILS.
C         NOTE..   IF THE SOLVER DETECTS AN INFINITE LOOP OF CALLS
C         TO THE SOLVER WITH ILLEGAL INPUT, IT WILL CAUSE
C         THE RUN TO STOP.
C     -4  MEANS THERE WERE REPEATED ERROR TEST FAILURES ON
C         ONE ATTEMPTED STEP, BEFORE COMPLETING THE REQUESTED
C         TASK, BUT THE INTEGRATION WAS SUCCESSFUL AS FAR AS T.
C         THE PROBLEM MAY HAVE A SINGULARITY, OR THE INPUT
C         MAY BE INAPPROPRIATE.
C     -5  MEANS THERE WERE REPEATED CONVERGENCE TEST FAILURES ON
C         ONE ATTEMPTED STEP, BEFORE COMPLETING THE REQUESTED
C         TASK, BUT THE INTEGRATION WAS SUCCESSFUL AS FAR AS T.
C         THIS MAY BE CAUSED BY AN INACCURATE JACOBIAN MATRIX,
C         IF ONE IS BEING USED.
C     -6  MEANS EWT(I) BECAME ZERO FOR SOME I DURING THE
C         INTEGRATION.   PURE RELATIVE ERROR CONTROL (ATOL(I)=0.0)
C         WAS REQUESTED ON A VARIABLE WHICH HAS NOW VANISHED.
C         THE INTEGRATION WAS SUCCESSFUL AS FAR AS T.
C
C     NOTE..   SINCE THE NORMAL OUTPUT VALUE OF ISTATE IS 2,
C     IT DOES NOT NEED TO BE RESET FOR NORMAL CONTINUATION.
C     ALSO, SINCE A NEGATIVE INPUT VALUE OF ISTATE WILL BE
C     REGARDED AS ILLEGAL, A NEGATIVE OUTPUT VALUE REQUIRES THE
C     USER TO CHANGE IT, AND POSSIBLY OTHER INPUTS, BEFORE
C     CALLING THE SOLVER AGAIN.
C
```

```
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 (SINGLE PRECISION).
C              THE LENGTH OF RWORK MUST BE AT LEAST
C                 20 + NYH*(MAXORD + 1) + 3*NEQ + LWM     WHERE
C              NYH    = THE INITIAL VALUE OF NEQ,
C              MAXORD = 12 (IF METH = 1) OR 5 (IF METH = 2) (UNLESS A
C                       SMALLER VALUE IS GIVEN AS AN OPTIONAL INPUT),
C              LWM    = 0              IF MITER = 0,
C              LWM    = NEQ**2 + 2     IF MITER IS 1 OR 2,
C              LWM    = NEQ + 2        IF MITER = 3, AND
C              LWM    = (2*ML+MU+1)*NEQ + 2 IF MITER IS 4 OR 5.
C              (SEE THE MF DESCRIPTION FOR METH AND MITER.)
C              THUS IF MAXORD HAS ITS DEFAULT VALUE AND NEQ IS CONSTANT,
C              THIS LENGTH IS..
C                 20 + 16*NEQ                     FOR MF = 10,
C                 22 + 16*NEQ + NEQ**2            FOR MF = 11 OR 12,
C                 22 + 17*NEQ                     FOR MF = 13,
C                 22 + 17*NEQ + (2*ML+MU)*NEQ     FOR MF = 14 OR 15,
C                 20 +  9*NEQ                     FOR MF = 20,
C                 22 +  9*NEQ + NEQ**2            FOR MF = 21 OR 22,
C                 22 + 10*NEQ                     FOR MF = 23,
C                 22 + 10*NEQ + (2*ML+MU)*NEQ     FOR MF = 24 OR 25.
C              THE FIRST 20 WORDS OF RWORK ARE RESERVED FOR CONDITIONAL
C              AND OPTIONAL INPUTS AND OPTIONAL OUTPUTS.
C
C              THE FOLLOWING WORD IN RWORK IS A CONDITIONAL INPUT..
C                 RWORK(1) = TCRIT = CRITICAL VALUE OF T WHICH THE SOLVER
C                            IS NOT TO OVERSHOOT.  REQUIRED IF ITASK IS
C                            4 OR 5, AND IGNORED OTHERWISE.  (SEE ITASK.)
C
C LRW      = THE LENGTH OF THE ARRAY RWORK, AS DECLARED BY THE USER.
C              (THIS WILL BE CHECKED BY THE SOLVER.)
C
C IWORK    = AN INTEGER WORK ARRAY.  THE LENGTH OF IWORK MUST BE AT LEAST
C                 20         IF MITER = 0 OR 3 (MF = 10, 13, 20, 23), OR
C                 20 + NEQ   OTHERWISE (MF = 11, 12, 14, 15, 21, 22, 24, 25).
C              THE FIRST FEW WORDS OF IWORK ARE USED FOR CONDITIONAL AND
C              OPTIONAL INPUTS AND OPTIONAL OUTPUTS.
C
C              THE FOLLOWING 2 WORDS IN IWORK ARE CONDITIONAL INPUTS..
C                 IWORK(1) = ML      THESE ARE THE LOWER AND UPPER
C                 IWORK(2) = MU      HALF-BANDWIDTHS, RESPECTIVELY, OF THE
C                            BANDED JACOBIAN, EXCLUDING THE MAIN DIAGONAL.
C                            THE BAND IS DEFINED BY THE MATRIX LOCATIONS
C                            (I,J) WITH I-ML .LE. J .LE. I+MU.  ML AND MU
C                            MUST SATISFY  0 .LE. ML,MU .LE. NEQ-1.
C                            THESE ARE REQUIRED IF MITER IS 4 OR 5, AND
C                            IGNORED OTHERWISE.  ML AND MU MAY IN FACT BE
C                            THE BAND PARAMETERS FOR A MATRIX TO WHICH
C                            DF/DY IS ONLY APPROXIMATELY EQUAL.
C
C LIW      = THE LENGTH OF THE ARRAY IWORK, AS DECLARED BY THE USER.
C              (THIS WILL BE CHECKED BY THE SOLVER.)
C
```

```
C NOTE.. THE WORK ARRAYS MUST NOT BE ALTERED BETWEEN CALLS TO LSODE
C FOR THE SAME PROBLEM, EXCEPT POSSIBLY FOR THE CONDITIONAL AND
C OPTIONAL INPUTS, AND EXCEPT FOR THE LAST 3*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 LSODE BETWEEN CALLS, IF
C DESIRED (BUT NOT FOR USE BY F OR JAC).
C
C JAC      = 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
C                   SUBROUTINE JAC (NEQ, T, Y, ML, MU, PD, NROWPD)
C                   DIMENSION Y(1), PD(NROWPD,1)
C              WHERE NEQ, T, Y, ML, MU, AND NROWPD ARE INPUT AND THE ARRAY
C              PD IS TO BE LOADED WITH PARTIAL DERIVATIVES (ELEMENTS OF
C              THE JACOBIAN MATRIX) ON OUTPUT.  PD MUST BE GIVEN A FIRST
C              DIMENSION OF NROWPD.  T AND Y HAVE THE SAME MEANING AS IN
C              SUBROUTINE F.  (IN THE DIMENSION STATEMENT ABOVE, 1 IS A
C              DUMMY DIMENSION.. IT CAN BE REPLACED BY ANY VALUE.)
C                   IN THE FULL MATRIX CASE (MITER = 1), ML AND MU ARE
C              IGNORED, AND THE JACOBIAN IS TO BE LOADED INTO PD IN
C              COLUMNWISE MANNER, WITH DF(I)/DY(J) LOADED INTO PD(I,J).
C                   IN THE BAND MATRIX CASE (MITER = 4), THE ELEMENTS
C              WITHIN THE BAND ARE TO BE LOADED INTO PD IN COLUMNWISE
C              MANNER, WITH DIAGONAL LINES OF DF/DY LOADED INTO THE ROWS
C              OF PD.  THUS DF(I)/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 IN THE TWO TRIANGULAR AREAS WHICH
C              CORRESPOND TO NONEXISTENT MATRIX ELEMENTS CAN BE IGNORED
C              OR LOADED ARBITRARILY, AS THEY ARE OVERWRITTEN BY LSODE.
C                   JAC NEED NOT PROVIDE DF/DY EXACTLY.  A CRUDE
C              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 JAC.
C              EACH CALL TO JAC IS PRECEDED BY A CALL TO F WITH THE SAME
C              ARGUMENTS NEQ, T, AND Y.  THUS TO GAIN SOME EFFICIENCY,
C              INTERMEDIATE QUANTITIES SHARED BY BOTH CALCULATIONS MAY BE
C              SAVED IN A USER COMMON BLOCK BY F AND NOT RECOMPUTED BY JAC,
C              IF DESIRED.  ALSO, JAC MAY ALTER THE Y ARRAY, IF DESIRED.
C              JAC MUST BE DECLARED EXTERNAL IN THE CALLING PROGRAM.
C                   SUBROUTINE JAC MAY ACCESS USER-DEFINED QUANTITIES IN
C              NEQ(2),... AND Y(NEQ(1)+1),... IF NEQ IS AN ARRAY
C              (DIMENSIONED IN JAC) AND Y HAS LENGTH EXCEEDING NEQ(1).
C              SEE THE DESCRIPTIONS OF NEQ AND Y ABOVE.
C
C MF       = THE METHOD FLAG.  USED ONLY FOR INPUT.  THE LEGAL VALUES OF
C              MF ARE 10, 11, 12, 13, 14, 15, 20, 21, 22, 23, 24, AND 25.
C              MF HAS DECIMAL DIGITS METH AND MITER.. MF = 10*METH + MITER.
C              METH INDICATES THE BASIC LINEAR MULTISTEP METHOD..
C                METH = 1 MEANS THE IMPLICIT ADAMS METHOD.
C                METH = 2 MEANS THE METHOD BASED ON BACKWARD
C                           DIFFERENTIATION FORMULAS (BDF-S).
C              MITER INDICATES THE CORRECTOR ITERATION METHOD..
C                MITER = 0 MEANS FUNCTIONAL ITERATION (NO JACOBIAN MATRIX
C                            IS INVOLVED).
C                MITER = 1 MEANS CHORD ITERATION WITH A USER-SUPPLIED
C                            FULL (NEQ BY NEQ) JACOBIAN.
C                MITER = 2 MEANS CHORD ITERATION WITH AN INTERNALLY
C                            GENERATED (DIFFERENCE QUOTIENT) FULL JACOBIAN
C                            (USING NEQ EXTRA CALLS TO F PER DF/DY VALUE).
C                MITER = 3 MEANS CHORD ITERATION WITH AN INTERNALLY
```

```
C                         GENERATED DIAGONAL JACOBIAN APPROXIMATION.
C                         (USING 1 EXTRA CALL TO F PER DF/DY EVALUATION).
C            MITER = 4 MEANS CHORD ITERATION WITH A USER-SUPPLIED
C                         BANDED JACOBIAN.
C            MITER = 5 MEANS CHORD ITERATION WITH AN INTERNALLY
C                         GENERATED BANDED JACOBIAN (USING ML+MU+1 EXTRA
C                         CALLS TO F PER DF/DY EVALUATION).
C            IF MITER = 1 OR 4, THE USER MUST SUPPLY A SUBROUTINE JAC
C            (THE NAME IS ARBITRARY) AS DESCRIBED ABOVE UNDER JAC.
C            FOR OTHER VALUES OF MITER, A DUMMY ARGUMENT CAN BE USED.
C-------------------------------------------------------------------------
C OPTIONAL INPUTS.
C
C 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 IOPT = 1, AND IN THAT
C CASE ALL OF THESE INPUTS ARE EXAMINED.  A VALUE OF ZERO FOR ANY
C OF THESE OPTIONAL INPUTS WILL CAUSE THE DEFAULT VALUE TO BE USED.
C THUS TO USE A SUBSET OF THE OPTIONAL INPUTS, SIMPLY PRELOAD
C LOCATIONS 5 TO 10 IN RWORK AND IWORK TO 0.0 AND 0 RESPECTIVELY, AND
C THEN SET THOSE OF INTEREST TO NONZERO VALUES.
C
C NAME      LOCATION         MEANING AND DEFAULT VALUE
C
C H0        RWORK(5)    THE STEP SIZE TO BE ATTEMPTED ON THE FIRST STEP.
C                       THE DEFAULT VALUE IS DETERMINED BY THE SOLVER.
C
C HMAX      RWORK(6)    THE MAXIMUM ABSOLUTE STEP SIZE ALLOWED.
C                       THE DEFAULT VALUE IS INFINITE.
C
C HMIN      RWORK(7)    THE MINIMUM ABSOLUTE STEP SIZE ALLOWED.
C                       THE DEFAULT VALUE IS 0.  (THIS LOWER BOUND IS NOT
C                       ENFORCED ON THE FINAL STEP BEFORE REACHING TCRIT
C                       WHEN ITASK = 4 OR 5.)
C
C MAXORD    IWORK(5)    THE MAXIMUM ORDER TO BE ALLOWED.  THE DEFAULT
C                       VALUE IS 12 IF METH = 1, AND 5 IF METH = 2.
C                       IF MAXORD EXCEEDS THE DEFAULT VALUE, IT WILL
C                       BE REDUCED TO THE DEFAULT VALUE.
C                       IF MAXORD IS CHANGED DURING THE PROBLEM, IT MAY
C                       CAUSE THE CURRENT ORDER TO BE REDUCED.
C
C MXSTEP    IWORK(6)    MAXIMUM NUMBER OF (INTERNALLY DEFINED) STEPS
C                       ALLOWED DURING ONE CALL TO THE SOLVER.
C                       THE DEFAULT VALUE IS 500.
C
C MXHNIL    IWORK(7)    MAXIMUM NUMBER OF MESSAGES PRINTED (PER PROBLEM)
C                       WARNING THAT T + H = T ON A STEP (H = STEP SIZE).
C                       THIS MUST BE POSITIVE TO RESULT IN A NON-DEFAULT
C                       VALUE.  THE DEFAULT VALUE IS 10.
C-------------------------------------------------------------------------
C OPTIONAL OUTPUTS.
C
C AS OPTIONAL ADDITIONAL OUTPUT FROM LSODE, THE VARIABLES LISTED
C BELOW ARE QUANTITIES RELATED TO THE PERFORMANCE OF LSODE
C WHICH ARE AVAILABLE TO THE USER.  THESE ARE COMMUNICATED BY WAY OF
C THE WORK ARRAYS, BUT ALSO HAVE INTERNAL MNEMONIC NAMES AS SHOWN.
C EXCEPT WHERE STATED OTHERWISE, ALL OF THESE OUTPUTS ARE DEFINED
```

```
C ON ANY SUCCESSFUL RETURN FROM LSODE, AND ON ANY RETURN WITH
C ISTATE = -1, -2, -4, -5, OR -6.  ON AN ILLEGAL INPUT RETURN
C (ISTATE = -3), THEY WILL BE UNCHANGED FROM THEIR EXISTING VALUES
C (IF ANY), EXCEPT POSSIBLY FOR TOLSF, LENRW, AND LENIW.
C ON ANY ERROR RETURN, OUTPUTS RELEVANT TO THE ERROR WILL BE DEFINED,
C AS NOTED BELOW.
C
C NAME     LOCATION       MEANING
C
C HU       RWORK(11) THE STEP SIZE IN T LAST USED (SUCCESSFULLY).
C
C HCUR     RWORK(12) THE STEP SIZE TO BE ATTEMPTED ON THE NEXT STEP.
C
C TCUR     RWORK(13) THE CURRENT VALUE OF THE INDEPENDENT VARIABLE
C                    WHICH THE SOLVER HAS ACTUALLY REACHED, I.E. THE
C                    CURRENT INTERNAL MESH POINT IN T.  ON OUTPUT, TCUR
C                    WILL ALWAYS BE AT LEAST AS FAR AS THE ARGUMENT
C                    T, BUT MAY BE FARTHER (IF INTERPOLATION WAS DONE).
C
C TOLSF    RWORK(14) A TOLERANCE SCALE FACTOR, GREATER THAN 1.0,
C                    COMPUTED WHEN A REQUEST FOR TOO MUCH ACCURACY WAS
C                    DETECTED (ISTATE = -3 IF DETECTED AT THE START OF
C                    THE PROBLEM, ISTATE = -2 OTHERWISE).  IF ITOL IS
C                    LEFT UNALTERED BUT RTOL AND ATOL ARE UNIFORMLY
C                    SCALED UP BY A FACTOR OF TOLSF FOR THE NEXT CALL,
C                    THEN THE SOLVER IS DEEMED LIKELY TO SUCCEED.
C                    (THE USER MAY ALSO IGNORE TOLSF AND ALTER THE
C                    TOLERANCE PARAMETERS IN ANY OTHER WAY APPROPRIATE.)
C
C NST      IWORK(11) THE NUMBER OF STEPS TAKEN FOR THE PROBLEM SO FAR.
C
C NFE      IWORK(12) THE NUMBER OF F EVALUATICNS FOR THE PROBLEM SO FAR.
C
C NJE      IWORK(13) THE NUMBER OF JACOBIAN EVALUATIONS (AND OF MATRIX
C                    LU DECOMPOSITIONS) FOR THE PROBLEM SO FAR.
C
C NQU      IWORK(14) THE METHOD ORDER LAST USED (SUCCESSFULLY).
C
C NQCUR    IWORK(15) THE ORDER TO BE ATTEMPTED ON THE NEXT STEP.
C
C IMXER    IWORK(16) THE INDEX OF THE COMPONENT OF LARGEST MAGNITUDE IN
C                    THE WEIGHTED LOCAL ERROR VECTOR ( E(I)/EWT(I) ),
C                    ON AN ERROR RETURN WITH ISTATE = -4 OR -5.
C
C LENRW    IWORK(17) THE LENGTH OF RWORK ACTUALLY REQUIRED.
C                    THIS IS DEFINED ON NORMAL RETURNS AND ON AN ILLEGAL
C                    INPUT RETURN FOR INSUFFICIENT STORAGE.
C
C LENIW    IWORK(18) THE LENGTH OF IWORK ACTUALLY REQUIRED.
C                    THIS IS DEFINED ON NORMAL RETURNS AND ON AN ILLEGAL
C                    INPUT RETURN FOR INSUFFICIENT STORAGE.
C
C THE FOLLOWING TWO ARRAYS ARE SEGMENTS OF THE RWORK ARRAY WHICH
C MAY ALSO BE OF INTEREST TO THE USER AS OPTIONAL OUTPUTS.
C FOR EACH ARRAY, THE TABLE BELOW GIVES ITS INTERNAL NAME,
C ITS BASE ADDRESS IN RWORK, AND ITS DESCRIPTION.
C
C NAME     BASE ADDRESS       DESCRIPTION
C
C YH       21                 THE NORDSIECK HISTORY ARRAY, OF SIZE NYH BY
```

```
C                             (NQCUR + 1), WHERE NYH IS THE INITIAL VALUE
C                             OF NEQ.  FOR J = 0,1,...,NQCUR, COLUMN J+1
C                             OF YH CONTAINS HCUR**J/FACTORIAL(J) TIMES
C                             THE J-TH DERIVATIVE OF THE INTERPOLATING
C                             POLYNOMIAL CURRENTLY REPRESENTING THE SOLUTION,
C                             EVALUATED AT T = TCUR.
C
C ACOR        LENRW-NEQ+1     ARRAY OF SIZE NEQ USED FOR THE ACCUMULATED
C                             CORRECTIONS ON EACH STEP, SCALED ON OUTPUT
C                             TO REPRESENT THE ESTIMATED LOCAL ERROR IN Y
C                             ON THE LAST STEP.  THIS IS THE VECTOR E IN
C                             THE DESCRIPTION OF THE ERROR CONTROL.  IT IS
C                             DEFINED ONLY ON A SUCCESSFUL RETURN FROM LSODE.
C
C----------------------------------------------------------------------
C PART II.  OTHER ROUTINES CALLABLE.
C
C THE FOLLOWING ARE OPTIONAL CALLS WHICH THE USER MAY MAKE TO
C GAIN ADDITIONAL CAPABILITIES IN CONJUNCTION WITH LSODE.
C (THE ROUTINES XSETUN AND XSETF ARE DESIGNED TO CONFORM TO THE
C SLATEC ERROR HANDLING PACKAGE.)
C
C     FORM OF CALL                    FUNCTION
C     CALL XSETUN(LUN)        SET THE LOGICAL UNIT NUMBER, LUN, FOR
C                             OUTPUT OF MESSAGES FROM LSODE, IF
C                             THE DEFAULT IS NOT DESIRED.
C                             THE DEFAULT VALUE OF LUN IS 6.
C
C     CALL XSETF(MFLAG)       SET A FLAG TO CONTROL THE PRINTING OF
C                             MESSAGES BY LSODE.
C                             MFLAG = 0 MEANS DO NOT PRINT. (DANGER..
C                             THIS RISKS LOSING VALUABLE INFORMATION.)
C                             MFLAG = 1 MEANS PRINT (THE DEFAULT).
C
C                             EITHER OF THE ABOVE CALLS MAY BE MADE AT
C                             ANY TIME AND WILL TAKE EFFECT IMMEDIATELY.
C
C     CALL SVCOM (RSAV, ISAV) STORE IN RSAV AND ISAV THE CONTENTS
C                             OF THE INTERNAL COMMON BLOCKS USED BY
C                             LSODE (SEE PART III BELOW).
C                             RSAV MUST BE A REAL ARRAY OF LENGTH 219
C                             OR MORE, AND ISAV MUST BE AN INTEGER
C                             ARRAY OF LENGTH 41 OR MORE.
C
C     CALL RSCOM (RSAV, ISAV) RESTORE, FROM RSAV AND ISAV, THE CONTENTS
C                             OF THE INTERNAL COMMON BLOCKS USED BY
C                             LSODE.  PRESUMES A PRIOR CALL TO SVCOM
C                             WITH THE SAME ARGUMENTS.
C
C                             SVCOM AND RSCOM ARE USEFUL IF
C                             INTERRUPTING A RUN AND RESTARTING
C                             LATER, OR ALTERNATING BETWEEN TWO OR
C                             MORE PROBLEMS SOLVED WITH LSODE.
C
C     CALL INTDY(,,,,,)       PROVIDE DERIVATIVES OF Y, OF VARIOUS
C          (SEE BELOW)        ORDERS, AT A SPECIFIED POINT T, IF
C                             DESIRED.  IT MAY BE CALLED ONLY AFTER
C                             A SUCCESSFUL RETURN FROM LSODE.
C
C THE DETAILED INSTRUCTIONS FOR USING INTDY ARE AS FOLLOWS.
```

```
C THE FORM OF THE CALL IS..
C
C    CALL INTDY (T, K, RWORK(21), NYH, DKY, IFLAG)
C
C THE INPUT PARAMETERS ARE..
C
C T           = VALUE OF INDEPENDENT VARIABLE WHERE ANSWERS ARE DESIRED
C               (NORMALLY THE SAME AS THE T LAST RETURNED BY LSODE).
C               FOR VALID RESULTS, T MUST LIE BETWEEN TCUR - HU AND TCUR.
C               (SEE OPTIONAL OUTPUTS FOR TCUR AND HU.)
C K           = INTEGER ORDER OF THE DERIVATIVE DESIRED.  K MUST SATISFY
C               0 .LE. K .LE. NQCUR, WHERE NQCUR IS THE CURRENT ORDER
C               (SEE OPTIONAL OUTPUTS).  THE CAPABILITY CORRESPONDING
C               TO K = 0, I.E. COMPUTING Y(T), IS ALREADY PROVIDED
C               BY LSODE DIRECTLY.  SINCE NQCUR .GE. 1, THE FIRST
C               DERIVATIVE DY/DT IS ALWAYS AVAILABLE WITH INTDY.
C RWORK(21) = THE BASE ADDRESS OF THE HISTORY ARRAY YH.
C NYH         = COLUMN LENGTH OF YH, EQUAL TO THE INITIAL VALUE OF NEQ.
C
C THE OUTPUT PARAMETERS ARE..
C
C DKY         = A REAL ARRAY OF LENGTH NEQ CONTAINING THE COMPUTED VALUE
C               OF THE K-TH DERIVATIVE OF Y(T).
C IFLAG       = INTEGER FLAG, RETURNED AS 0 IF K AND T WERE LEGAL,
C               -1 IF K WAS ILLEGAL, AND -2 IF T WAS ILLEGAL.
C               ON AN ERROR RETURN, A MESSAGE IS ALSO WRITTEN.
C-----------------------------------------------------------------------
C PART III.  COMMON BLOCKS.
C
C IF LSODE IS TO BE USED IN AN OVERLAY SITUATION, THE USER
C MUST DECLARE, IN THE PRIMARY OVERLAY, THE VARIABLES IN..
C    (1) THE CALL SEQUENCE TO LSODE,
C    (2) THE TWO INTERNAL COMMON BLOCKS
C         /LS0001/  OF LENGTH  258  (219 SINGLE PRECISION WORDS
C                        FOLLOWED BY 39 INTEGER WORDS),
C         /EH0001/  OF LENGTH  2 (INTEGER WORDS).
C
C IF LSODE 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 IF THE SOLUTION OF A GIVEN PROBLEM BY LSODE 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,
C FOLLOWING THE RETURN FROM THE LAST LSODE CALL PRIOR TO THE
C INTERRUPTION, THE CONTENTS OF THE CALL SEQUENCE VARIABLES AND THE
C INTERNAL COMMON BLOCKS, AND LATER RESTORE THESE VALUES BEFORE THE
C NEXT LSODE CALL FOR THAT PROBLEM.  TO SAVE AND RESTORE THE COMMON
C BLOCKS, USE SUBROUTINES SVCOM AND RSCOM (SEE PART II ABOVE).
C
C NOTE.. IN THIS VERSION OF LSODE, THERE ARE TWO DATA STATEMENTS,
C IN SUBROUTINES LSODE AND XERRWV, WHICH LOAD VARIABLES INTO THESE
C LABELED COMMON BLOCKS.  ON SOME SYSTEMS, IT MAY BE NECESSARY TO
C MOVE THESE TO A SEPARATE BLOCK DATA SUBPROGRAM.
C
C-----------------------------------------------------------------------
C PART IV.  OPTIONALLY REPLACEABLE SOLVER ROUTINES.
C
C BELOW ARE DESCRIPTIONS OF TWO ROUTINES IN THE LSODE PACKAGE WHICH
```

```
C RELATE TO THE MEASUREMENT OF ERRORS.  EITHER ROUTINE CAN BE
C REPLACED BY A USER-SUPPLIED VERSION, IF DESIRED.  HOWEVER, SINCE SUCH
C A REPLACEMENT MAY HAVE A MAJOR IMPACT ON PERFORMANCE, IT SHOULD BE
C 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 (A) EWSET.
C THE FOLLOWING SUBROUTINE IS CALLED JUST BEFORE EACH INTERNAL
C INTEGRATION STEP, AND SETS THE ARRAY OF ERROR WEIGHTS, EWT, AS
C DESCRIBED UNDER ITOL/RTOL/ATOL ABOVE..
C      SUBROUTINE EWSET (NEQ, ITOL, RTOL, ATOL, YCUR, EWT)
C WHERE NEQ, ITOL, RTOL, AND ATOL ARE AS IN THE LSODE CALL SEQUENCE,
C YCUR CONTAINS THE CURRENT DEPENDENT VARIABLE VECTOR, AND
C EWT IS THE ARRAY OF WEIGHTS SET BY EWSET.
C
C IF THE USER SUPPLIES THIS SUBROUTINE, IT MUST RETURN IN EWT(I)
C (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 VNORM ROUTINE (SEE BELOW), AND ALSO USED BY LSODE IN THE COMPUTATION
C OF THE OPTIONAL OUTPUT IMXER, 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 NQ
C ARE AVAILABLE FROM THE HISTORY ARRAY YH, DESCRIBED ABOVE UNDER
C OPTIONAL OUTPUTS.  IN EWSET, YH IS IDENTICAL TO THE YCUR ARRAY,
C EXTENDED TO NQ + 1 COLUMNS WITH A COLUMN LENGTH OF NYH AND SCALE
C FACTORS OF H**J/FACTORIAL(J).  ON THE FIRST CALL FOR THE PROBLEM,
C GIVEN BY NST = 0, NQ IS 1 AND H IS TEMPORARILY SET TO 1.0.
C THE QUANTITIES NQ, NYH, H, AND NST CAN BE OBTAINED BY INCLUDING
C IN EWSET THE STATEMENTS..
C      COMMON /LS0001/ RLS(219),ILS(39)
C      NQ = ILS(35)
C      NYH = ILS(14)
C      NST = ILS(36)
C      H = RLS(213)
C THUS, FOR EXAMPLE, THE CURRENT VALUE OF DY/DT CAN BE OBTAINED AS
C YCUR(NYH+I)/H  (I=1,...,NEQ)  (AND THE DIVISION BY H IS
C UNNECESSARY WHEN NST = 0).
C
C (B) VNORM.
C THE FOLLOWING IS A REAL FUNCTION ROUTINE WHICH COMPUTES THE WEIGHTED
C 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( (1/N) * SUM(V(I)*W(I))**2 ).
C VNORM IS CALLED WITH N = NEQ AND WITH W(I) = 1.0/EWT(I), WHERE
C EWT IS AS SET BY SUBROUTINE EWSET.
C
C IF THE USER SUPPLIES THIS FUNCTION, IT SHOULD RETURN A NON-NEGATIVE
C VALUE OF VNORM SUITABLE FOR USE IN THE ERROR CONTROL IN LSODE.
C NONE OF THE ARGUMENTS SHOULD BE ALTERED BY VNORM.
C FOR EXAMPLE, A USER-SUPPLIED VNORM ROUTINE MIGHT..
C   -SUBSTITUTE A MAX-NORM OF (V(I)*W(I)) FOR THE RMS-NORM, OR
C   -IGNORE SOME COMPONENTS OF V IN THE NORM, WITH THE EFFECT OF
C      SUPPRESSING THE ERROR CONTROL ON THOSE COMPONENTS OF Y.
```

```
C------------------------------------------------------------------------
C------------------------------------------------------------------------
C OTHER ROUTINES IN THE LSODE PACKAGE.
C
C IN ADDITION TO SUBROUTINE LSODE, THE LSODE PACKAGE INCLUDES THE
C FOLLOWING SUBROUTINES AND FUNCTION ROUTINES..
C   INTDY     COMPUTES AN INTERPOLATED VALUE OF THE Y VECTOR AT T = TOUT.
C   STODE     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   PREPJ     COMPUTES AND PREPROCESSES THE JACOBIAN MATRIX J = DF/DY
C             AND THE NEWTON ITERATION MATRIX P = I - H*L0*J.
C   SOLSY     MANAGES SOLUTION OF LINEAR SYSTEM IN CHORD ITERATION.
C   EWSET     SETS THE ERROR WEIGHT VECTOR EWT BEFORE EACH STEP.
C   VNORM     COMPUTES THE WEIGHTED R.M.S. NORM OF A VECTOR.
C   SVCOM AND RSCOM   ARE USER-CALLABLE ROUTINES TO SAVE AND RESTORE,
C             RESPECTIVELY, THE CONTENTS OF THE INTERNAL COMMON BLOCKS.
C   SGEFA AND SGESL   ARE ROUTINES FROM LINPACK FOR SOLVING FULL
C             SYSTEMS OF LINEAR ALGEBRAIC EQUATIONS.
C   SGBFA AND SGBSL   ARE ROUTINES FROM LINPACK FOR SOLVING BANDED
C             LINEAR SYSTEMS.
C   SAXPY, SSCAL, ISAMAX, AND SDOT   ARE BASIC LINEAR ALGEBRA MODULES
C             (BLAS) USED BY THE ABOVE LINPACK ROUTINES.
C   R1MACH    COMPUTES THE UNIT ROUNDOFF IN A MACHINE-INDEPENDENT MANNER.
C   XERRWV, XSETUN, AND XSETF   HANDLE THE PRINTING OF ALL ERROR
C             MESSAGES AND WARNINGS.  XERRWV IS MACHINE-DEPENDENT.
C NOTE..  VNORM, ISAMAX, SDOT, AND R1MACH ARE FUNCTION ROUTINES.
C ALL THE OTHERS ARE SUBROUTINES.
C
C THE INTRINSIC AND EXTERNAL ROUTINES USED BY LSODE ARE..
C ABS, AMAX1, AMIN1, FLOAT, MAX0, MIN0, MOD, SIGN, SQRT, AND WRITE.
C
C------------------------------------------------------------------------
C  END OF WRITEUP
```

APPENDIX D
SENSITIVITY DERIVATION

APPENDIX D. Sensitivity Derivation

This appendix gives a detailed description of the derivation of the sensitivity analysis equations as given by Piepho [D1]. The derivation has been supplemented in some areas for clarity and completeness, but it is basically as presented by Piepho. Some generalization of the original presentation is also added, again for clarity. The symbols have been changed from the original work to be consistent with this thesis.

DERIVATION OF IMPORTANCE FUNCTIONS

Adjoint or importance theory in this context assumes a set of model equations in the form of ordinary differential equations (ODE's):

$$\frac{dC_i}{dt} = f_i(C,\eta,t) \qquad\qquad D.1$$

where C represents the variables of interest, η represents the parameters of interest, and t generally represents time. As described in Chapter 5, the requirement of ODE's is not at all restrictive, as many spatially-dependent problems can be formulated in this way and have been solved[A2].

The response functions for this system are defined, in general form, as:

$$\mathfrak{R} = \int_0^{t_f} L(C,\eta,t)\, dt \qquad\qquad \text{D.2}$$

This implies the following:

$$\dot{\mathfrak{R}} = L(C,\eta,t) \quad \text{with } \mathfrak{R}(0) = 0.0 \qquad\qquad \text{D.3}$$

It is of interest to know how a change in the model will affect the response variable. The differential change in response is defined using the following definition:

$$\delta\mathfrak{R} = \sum_{i=1}^{n} C_i^*(t')\, \delta C_i' \qquad\qquad \text{D.4}$$

where C_i^* is the importance function associated with the ith variable C_i, and $\delta C_i'$ is the momentary change in C_i at time t'. The following analysis provides a means of evaluating the importance functions and also the overall response to changes in the model.

Let the momentary change in one or more of the variables be denoted by:

$$C_i(t) = \begin{bmatrix} C_{io}(t) & t < t' \\ C_{io}(t) + \delta C_i & t = t' \\ C_{io}(t) + \delta C_i(t) & t > t' \end{bmatrix} \qquad \text{D.5}$$

where one or more of the $\delta C_i'$ are non-zero at $t = t'$. We assume that the changes in the variables are small relative to the variable itself; therefore, the original system (Equation D.1) can be expanded in a Taylor series about the unperturbed value C_0, neglecting higher terms of the series, to obtain:

$$\frac{dC_{io}(t)}{dt} + \frac{d\left[dC_i(t)\right]}{dt} = f_i(C_o) + \frac{\partial f_i}{\partial C_k}\Big|_{C_o(t)} \delta C_k(t) \qquad \text{D.6}$$

A summation over the index k is also not shown but should be assumed. Subtracting Equation D.1 gives the following:

$$\frac{d\left[dC_i(t)\right]}{dt} = \frac{\partial f_i}{\partial C_k}\Big|_{C_o(t)} \delta C_k(t) \qquad \text{D.7}$$

with the initial condition

$$\delta C_i(t') = C'_i \qquad \text{D.8}$$

Applying the same small changes to the response function, Equation D.2, and neglecting summation over the index i for the moment yields:

$$\delta\Re = \int_0^{t_f} \frac{\partial L}{\partial C_i} \Big|_{C_o(t')} \, \delta C_i(t') \, dt' \qquad\qquad D.9$$

This can be rewritten as:

$$\delta\Re = \int_0^{t} \frac{\partial L}{\partial C_i} \delta C_i \, dt' \; + \; \int_t^{t_f} \frac{\partial L}{\partial C_i} \delta C_i \, dt' \qquad\qquad D.10$$

where t is arbitrary on the interval from 0 to t_f. Arbitrary changes to the system prior to time t are duplicated by assuming that an appropriate momentary change $\delta C_i(t)$ is made to the system at time t. Therefore, substituting the definition of D.4 we get:

$$\int_0^{t_f} \frac{\partial L}{\partial C_i} \delta C_i \, dt' \; = \; C_i^*(t) \, \delta C_i(t) \qquad\qquad D.11$$

Introducing this result into D.10 and differentiating with respect to t yields:

$$0 = \frac{\partial L}{\partial C_i} \delta C_i(t) + \frac{d}{dt}\left[C_i^* \delta C_i\right] \qquad \text{D.12}$$

Substituting D.7 and rearranging gives:

$$0 = \delta C_i \left[\frac{\partial L}{\partial C_i} + \frac{dC_i^*}{dt} + \sum_{k=1}^{n} C_k^* \frac{\partial f_k}{\partial C_i}\right] \qquad \text{D.13}$$

This equation must hold for the case of small, finite, changes in δC_i, therefore, the parenthetical portion of D.13 must go to zero:

$$0 = \frac{\partial L}{\partial C_i} + \frac{dC_i^*}{dt} + \sum_{k=1}^{n} C_k^* \frac{\partial f_k}{\partial C_i} \qquad \text{D.14}$$

This is then a set of ordinary differential equations in C_i^*, with the initial condition that:

$$C_i^* (t_f) = 0 \qquad \text{D.15}$$

This states simply that things that occur at time t_f have no effect on the response of the system. This new system is solved backwards in time; hence the label of "adjoint" theory is often given to this type of analysis.

TOTAL SENSITIVITY

An important result that is derived from the calculation of the importance functions is the total sensitivity equation. This equation can be used to get the total relative change in a particular response due to a change in particular variables or parameters of the model. The derivation of total sensitivity starts by differentiating Equation D.2 with respect to some parameter of interest, say $\Re$, and applying the chain rule:

$$\frac{d\Re}{d\alpha} = \int_0^{t_f} \left[\frac{\partial L}{\partial \alpha} + \frac{\partial L}{\partial C} \frac{dC}{d\alpha} \right] dt \qquad D.16$$

In this case there is no loss of generality by using only one process variable, C, to perform the derivation; the final form will be generalized to an arbitrary number of variables. Using D.14 to eliminate the $\partial L/\partial C$ term from D.16 gives:

$$\frac{\delta\Re}{\delta\alpha} = \int \frac{\partial L}{\partial \alpha} dt - \int \frac{\partial C^*}{\partial t} \frac{dC}{d\alpha} dt - \int C^* \frac{\partial f}{\partial C} \frac{dC}{d\alpha} dt \qquad D.17$$

The first term gives the explicit sensitivity of the total sensitivity and should be straightforward to calculate. The second term is reduced to simpler terms by an integration by parts:

$$-\int \frac{\partial C^*}{\partial t} \frac{dC}{d\alpha} dt \quad = -\int dC^* \frac{dC}{d\alpha}$$

$$= -C^* \frac{dC}{d\alpha}\Big|_0^{t_f} - \left[-\int C^* d\left[\frac{dC}{d\alpha}\right]\right]$$

$$= -C^* \frac{dC}{d\alpha}(t_f) + C^* \frac{dC}{d\alpha}(0) + \int C^* \frac{d}{dt}\left[\frac{dC}{d\alpha}\right]dt$$

$$= C^* \frac{dC}{d\alpha}(0) + \int C^* \frac{df}{d\alpha} dt$$

$$= C^* \frac{dC}{d\alpha}(0) + \int \left[C^* \frac{\partial f}{\partial \alpha} + C^* \frac{\partial f}{\partial C} \frac{dC}{d\alpha}\right] dt$$

$$D.18$$

The third term of Equation D.17 is equal and opposite to the last term in D.18. Cancelling these terms and rearranging Equation D.17 yields the total sensitivity equation, now with summation over multiple variables:

$$\frac{\delta\mathfrak{R}}{\delta\alpha} = \sum_{k=1}^{n}\left[C_k^*(0) \frac{dC_k(0)}{d\alpha}\right] + \int_0^{t_f}\left[\frac{\partial L}{\partial\alpha} + \sum_{k=1}^{n}\left[C_k^* \frac{\partial f_k}{\partial\alpha}\right]\right] dt \quad D.19$$

The C_k^*'s are calculated from Equation D.13 and D.14; L is known explicitly(or should be); f_k is determined in the forward calculation, Equation D.1; and α is arbitrarily chosen as one of the model

parameters or one of the variables. The only difficulty may lie in the calculation of the initial effects of α on the variables, i.e., the $dC_k(0)/d\alpha$ term. In general, the expression for C_k is not known explicitly (we know only dC_k/dt explicitly), but for most cases of interest this quantity can be estimated. If α is chosen as the initial condition of a variable ($\alpha=C_i(0)$) then:

$$\frac{dC_k(0)}{d\alpha} = \delta_{i,k} \begin{bmatrix} \text{Kronecker delta;} \\ = 1 \text{ if } i = k \\ = 0 \text{ if } i \neq k \end{bmatrix} \qquad \text{D.20}$$

In other cases, this quantity will have to be estimated if it is deemed to be important to the calculation.

REFERENCES

D1   Piepho, M. G., K. B. Cady, and M. A. Kenton. 1981. "An Importance Theory for Nonlinear Ordinary Differential Equations," HEDL-TME 81-24, UC-79T, Tp. Hanford Engineering Development Laboratory, Richland, Washington.

APPENDIX E

MITIRAD USERS' GUIDE

User's Guide to the MITIRAD Code package

Scott A. Simonson[1]

August, 1988

---

[1]Massachusetts Institute of Technology, Department of Nuclear Engineering

SUMMARY

This document describes the code package MITIRAD 1.0, developed at the Massachusetts Institute of Technology. MITIRAD was designed to solve problems related to the radiation chemistry of aqueous and two-phase systems, including a complete sensitivity analysis of all model parameters. The package can consider homogeneous and heterogeneous reaction schemes and may be easily modified to accommodate many possible problems. Version 1.0 was intended to operate as a "black box" for the most part so no user intervention into the numerics is needed for a wide range of problems.

TABLE OF CONTENTS

1.0 INTRODUCTION

The computer code MITIRAD was written to solve equations describing various aspects of the radiolysis of aqueous solutions. The accompanying codes MITIAD and LINTY perform an adjoint sensitivity analysis of the equations solved by MITIRAD. The codes were designed for ease of use and for a wide range of situations encountered in radiation chemistry. This appendix presents the necessary information for the use of these computer codes on a MicroVAX computer. The coding is also provided as part of the thesis; therefore, modifications can be performed easily. Details of possible modifications are provided in Chapter 8 of the thesis. Although the codes are fairly robust, they are not bullet proof and some unforeseen errors are possible. Most of the time this is due to the user trying to force the code to perform calculations without having the proper equations or initial conditions to represent the physical reality of the systems being modeled. The codes may work in these instances (i.e. run without error) and provide useless output. When analyzing chemical systems, it is best to run an equilibrium type computer code such as MINEQL[F1] or EQ3/6[F2] using just the major constituents to get the initial conditions close to those of the real system before running MITIRAD.

2.0 USING MITIRAD VERSION 1.0

The version 1.0 of MITIRAD was designed as a black box type code to be used for radiation chemistry problems related to nuclear waste management. Limited computer knowledge is assumed, and no programming experience is needed to operate this version. The default settings for most of the numerical aspects of the code are suitable for most problems. The user of the code need only supply an input file with the following:

a) the appropriate rate constants for the following generalized chemical equation(s) to simulate chemical kinetics:

$$[A] + [B] + [C] \Longrightarrow [D] + [E] + [F] + [G]$$

b) G-values for the radiolytic production of species and dose rates of the high and low LET radiations

c) activation energies for Arrhenius temperature dependence when temperature dependencies are not explicitly entered into part a, above.

d) boundary conditions and diffusion coefficients when transport is required.

e) initial and final times for the calculation.

f) liquid and gas volumes, and Henry's Law coefficients for partitioning between the liquid and gas phases.

Although the MITIRAD, MITIAD and LINTY codes were written for radiation chemistry problems, the code can be used to model other phenomena. The first example in this guide models the burning of a cesium flare, and the second example is a simple model of atmospheric ozone concentrations.

The subsequent sections of this appendix give the details necessary to understand the code design, design input data sets for MITIRAD, MITIAD and LINTY, and interpret results. Throughout the text, aid is provided to interpret potential errors/problems. The following section gives a brief description of the mathematical form of the models and how they are solved.

3.0 MATHEMATICAL FORM OF THE EQUATIONS

The package solves three distinct systems of equations. The first is a system of ordinary differential equations (ODE's) that are solved by MITIRAD forward in time, usually from time zero to some later time t_f, to obtain concentrations of species as a function of time. The second system is also a set of ODE's but MITIAD solves these backwards in time to obtain importance functions. The last system is an integration of the importance functions by LINTY to obtain the total sensitivities of the model.

3.1 MITIRAD Equations

The mathematical form of the equations solved by MITIRAD is derived from chemical kinetics and one-dimensional transport theory and accounts for first, second, and third order chemical reactions, as well as accounting for catalytic chemical reactions and convective and diffusive transport. The mathematical representation of these reactions is a set of ordinary differential equations (ODE's) that are coupled with source terms for the radiolytic generation of various radical and molecular species and integrated over time by MITIRAD to give the concentration of species as a function of time. The generalized form of the equations solved by MITIRAD for the individual concentrations, C_i, is as follows:

$$\frac{dC_i}{dt} = \sum_R G_i D_R \qquad \text{Production}$$

$$+ \; D_i \nabla^2 C_i \; + \; u \; div(C_i) \qquad \text{Transport}$$

$$+ \sum_{j=1}^{nrtn} k_j \mu_{ij} \prod_{n=1}^{neq} C^{|\mu_{in}|} \qquad \text{Kinetics}$$

where

i is the number of the individual species (of quantity neq),

R is type radiation,

j is the reaction number,

n is synonymous with i.

G is the production rate of species i for radiation type R,

D_r is the dose rate of radiation type R.

D is the diffusion coefficient of species i,

t is the time variable.

nrtn is the total number of reactions,

k is the reaction rate constant of reaction j,

μ is the stoichiometric coefficient for species i in reaction j

MITIRAD automatically sets up the above ODE's, so that the user only needs to provide the chemical equations and the above parameters as a data file, described in the Data Input Section.

3.2 MITIAD and LINTY Equations

The accompanying codes MITIAD and LINTY perform an adjoint sensitivity analysis of the equations solved by MITIRAD. The results of these analyses are the normalized sensitivities of a species of interest (the response) relative to changes in the parameters of the system. Sensitivities for a particular species with respect to all of the rate constants, g-values, dose rates, and diffusion coefficients are calculated. The code can be modified to provide an abbreviated set of sensitivities but in the "black-box" version, everything is calculated.

MITIAD calculates the importance functions, C_i^* utilizing the following formulation, derived in more detail in Chapter 5 and Appendix D of the thesis:

$$\frac{dC_i^*}{dt} = -\frac{dL}{dC_i} - \sum_{k=1}^{NEQ} C_k^* \frac{df_k}{dC_i}$$

$$C_i^*(t_f) = 0.0$$

The variable L is the response rate; the functional form of this is simply:

$$L = \frac{dC_{IA}}{dt}$$

where IA is the index of a species of interest, chosen by the user. This equation is integrated backwards in time, from the final time

t_f to zero, in much the same way that MITIRAD integrates the "forward" solutions.

Once the importance functions have been obtained, the total sensitivities can be evaluated. The context of total sensitivity is defined as:

$$\text{Total Sensitivity} = \frac{[\text{parameter}]}{[\text{Concentration}]} \frac{d[\text{Concentration}]}{d[\text{parameter}]}$$

This can also be interpreted as the relative sensitivity of a particular concentration to a fractional change in a particular parameter. The equation used to evaluate the total sensitivity is described by:

$$\frac{\alpha}{C}\frac{\delta C}{\delta \alpha} = \sum_{k=1}^{NEQ}\left[C_k^*(0)\ \frac{dC_k(0)}{d\alpha}\right] + \int_0^{t_f}\left[\frac{\partial L}{\partial \alpha} + \sum_{k=1}^{NEQ}\left[C_k^*\ \frac{\partial f_k}{\partial \alpha}\right]\right]$$

where α is used to represent a parameter of interest. This is integrated using a trapezoidal integration method. The variable C is the concentration associated with the response rate L.

4.0 COMPUTATIONAL ACCURACY AND STABILITY CONDITIONS

MITIRAD provides for the selection of the absolute and relative tolerances required for a particular calculation. The default values for MITIRAD are 10^{-15} and 10^{-5} for the absolute and relative tolerances, respectively. The value for the relative tolerance is tight enough that the computational error introduced by the calculation is small compared to the uncertainty associated with the input data. The absolute tolerance value is problem dependent, but with proper scaling, the default value should be adequate for most problems. The values may be altered by changing the values for ATOL and RTOL in the namelist LSIN, in the Namelist file defined by the file "SET.FIL".

Due to the nature of the species produced by radiation, i.e. very short-lived, the code was required to integrate stiff systems of equations (stiff meaning very different characteristic rate constants for the various species.) The integration of the stiff equations was accomplished through the use of the LSODE (Livermore Solver Ordinary Differential Equations) [F3] subroutine package which is based upon Gear's method of solution of stiff ODE's [F4]. The description of the LSODE package of subroutines is provided in Appendix G.

The software was written in the widely used VAX Fortran (77) language using double precision real numbers, and the source code is supplied with this document in Appendix G. The code listing is liberally annotated, and details related to the code can be

found in the listing. The code is very efficient and, with slight modification, is amenable to vectorization. The non-standard statement NAMELIST is used for much of the input. The appropriate input statements would have to be added to the code when used on systems that do not support this FORTRAN construct.

5.0 DATA INPUT

This section describes the necessary data input used and generated by MITIRAD, MITIAD and LINTY. A brief section discussing the use of command files to run the code in batch (background) mode on the MicroVAX computer is also included since this would be the preferable method of making computer runs. A completely interactive method of running the computer codes is also available that allows for the use of most of the options of the code package. This is also described in the Data Input Section.

5.1 Command Files

Since MITIRAD, MITIAD and LINTY were written on a VAX computer, much of the input/output is designed to operate under the VMS operating system. The codes generally take 10 minutes to an hour to run so it is best to run them in batch (background) mode. This is accomplished by using a command file (one called MITIRAD.COM is provided) that is submitted to batch mode using:

```
$SUBMIT MITIRAD.COM
```

A simplified, yet typical command file for running MITIRAD under VMS after the data files have been generated may look like:

```
$ !    Command File:   MITIRAD.COM
$!
$      SET NOVERIFY
```

```
$!
$     SET DEFAULT DUA0:[MITIRAD]
$!
$     RUN MITIRAD
$     RUN MITIAD
$     RUN LINTY
$!
```

The first two commands tell the system not to verify the reads and writes, and to define the default directory as DUA0:[MITIRAD] (exclamation marks allow for comments.)   The default directory must contain the executable file MITIRAD.EXE and the input data files described below.   The third, forth and fifth command lines run the codes MITIRAD, MITIAD and LINTY.   A new output file will be generated in the default directory each time MITIRAD is run. Depending on the needs of the user, the command file may be altered to provide other features in the documentation of the run (e.g., incorporation of standard blocks of text), or to only run a portion of the codes.   The system will notify the user of the completion of the program execution if the user is logged onto the system when this occurs.

5.2   Data Files

MITIRAD requires certain data files to tell it what to do. These files can be generated directly using the editor provided on the MicroVAX in the formats described below.   In addition, they can be generated using the program FILEMAKER that is provided. It is highly recommended that FILEMAKER be used to generate the

input files the first time around. This guarantees that the formats will be correct, a major source of errors when starting out.

After having logged into the account that contains the MITIRAD files, the user can invoke FILEMAKER using the command:

$RUN FILEMAKER

The program begins by generating the file SET.FIL, described in the next section. The user is only prompted for two names: the input and namelist file names. The remaining file names are given default values. It is very important to edit the file SET.FIL after running FILEMAKER to make the output file names unique to the particular run. These names are used in all the output files and in the graphics to identify particular runs so it is important that they are unique. It is advisable to review the variables described in namelist STATE, Figure F.2, as FILEMAKER will prompt the user for each of these quantities. The user will also be prompted to choose the appropriate reaction data sets, boundary conditions for each species (if it is a spatially dependent run, i.e. MESHPTS > 0) and the initial value for each species. It is instructive to run FILEMAKER once to become familiar with the information requests and then again when you have compiled the necessary information.

The standard data input consists of three files: one that reads the names of the files with which the particular run of the code will interact, one that contains the input data for a particular run of the code, and another that contains optional changes to the numerical parameters of the code. The first file must be entitled 'SET.FIL'.

The format and description of SET.FIL are given in Figure F.1. If FILEMAKER was run to generate the input files, this is the only file that will need to be modified to run MITIRAD. FILEMAKER indents the files that need to be edited in SET.FIL. This indentation should be removed (i.e. the file names should be left justified) and the files should be given unique names.

SET.FIL establishes the thread that the codes use to interact with each other. It is very important to have the names of the files given in SET.FIL uniquely identify particular runs of the code. SET.FIL and the input files named in records 1 and 4 must exist in the default directory or an error will result. The VMS environment will generate the output files. When no sensitivity calculations are being performed, only the first four records are needed. The file named in record 3 is used to transmit intermediate results from MITIRAD to MITIAD, and also to the RS/1 routine POST_PRE, that is used to generate plots. The formats of all files are described in the next section. In general, the user only needs to be concerned with the format of the files defined in records 1 and 4.

Figure F.1:   Format and description for MITIRAD input file 'SET.FIL'

RECORD	FORMAT	DESCRIPTION	DEFAULT
1	A35	Input file name	none
2	A35	Output file name	none
3	A35	Plot file name	none
4	A35	Namelist File Name	none
5	A35	Output from Adjoint	none
6	A35	Intermediate Results	none
7	A35	Sensitivity Output file	none
8	A40	Spline Information File	none
9	A35	Diagnostics File	none

5.3 Input File

If the user begins by running FILEMAKER, the input file will already exist and no editing will be needed. Figure F.2 contains the file format for the input file named in SET.FIL, record 1. Sample input files are also provided in Examples 1 and 2. This file is read by all three codes. The description of the data file has been divided into four segments for ease of understanding. The actual data file must be a continuous file.

5.3.1 Namelist Segment

The Namelist Segment in Figure F.2 specifies the namelists in their required order in the file defined in record 1 of "SET.FIL". The first statement of the namelist(i.e., $SIZE) must have the $ starting in column 2. Entries following the first statement also should not begin in column 1. The namelist is ended with a $END statement; again, the $ must be in column 2. The namelist statement is used to change only those parameters that will be different from the default settings. As discussed below, some of the parameters necessarily need to be defined in the respective namelists while others may use the default values. The general format for input of the parameters is:

=

or if the parameter is an array:

<array name> = element 1, element 2, ...

Further information concerning NAMELIST statements can be found in the VAX FORTRAN Manual [F5]. Once a correctly-formatted file has been generated, subsequent input files can be generated from copies of this file.

The first namelist, SIZE, contains two necessary parameters, NEQ and NRTN, and three optional parameters, MESHPTS, XLOW, and XHIGH. NEQ defines the number of different species that will be tracked, whereas NRTN defines the number of chemical reactions in which the species will participate. The parameters NEQ and NRTN will also define the lengths of Segments 1 and 2. Errors may result if there are more species in segments 1 and/or 2 than specified by NEQ , or if there are more or less reactions in segment 1 than specified by NRTN. When MESHPTS, the number of uniformly-spaced points, is zero, the code runs in a pure reaction mode and no transport is used. XLOW and XHIGH, the lower and upper spatial boundary values, are only used when MESHPTS is greater than zero.

The second namelist, STATE, defines the state of the system in terms of some of the physical parameters, namely VL, VG, TEM, TEMR, DSRATE, and DHRATE. The liquid and gas volumes, represented by the parameters VL and VG, are ignored if the problem is single phase (i.e., VG = 0.). When the system is two-phase, the reaction rates can be input as Henry's Law coefficients, (see Section "Segment 1"). The temperature parameters, TEM and

TEMR, refer, respectively, to the temperature at which the Arrhenius expressions for the rate constants are to be evaluated, and the temperature at which the rate constants were measured. The two dose rates, DSRATE and DHRATE, allow for mixed radiation types or for different dose rates of the same radiation type (mainly in a two-phase system.)

The namelist STATE is also where the run-time parameters, namely TOUT, TSTEP, TFINAL, and MULTIME are determined. TOUT is the first time to be evaluated, and is assumed by the sensitivity analysis to be zero. TSTEP defines the method (i.e., additive or multiplicative) and size of the time steps to be used in the computer run. If TSTEP is less than zero, the step is additive:

$$T_{next} = T_{last} - TSTEP*MULTIME$$

As its name implies, MULTIME is used as a multiplier for the time stepping performed by MITIRAD. If TSTEP is greater than zero, the step is multiplicative:
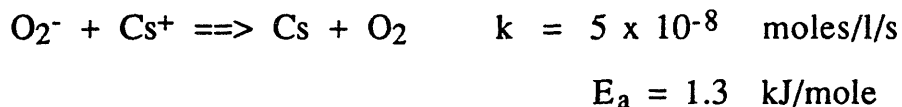
$$T_{next} = T_{last} * MULTIME$$

TSTEP also defines the first time step to be taken by the solver. For complicated problems, the absolute value of TSTEP should be small (often as small as 1.0×10^{-5}.)

The last namelist in this section of the file, NAMES, contains the names of the various species being tracked and identifies the concentration to be used as the response variable. The position of the particular species in the array ARS must be noted since this sequencing is used when setting up the chemical reactions in the next segment. Species names that contain a "G" can optionally be interpreted as gaseous species, and their partial pressures are given in the output (see listing of MITIRAD for details on invoking this option). There must be NEQ names defined in the array ARS. The last entry in the namelist is the integer variable IA. This integer corresponds to the species number, defined in the order of array ARS, to be used as the response variable in the sensitivity analysis.

5.3.2   Segment 1

Segment 1 contains the reaction information and must begin immediately after the $END statement of the namelist NAMES. Segment 1 is best explained through the use of an example. Consider the input for the following chemical reaction:

$$O_2^- + Cs^+ ==> Cs + O_2 \qquad k = 5 \times 10^{-8} \quad moles/l/s$$
$$E_a = 1.3 \quad kJ/mole$$

If namelist NAMES is set up as follows:

$NAMES

$$ARS = Cs, O_2^-, Cs^+, O_2$$

$END

then Segment 1 would be:

```
1   2 3    1 4     5.D-8
1.3
```

The first number that appears in the first record of Segment 1 is the reaction number; this number is arbitrary but may be used to group sets of reactions. The second and third numbers (i.e., 2 and 3) are the identifiers for the reactants. This example gives only two reactants, but blank space must be left in the record for a third reactant that MITIRAD will attempt to read. The next two numbers (i.e., 1 and 4) identify the products of the reaction. Again, this example gives only two products; blank space must be left in the record for a third and fourth product. These 7 numbers (3 possible reactants and 4 possible products) are right-justified integers (format I3). Starting in column 27, the rate constant is given in double-precision format of D18.8; abbreviations are accepted in this format.

The subsequent record (i.e., "1.3" in the above example) contains the activation energy for the reaction and is read by MITIRAD in the format D18.8. Again, MITIRAD accepts abbreviations of the format as shown in the example. If the system is two-phase, the exchange of species between the liquid and gas phases can be handled using a Henry's Law coefficient. To invoke

this, a sequence of two reactions is needed. The first is the exchange from a liquid to a gas; for this, the rate constant is set to a negative number. The second reaction must be the exchange from a gas to a liquid; for this, the rate constant is set to the Henry's Law constant. MITIRAD will recognize all species with a "G" as part of the name defined by ARS (see namelist NAMES) as gaseous (when VG >0; see namelist STATE) and will compute the partial pressure as part of the output.

The process of defining reactions must be repeated until all reactions specified by NRTN (see namelist SIZE) are defined. This segment contains 2*NRTN records. The output file recapitulates the input, but substitutes actual names of the species for the numbers so that the reaction input data may be easily verified.

5.3.3   Segment 2

Segment 2 contains information related to production rates, boundary conditions and diffusion coefficients for the NEQ species (see namelist SIZE). The first two lines of the segment may be used for descriptive information; otherwise, they are left blank. The subsequent five records are used for the first G-value (production rate = $G*DSRATE*1.033 \times 10^{-9}$[1]), second G-value (production rate = $GH * DHRATE*1.033 \times 10^{-9}$), first boundary condition, second boundary condition, and diffusion coefficient of the first species identified in namelist NAMES. Five spaces are required prior to the numbers; this space may be used to annotate the input file (see

---

[1]Conversion factor, 1.033×10^{-9} [100 ev-moles/liter-rad].

sample input file in Appendix C.) If radiation effects are not being used, a blank or zero must be inserted for each G-value. Boundary conditions are pre-programmed into three categories, as defined in Figure F.3. The process (i.e., identification of the G-values, boundary conditions and diffusion coefficients) must continue until all species named in NAMES have been initialized. The order of the information must be the same as the ordering for array ARS (see namelist NAMES). This segment contains (5*NEQ + 2) records (see namelist SIZE).

5.3.4 Optional Segment

A final namelist is included for the optional input of the non-zero initial values. The subroutine YINITIAL (see program listing) assigns an initial value of zero to each parameter. This subroutine may be modified to calculate initial values, as is done in Example 2 of the benchmark problems. Using the namelist method of input avoids having to recompile the entire code to include the modified subroutine YINITIAL; therefore it is recommended that this optional segment be used.

The most common error encountered in the reading of the data file is the result of improper accounting for all of the species and reactions defined by the variable SIZE. This type of error usually occurs when species and/or reactions are added to an existing data file.

The format of the data file is such that it may be annotated, which makes reference to older or numerous computer runs much

easier. If the number of species and/or reactions becomes large, the dimensions of MITIRAD may have to be changed. This is accomplished by changing the parameter statements in all INCLUDE files listed in Appendix B.

5.4 Namelist File

The namelist file is a separate file that is used to change the default parameters that are associated with the numerical and output options of MITIRAD and MITIAD. In general, these parameters are set by the codes and will not need modification. Figure F.4 lists the variables that can be changed and a brief description of each. For many parameters, the LSODE write-up is referenced for a more detailed description. This is indicated in Figure F.4 by an asterisk. FILEMAKER will generate a new default file for the namelist file if requested. Changes to the default values must be made with the editor, as FILEMAKER does not allow for changes in the numerical parameters.

Figure F.2: Format and description of input data file for MITIRAD

RECORD FORMAT   VARIABLE DESCRIPTION   DEFAULT

Namelist Segment

NAMELIST /SIZE/   Sets the size of the problem
$SIZE

I3	NEQ	# of species being evaluated	none, >0
I3	NRTN	# of chemical reactions	ncne, >0
I3	MESHPTS	# of spatial mesh points	0
REAL*8	XLOW	Lower spatial coordinate	0.0D0
REAL*8	XHIGH	Upper spatial coordinate	1.0D0

$END

NAMELIST /STATE/   Contains the state variables
$STATE

REAL*8	VL	Liquid volume (ml)	0.0
REAL*8	VG	Gas volume (ml)	0.0
REAL*8	TEM	Temperature being evaluated (K)	0.0
REAL*8	TEMR	Rate-constant reference Temperature (K)	0.0
REAL*8	DSRATE	Low LET dose rate (rads/s)	0.0
REAL*8	DHRATE	High LET dose rate (rads/s)	0.0
REAL*8	TOUT	Initial time (s)	0.0
REAL*8	TFINAL	Final time (s)	0.0
REAL*8	TSTEP	First time step (s)	0.0
REAL*8	MULTIME	Time multiple (dimensionless)	$10^{1/2}$

$END

NAMELIST /NAMES/   Names of the species in array ARS

$NAMES

A 8	ARS(1)	First species name	blank
"	.	.	"
"	.	.	"
A 8	ARS(NEQ)	Last species name	
INTEGER	IA	Defines the concentration to be used as the response in the sensitivity analysis.	1

$END

Segment 1

Starting in the line immediately following $END from namelist NAMES:

1	A3,2X,7I3, D18.8	RC(1)	reaction #1,3 reactant #'s, 4 product #'s, rate constant (moles/liter/s)	0,7 blanks,0.0
2	D18.8	EA(1)	Activation energy (kJ/mole)	0.0
.	.	.	.	.
.	.	.	.	.
(2*NRTN 0.0	I3,2X,7I3,	RC(NRTN)	Reaction #NRTN, 3 reactant #'s,	0, 7 blanks,
-1)	D18.8		4 product #'s, rate constant (moles/liter/s)	
2*NRTN	D18.8	EA(NRTN)	Activation Energy (kJ/mole)	0.0

Segment 2

(2*NRTN +1)			Blank or descriptive	
(2*NRTN +2)			Blank or descriptive	
...+3	5X,D18.8	G(1)	Low LET G-value for first species (#species/100 ev)	0.0
...+5	5X,D18.8	GH(1)	High LET G-value for first species (#species/100 ev)	0.0

...+6	5X,D18.8	BCL(1)	Left-hand boundary condition for first species	0.0
...+7	5x,D18.8	BCR(1)	Right-hand boundary condition for first species	0.0
...+8	5X,D18.8	DIF(I)	Diffusion coefficient for the first species	0.0
~				
+5*NEQ - 4	5X,D18.8	G(NEQ)	Low LET G-value for last species (#species/100 ev)	0.0
+5*NEQ - 3	5X,D18.8	GH(NEQ)	Right-hand boundary condition for the last species	0.0
+5*NEQ - 2	5X,D18.8	BCL(NEQ)	Left-hand boundary condition for the last species	0.0
+5*NEQ - 1	5X,D18.8	BCR(NEQ)	High LET G-value for last species	0.0
+5*NEQ	5X,D18.8	DIF(NEQ)	Diffusion coefficient for last species (cm^2/s)	0.0

Optional Namelist Segment

$INITIAL   Set non-zero initial values

        Y(...) = ...

$END

Figure F.3 Boundary Condition Options

		BCL	BCR
Constant Concentration	$\dfrac{dC}{dt} = 0;$	2.0	-2.0
Zero Flux	$\dfrac{dC}{dx} = 0;$	0.0	-1.0
Constant Flux	$\dfrac{dC}{dx} = \kappa;$	κ	$-\kappa$

Figure   F.4   Namelist   File   Optionally-Modified   Parameters

Parameter	Description	Default
IOPT*	Notifies LSODE of various options	0
ITASK*	Specifies details of integration step to be taken	1
RTOL*	Relative tolerance parameter	1.D-5
ITOL*	Indicates how tolerances will be handled	1
ATOL*	Absolute tolerance parameter	1.D-16
RWORK*	Real*8 array used by LSODE	none
IWORK*	Integer array used by LSODE	none
IDERV	Flag for output of time derivatives 0 = no derivatives ≥ 1 = derivative of order IDERV computed	0
ISTATE*	Used for input and output of the state of the calculation	1
NPOT	Parameter used in MITIAD to control integration steps so as not to integrate past next requested time point 0 = no control 1 = step to next requested point only 2 = do not step past zero	1
NORSOUT	Flag to stop generating plot file 0 = plot file generated 1 = no plot file	0

NPOINTS	Number of points used to perform trapezoidal integration of total sensitivity.	100
MF*	Flag to indicate to LSODE the type of integration to be performed.	MITIRAD =12 MITIAD = 22
NORSDIA	Flag to generate a diagnostics file from LINTY. = 0　no diagnostics = 1　diagnostics file generated	0

6.0 MITIRAD OUTPUT

The following sections describe the output files generated. The formats of these files are not particularly important to the user since all operations involving these files are transparent to the user.

6.1 Sample Output

The main output from MITIRAD is a sequential file of the input data and the computed results. The name is defined by record 2 of SET.FIL. Appendix F gives the output file from Example 1 of the benchmark problems. The files defined by records 5 and 7 in SET.FIL are essentially identical to this output, except that they contain the output from MITIAD and LINTY. The output from MITIRAD is divided into two parts: a recapitulation of the input data, and the output concentrations at the various time steps. The first part of the output (i.e., the statement of the input information) also includes the code version number and the names of the files used for the input and output. The output of the time steps is controlled by the input parameter TSTEP. The example input uses a time step of $10^{1/2}$ for appearances on a log-log graph of the output. The output in this case does not include any spatial results as it was a pure reaction problem.

The final section gives some statistics on the particular run. These are useful for comparing how hard the code worked in achieving the results for each run. Any error messages from the

solver LSODE will also appear in this file; the LSODE write-up describes the meaning of the error parameters.

6.2 Plot File Output

An optional output (this is essential if sensitivity analyses or graphic output is desired), named in record 3 of file SET.FIL, is available that writes the species names, time steps, and species concentrations calculated by MITIRAD into a specified file. The file defined by record 6 of SET.FIL writes the results of MITIAD in the identical format to this file. The first four records contain the values of NEQ, ITER (the number of time steps), MESHPTS, and ILOGLOG, respectively. The format of the records is (1x,I3). The next NEQ records contain the species names in the format (1x,A8). The subsequent records are divided into segments of (NEQ+MESHPTS+1) records. Each of these segments contain the value of the time step followed by the (NEQ+MESHPTS) species concentration values at the particular time step and location, using the format (1x, E21.14). These segments are repeated ITER times. This provides a file of the output that is easily read into the RS/1 database management system, or graphics software, for further manipulation of the calculated results.

The RS/1 procedure POST_PRE is used to read the plot files into the RS/1 system and generate graphic information. The listing of this procedure is given in Figure F.5.[1] POST_PRE is usually run

---

[1] The details on using the table made by this procedure to generate plots and other analyses using RS/1 are described in the RS/1 users manuals [F6]

after all the codes have been run. If only MITIRAD is run, POST_PRE can still be used to generate plots, but no spline information will be available. The spline information is generated by MITIAD and put into the file defined in record 8 of SET.FIL. Users can get into RS/1 with the command:

$RS1

Graphics are significantly better if the VT240 or the workstation in Tektronics mode are used. If using the VT240, type "VT240" prior to typing RS1; if using the workstation, type "TEK" prior to using RS1. The procedure POST_PRE is invoked by typing CALL POST_PRE at the RS/1 prompt (#). The user has the option of looking at the concentrations or the adjoints (if MITIAD has been run). POST_PRE prompts the user for a table name that is used in RS/1 to store the information. The user may generate a graph of the data and is prompted for the necessary information. Printed output of any graphs generated is also performed if the user so desires. For more detailed manipulations of the data than are provided by POST_PRE, the user is directed to the RS/1 User's Guide [BBN, 1986]

The plot files are also used to pass the necessary output information from MITIRAD to MITIAD, and from MITIAD to LINTY in the file defined in record 6 of SET.FIL.

Figure F.5   RS/1 Procedure to Read Plot File

```
procedure;
ERASE;
AST = INFILE("SET.FIL");
DO I = 1 TO 3;
        A = GETLINE(AST);
END;
DO I = 1 TO 5;
        SF = GETLINE(AST);
END;
CLOSE(AST);
ch  =  infile(A,len,exists,ftype);
I1  =  getline(ch);
I2  =  getline (ch);
I3  =  getline (ch);
I4 = GETLINE (CH);
TB = GETTABLE("NEW TABLE NAME [File Read:" CAT A CAT "]",TRUE);
i1d  =  decode(I1,"I(4)");
i2d  =  decode(I2,"I(4)");
ID3  =  DECODE(I3,"I(4)");
n = i1d + 1;
allocate table(tb) n rows by i2d columns;
SET COL 0 ROW 0 OF TABLE (TB) TO A;
IF ID3 > 0 THEN GO TO MESH;
set col 0 row 1 of table(tb) to "time (s)";
do i = 1 to i1d;
        nme = getline (ch);
        n = i + 1;
        set col 0 row n of table(tb) to nme;
END;
do i = 1 to i2d;
        tme = getline(ch);
        dtme = decode (tme,"e(22,14)");
        set col i row 1 of table(tb) to dtme;
        do j = 1 to i1d;
                conc = getline(ch);
                dconc = decode(conc,"e(22,14)");
                n = j + 1;
                set col i row n of table(tb) to dconc;
        END;
END;
close (ch);
GO TO ENDS;
MESH: TYPE "DO YOU WANT TO LOOK AT A SPECIFIC SPECIES AT ALL NODES,";
        TYPE "OR ALL SPECIES AT A SPECIFIC NODE?";
        TQ = GETTEXT("A = ALL NODES, B = ALL SPECIES");
        IF TQ = "A"  THEN
        BEGIN;
        SPEC = GETNUMBER("WHAT IS THE NUMBER OF THE SPECIES YOU WANT:");
        set col 0 row 1 of table(tb) to "NODE\TIME";
                do I = 1 to i1d;
                        nme = getline (ch);
```

```
                    IF(I=SPEC) THEN
                        set col 1 row 0 of table(tb) to nme;
                    END;
        do I = 1 to i2d;
                    tme  = getline(ch);
                    dtme = decode (tme,"e(22,14)");
                    set col I row 1 of table(tb) to dtme;
                    DO K = 1 TO ID3;
                            do j = 1 to i1d;
                                    conc = getline(ch);
                                    dconc = decode(conc,"e(22,14)");
                                    n = K + 1;
                                    IF J = SPEC THEN
                                    set col i row N of table(tb) to dconc;
                            END;
                    END;
        END;
        END;
        IF TQ = B THEN
        BEGIN;
        NOD = GETNUMBER("WHAT IS THE NODE THAT YOU WANT:");
        set col 0 row 1 of table(tb) to "SPEC\TIME";
        do i = 1 to i1d;
                N = I + 1;
                nme  = getline (ch);
                set col 1 row I of table(tb) to nme;
        END;
        do i = 1 to i2d;
                tme  = getline(ch);
                dtme = decode (tme,"e(22,14)");
                set col i row 1 of table(tb) to dtme;
                DO K = 1 TO ID3;
                        do j = 1 to i1d;
                                conc = getline(ch);
                                dconc = decode(conc,"e(22,14)");
                                n = J + 1;
                                IF K = NOD THEN
                                set col i row N of table(tb) to dconc;
                        END;
                END;
        END;
        END;
ENDS:   GQ = GETTEXT(" DO YOU WANT TO GRAPH THE DATA? [Y/N]");
        IF GQ <> "Y" THEN GO TO LASTEND;
        GNAME = TB CAT "G";
        CONT = 0;
        IF ID3 = 0 THEN
        BEGIN;
                MAKE GRAPH(GNAME) FROM GRAPH("TEMPURE");
                SET TITLE OF GNAME TO "Data From File " CAT A;
MORE:           CONT = CONT + 1;
                DIS COL 0 OF TABLE(TB);
                SPNAM = GETNUMBER("WHICH SPECIES WOULD YOU LIKE TO GRAPH?
BY ROW NUMBER");
```

```
            ADD CURVE TO GRAPH(GNAME) FROM ROW 1 OF TABLE(TB) VS ROW
            SPNAM OF TABLE(TB);
            SET CONNECTED OF CURVE(CONT) OF GRAPH(GNAME) TO "YES";
            SET LABEL OF CURVE(CONT) OF GRAPH(GNAME) TO COL 0 ROW SPNAM
            OF TABLE(TB);
            DISPLAY GRAPH(GNAME) AT (0.5,0.);
            QUEST = GETTEXT("WOULD YOU LIKE TO ADD ANOTHER? [Y/N]") ;
            IF (QUEST = "Y") THEN GO TO MORE;
      END;
      IF ID3<>0 THEN
      BEGIN;
            MAKE GRAPH(GNAME) FROM GRAPH("TEMPNODE");
            SET TITLE OF GNAME TO "Data From File " CAT A;
MORES:              CONT = CONT + 1;
            DIS ROW 1 OF TABLE(TB);
            SPNAM = GETNUMBER("WHICH TIME WOULD YOU LIKE TO GRAPH?
BY COLUMN NUMBER");
            ADD CURVE TO GRAPH(GNAME) FROM ROWS 2 TO LASTROW OF COL 0 OF
            TABLE(TB) VS ROWS 2 TO LASTROW OF COL SPNAM OF TABLE(TB);
            SET SYMBOL OF CURVE(CONT) OF GRAPH(GNAME) TO "=EMPTY";
            SET CONNECTED OF CURVE(CONT) OF GRAPH(GNAME) TO "YES";
            SET LABEL OF CURVE(CONT) OF GRAPH(GNAME) TO ROW 1 COL SPNAM
            OF TABLE(TB);
            SET GRAPHNOTES OF GRAPH(GNAME) TO ROW 0 COL 0 OF TABLE(TB);
            QUEST = GETTEXT("WOULD YOU LIKE TO ADD ANOTHER? [Y/N]");
            IF (QUEST = "Y") THEN GO TO MORES;
      END;
LASTEND:
RLEND:        ERASE;
      TYPE "  END OF POST PRE";
      TYPE "  THE FOLLOWING FILE WAS CREATED:   ",TB;
      TYPE "  THE FOLLOWING GRAPH WAS CREATED:  ",GNAME;
END;
```

6.3 Diagnostics Output

The last file specified in file SET.FIL is a file used to obtain diagnostics information from LINTY. For the most part, LINTY operates with few problems and this file is not needed. In the event that diagnostics of the operation of LINTY are desired, the flag NORDIA must be set to 1 using the namelist file of the previous section, and a file name must be edited into record 8 of SET.FIL. The resulting file can be read into the RS/1 database system using the procedure LDIAGNOSTICS, accessed from RS1 using:

#CALL LDIAGNOSTICS

. This generates a table of the integrands (see equation for total sensitivity) that can be plotted using RS/1.

7.0 BENCHMARK CALCULATIONS

Two benchmark calculations are provided to test the majority of the code operation. The first problem tests the numerics of MITIRAD, and provides a check of the sensitivity portion of the code, MITIAD and LINTY. The second example exercises the partial differential equation portion of the code by including a spatial dependence to a reaction problem. These results should be duplicated when using the code package on new computers or when changes are made to the operating system or programming language software.

7.1   Pure Reaction Problem, Example 1

The particular test case chosen to demonstrate the working of the pure reaction portion is also a numerical benchmark for the code. A similar code [F7] was used to solve this benchmark and produced identical results (to the precision of the calculations.) This benchmark has been recognized [F8, F9] as an excellent test of the ability of a chemical kinetics code in handling stiff systems of equations.

The first example is the burning of a cesium flare taken from Edelson [F9]. The input file is given in Figure F.7. It involves six species undergoing ten different chemical reactions. The output file for the code is given in Appendix F, along with a graphic of the output in Figure F.8. The results are identical to those given in Warner [F8].

Figure F.7 Input Data for Example 1, Pure Reaction Problem, Cesium Flare

```
$SIZE
        NEQ= 6
        NRTN= 10
$END

$STATE
        VL = 0.0
        VG = 0.0
        TEM = 298.
        TEMR = 298.
        DSRATE = 0.0
        DHRATE = 0.0
        TOUT = 0.0
        TFINAL = 1.00D3
        TSTEP = 1.0D-5
        MULTIME = 3.16227766
$END


$NAMES
        ARS = e-, O2-, CS+, CS, CSO2, O2
$END
```

1	2		1	6		0.4
12.6						
2	3	1	4			1.D-12
12.6						
3	6	1	2			1.40D-16
12.6						
4	2	3	4	6		5.0D-8
12.6						
5	4		3	1		3.24D-3
12.6						
6	6	4	4	5	4	1.0D-31
12.6						
7	6	4	5	5	5	1.0D-31
12.6						
8	6	4		5		1.4D-16
12.6						
9	6	6	4	5	6	1.0D-31
12.6						
10	6	6	1	2	6	1.24D-30
18.8						

SPECIES

G-VALUES (#/100ev), BOUNDARY CONDITIONS, DIFFUSION COEFFICIENTS

E-	0.0
	0.0
	0.0
	0.0
	0.0
O2-	0.0
	0.0

```
              0.0
              0.0
              0.0
       CS+    0.0
              0.0
              0.0
              0.0
              0.0
       CS     0.0
              0.0
              0.0
              0.0
              0.0
       CSO2   0.0
              0.0
              0.0
              0.0
              0.0
       O2     0.0
              0.0
              0.0
              0.0
              0.0

$INITIAL
       Y(1) = 1.0D+2
       Y(2) = 5.2D+2
       Y(3) = 6.2D+2
       Y(4) = 1.0D+12
       Y(6) = 3.6D+14

$END
```
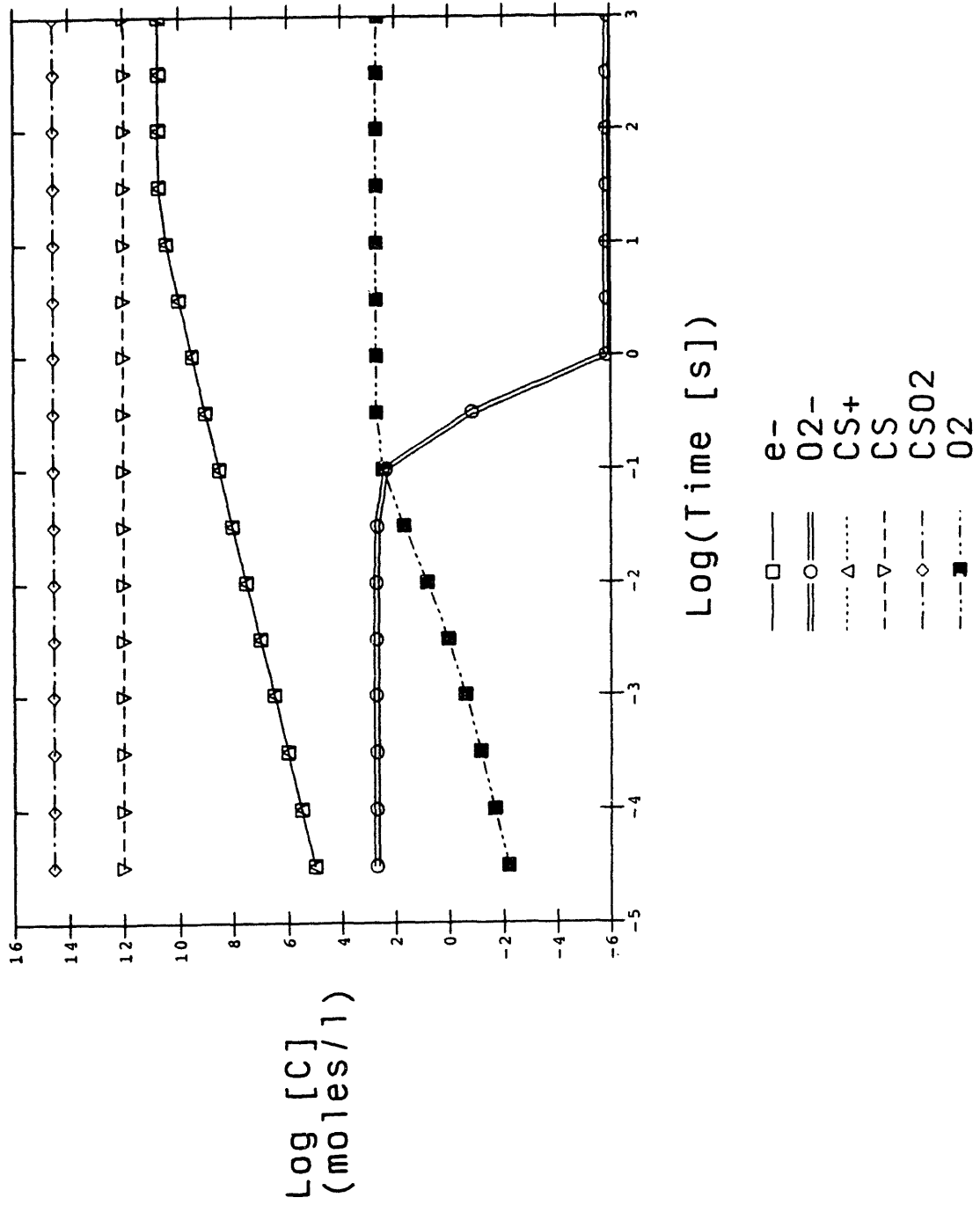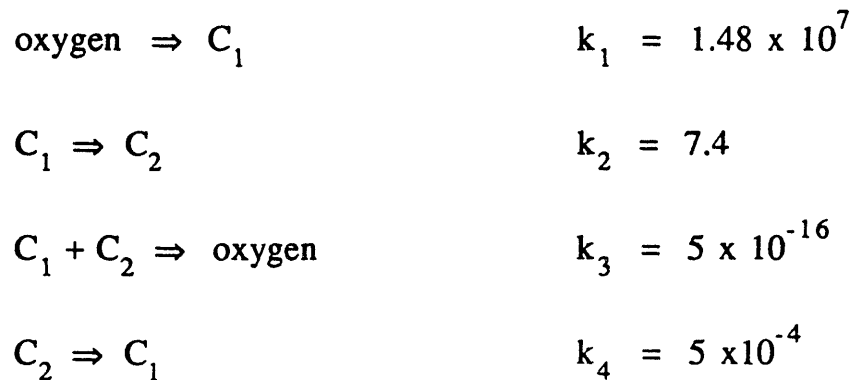
Figure F.8   Graphic Output Generated Using RS/1 for Example 1

7.2 Reaction and Diffusion Problem, Example 2

This example was proposed by Hindmarsh [F10] as a test problem of LSODE utilizing the method-of-lines approach to solving partial differential equations. The problem models simple ozone reactions in the upper atmosphere, 30 to 50 km above the earth. In the equations below, C_1 is singlet oxygen and C_2 is ozone. The two species interact in the following four reactions:

$$\text{oxygen} \Rightarrow C_1 \qquad\qquad k_1 = 1.48 \times 10^7$$

$$C_1 \Rightarrow C_2 \qquad\qquad k_2 = 7.4$$

$$C_1 + C_2 \Rightarrow \text{oxygen} \qquad\qquad k_3 = 5 \times 10^{-16}$$

$$C_2 \Rightarrow C_1 \qquad\qquad k_4 = 5 \times 10^{-4}$$

The species are allowed to diffuse with equal diffusion coefficients of 3×10^{-5} cm^2/s. The boundary conditions are set to zero flux at both boundaries, for both species. The input file for this example is given in Figure F.9.

This example also demonstrates the use of a variable initial condition defined by a modified version of subroutine YINITIAL (see Figure F.10). The equation for the initial conditions is:

$$C_1 = 10^6 * \left[1 - \left[\frac{x}{10} - 4 \right]^2 + \left[\frac{x}{10} - 4 \right]^4 \right]$$

$$C_2 = 10^6 * C_1$$

where x is the distance above 30 km.

The results obtained using MITIRAD are plotted on Figure F.11. Comparison of the numerical values shows that the results are identical.

Figure F.9   Input for Example 2, Diffusion and Reaction

```
$SIZE
 NEQ = 2
 NRTN = 4
 MESHPTS = 50
 XLOW  = 30.0
 XHIGH = 50.0
$END

$STATE
DSRATE = 0.0
DHRATE = 0.0
VL = 0.0
VG = 0.0
TEM = 298.
TEMR = 298.
TOUT = 0.0
TFINAL = 86400.D0
TSTEP = -3600.D0
MULTIME = 3.D0
$END

$NAMES
 ARS='O1','O3'
$END
 1         1        1.48D7
12.6
 2  1      2        7.4
12.6
 3  2  1            5.D-16
12.6
 4  2      1        5.0D-4
12.6
SPECIES  INITIAL CONCENTRATION (MOLES/LITRE)
    G-VALUES LOW/HIGH (#/100EV)
O1  0.0
GH  0.0
BCL 0.0
BCH -1.0
DIF 3.D-5
O3  0.0
GH  0.0
BCL 0.0
BCH -1.0
DIF 3.D-5
```
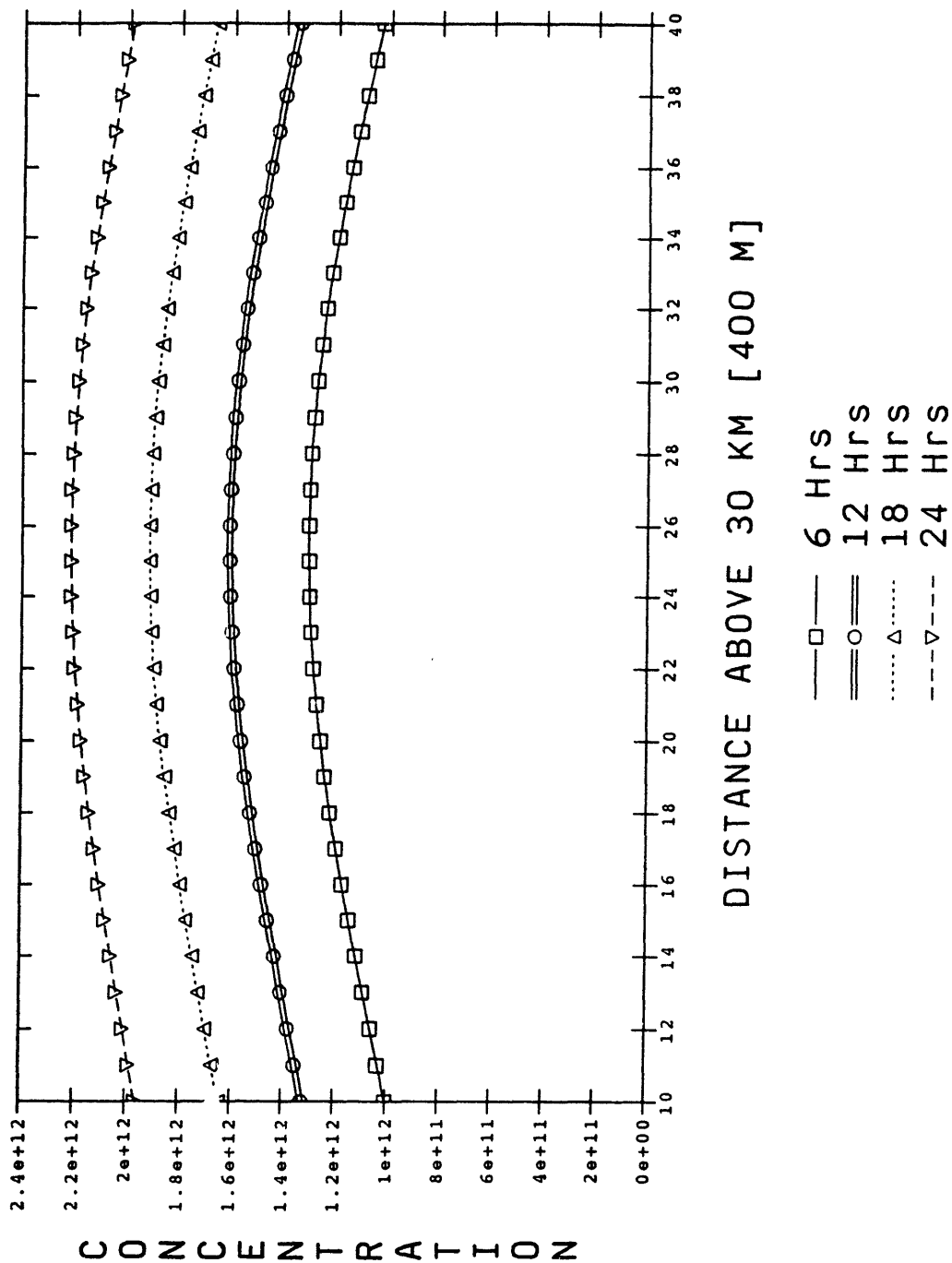
Figure F.10 Modified Subroutine YINITIAL for Example 2.

```fortran
        SUBROUTINE YINITIAL(Y)
C
C
C******************************************************************
C     VERSION: MIT 1.0
C     CODE CUSTODIAN: S. A. SIMONSON
C      DATE OF LAST CHANGE OF SUBROUTINE: 7/27/88
C      PART OF THE SPATIAL_RADIOLYSIS CODE PACKAGE
C      THIS SUBROUTINE IS CALLED BY THE SUBROUTINE READIN
C******************************************************************
C     YINITIAL SETS THE INITIAL VALUES OF ARRAY Y AT NODES 1 TO
C     MESHPTS
C
C******************************************************************
C
C      ASSIGN INITIAL VALUES TO ARRAY Y
C
        INCLUDE 'PARAMETER.BLK'
        INCLUDE 'DIMENSION.BLK'
        NAMELIST /VALUES/ YI
        DIMENSION  Y(ID2),YI(ID2)
C
C      SET-UP FOR HINDMARSH PROBLEM
C
        DO 30 I = 1, MESHPTS
            Z = 30.D0 + DFLOAT(I-1)*XINC
            Z1 = 0.1D0*Z - 4.D0
            Z1 = Z1**2
            GAMZ = 1. - Z1 + 0.5D0*Z1**2
            Y(2*I-1) = 1.0D6*GAMZ
            Y(2*I) = 1.D12*GAMZ
30      CONTINUE
C
C      COMMENT OUT READING FROM NAMELIST
C
C       READ(5,NML=VALUES)
C       DO 10 I=1,ID2
C       Y(I)=YI(I)
C10     CONTINUE
        RETURN
        END
```

Figure F.11   Graphical Presentation of Results Obtained by
                     Hindmarsh and Those Obtained Using MITIRAD

8.0 RECOMMENDATIONS AND LIMITATIONS

As already stated, the best way to start using the code package is through the use of FILEMAKER. A typical sequence of commands to generate the necessary input data, modify SET.FIL, run the codes and generate graphic and tabular output would be:

```
$RUN FILEMAKER        <answer all FILEMAKER questions>
$EDIT SET.FIL         <change output files to make them
                       unique>
$SUBMIT MITIRAD.COM   <run the code package>


<after run finishes....>


$PRINT  output file names in set.fil, separated by commas
$RS1                  <enter RS/1 environment>
#CALL POST_PRE        <generate and print graphs>
#LOGOUT               <leave RS/1>
$LOGOUT               <quit>
```

Unforeseen problems may arise, of course. The best advice for resolving these is to become familiar with the workings of the code and the listings. In general, well-posed physical problems should be relatively error-free. Errors from LSODES will appear in the output files, and they must be checked with the LSODES write-up for resolution.

The flexibility of the MITIRAD package comes at the expense of computer resources. The code can use very large amounts of dynamic memory and will run slowly if the amount of memory available is small. For most problems in radiation chemistry that do not involve transport, the computer resources of the MicroVAX should be sufficient.

The transport aspects of the code have been provided mainly for cases in which the number of species being tracked and the number of nodal points is small (let's say less than 10 species and 50 node points). For more involved transport problems, users should familiarize themselves with the workings of the code and make the necessary modifications to make the code package more efficient. VAX FORTRAN is portable to a Cray computer if the entire package requires significantly greater computer resources.

The sensitivity analysis for transport only looks at the first (sometimes referred to as the left) node. This aspect of the code would have to be altered if sensitivity analyses of the run were needed at interior nodes.

9.0 REFERENCES

F1 Morel, F. M. M. and J. J. Morgan. 1972. *Env. Sci. Technol.*, 6:58

F2 Wolery, T. J. 1983. EQ3NR. A Computer Program for Geochemical Aqueous Speciation-Solubility Calculations: User's Guide and Documentation, UCRL-53414, Lawrence Livermore National Laboratory, Livermore, California.

F3 Hindmarsh, A.C. 1983. "ODEPACK, A Systematized Collection of ODE Solvers," in Scientific Computing, North-Holland Publishing Co.

F4 Gear, C.W. 1971. "Automatic Integration of Ordinary Differential Equations." *Comm. ACM*, 14:176

F5 Digital Equipment Corporation, 1984. Programming in VAX FORTRAN, Maynard, Massachusetts.

F6 BBN Software Products Corporation. 1987. RS/1 Users Guide , Cambridge, Massachusetts.

F7 Carver, M. B., D.V. Hanley, and K.R. Chaplin. 1979. "MAKSIMA-CHEMIST A Program for Mass Action Kinetics Simulation by Automatic Chemical Equation Manipulation and Integration Using Stiff Techniques." AECL-6413, Chalk River Nuclear Laboratories, Chalk River, Ontario.

F8   Warner, D.D.   1977. "The Numerical Solution of the Equations of Chemical Kinetics." *J. Phys. Chem.*, 81(25):2329-34

F9   Edelson, D.   1975. "The New Look in Chemical Kinetics.", *J. Chem. Ed.*, 52:642-44

F10   Hindmarsh, A. C.   1981.   "ODE Solvers for Use With the Method of Lines," in Advances in Computer Methods for Partial Differential Equations. IV, R. Vichnevetsky and R. S. Stapleman, Eds., North-Holland Publishing Co.

Appendix F

Sample Output From MITIRAD

```
* ═══════════════════════════════════════════════ *
*              MITIRAD CODE PACKAGE OUTPUT         *
*              MITIRAD VERSION: MIT  1.0           *
  ───────────────────────────────────────────────
```

                                    INPUT
_____

```
        INPUT FILE NAME                 = CESIUM.FLA
        OUTPUT FILE NAME                = CM1.OUT
        PLOT FILE NAME                  = CP1.OUT
        NAMELIST FILE                   = RUN.NAME


        NUMBER OF CHEMICAL REACTIONS    = 10
        NUMBER OF SPECIES BEING EVALUATED = 6
        NUMBER OF UNIFORM MESH POINTS   = 0
        LEFT COORDINATE                 = 0.000D+00
        RIGHT COORDINATE                = 0.100D+01
        LIQUID VOLUME                   =   0.00000D+00
        GAS VOLUME                      =   0.00000D+00
        TEMPERATURE                     =   0.29800D+03
        REFERENCE TEMPERATURE           =   0.29800D+03
        LOW LET DOSE RATE               =   0.00000D+00
        HIGH LET DOSE RATE              =   0.00000D+00
        ABSOLUTE TOLERANCE              =   0.10000D-14
        RELATIVE TOLERANCE              =   0.10000D-04
        FINAL TIME EVALUATION           =   0.10000D+04
        TIME STEP                       =   0.10000D-04
        TIME MULTIPLE                   =   0.31622D+01
```
_____

CHEMICAL REACTIONS, RATE CONSTANTS, AND ACTIVATION ENERGIES

	REACTIONS				RATE CONSTANT	ACTIVATION ENERGIES
1	O2-			>e- O2	0.40D+00	0.13D+02
2	CS+	e-		>CS	0.10D-11	0.13D+02
3	O2	e-		>O2-	0.14D-15	0.13D+02
4	O2-	CS+		>CS O2	0.50D-07	0.13D+02
5	CS			>CS+ e-	0.32D-02	0.13D+02
6	O2	CS	CS	>CSO2 CS	0.10D-30	0.13D+02
7	O2	CS	CSO2	>CSO2 CSO2	0.10D-30	0.13D+02
8	O2	CS		>CSO2	0.14D-15	0.13D+02
9	O2	O2	CS	>CSO2 O2	0.10D-30	0.13D+02
10	O2	O2	e-	>O2- O2	0.12D-29	0.13D+02

LOW LET G-VALUES	HIGH LET G-VALUES	LOWER BOUND	UPPER BOUND	DIFFUSION COEFFICIENT

```
e-            0.00D+00    0.00D+00    0.00D+00    0.00D+00    0.00D+00

02-           0.00D+00    0.00D+00    0.00D+00    0.00D+00    0.00D+00

CS+           0.00D+00    0.00D+00    0.00D+00    0.00D+00    0.00D+00

CS            0.00D+00    0.00D+00    0.00D+00    0.00D+00    0.00D+00

CSO2          0.00D+00    0.00D+00    0.00D+00    0.00D+00    0.00D+00

02            0.00D+00    0.00D+00    0.00D+00    0.00D+00    0.00D+00
e-            0.10D+03
02-           0.52D+03
CS+           0.62D+03
CS            0.10D+13
CSO2          0.36D+15
02            0.00D+00
```

OUTPUT

CONCENTRATIONS AT TIME = 0.0000D+00

```
    e-     =    0.100000D+03 **     02-    =    0.520000D+03 **
    CS+    =    0.620000D+03 **     CS     =    0.100000D+13 **
    CSO2   =    0.360000D+15 **     02     =    0.000000D+00 **
NO. STEPS =       0
    LOW DOSE RATE (RAD) =  0.000D+00
    HIGH DOSE RATE (RAD) =  0.000D+00
```

CONCENTRATIONS AT TIME = 0.3162D-04

```
    e-     =    0.102555D+06 **     02-    =    0.519993D+03 **
    CS+    =    0.103075D+06 **     CS     =    0.100000D+13 **
    CSO2   =    0.360000D+15 **     02     =    0.661996D-02 **
NO. STEPS =      21
    LOW DOSE RATE (RAD) =  0.000D+00
    HIGH DOSE RATE (RAD) =  0.000D+00
```

CONCENTRATIONS AT TIME = 0.1000D-03

```
    e-     =    0.324084D+06 **     02-    =    0.519979D+03 **
    CS+    =    0.324604D+06 **     CS     =    0.100000D+13 **
    CSO2   =    0.360000D+15 **     02     =    0.212213D-01 **
NO. STEPS =      23
    LOW DOSE RATE (RAD) =  0.000D+00
    HIGH DOSE RATE (RAD) =  0.000D+00
```

```
                     CONCENTRATIONS AT TIME =  0.3162D-03

     e-       =   0.102460D+07 **    O2-     =    0.519930D+03 **
     CS+      =   0.102512D+07 **    CS      =    0.999999D+12 **
     CSO2     =   0.360000D+15 **    O2      =    0.699821D-01 **
NO.  STEPS =    24
     LOW DOSE RATE (RAD) =  0.000D+00
     HIGH DOSE RATE (RAD) =  0.000D+00
```

```
                     CONCENTRATIONS AT TIME =  0.9999D-03

     e-       =   0.323978D+07 **    O2-     =    0.519750D+03 **
     CS+      =   0.324030D+07 **    CS      =    0.999997D+12 **
     CSO2     =   0.360000D+15 **    O2      =    0.250046D+00 **
NO.  STEPS =    27
     LOW DOSE RATE (RAD) =  0.000D+00
     HIGH DOSE RATE (RAD) =  0.000D+00
```

```
                     CONCENTRATIONS AT TIME =  0.3162D-02

     e-       =   0.102446D+08 **    O2-     =    0.518922D+03 **
     CS+      =   0.102451D+08 **    CS      =    0.999990D+12 **
     CSO2     =   0.360000D+15 **    O2      =    0.107769D+01 **
NO.  STEPS =    31
     LOW DOSE RATE (RAD) =  0.000D+00
     HIGH DOSE RATE (RAD) =  0.000D+00
```

```
                     CONCENTRATIONS AT TIME =  0.9999D-02

     e-       =   0.323948D+08 **    O2-     =    0.513747D+03 **
     CS+      =   0.323953D+08 **    CS      =    0.999968D+12 **
     CSO2     =   0.360000D+15 **    O2      =    0.625262D+01 **
NO.  STEPS =    36
     LOW DOSE RATE (RAD) =  0.000D+00
     HIGH DOSE RATE (RAD) =  0.000D+00
```

```
                     CONCENTRATIONS AT TIME =  0.3162D-01

     e-       =   0.102435D+09 **    O2-     =    0.473528D+03 **
     CS+      =   0.102435D+09 **    CS      =    0.999898D+12 **
     CSO2     =   0.360000D+15 **    O2      =    0.464718D+02 **
NO.  STEPS =    44
     LOW DOSE RATE (RAD) =  0.000D+00
     HIGH DOSE RATE (RAD) =  0.000D+00
```

```
                     CONCENTRATIONS AT TIME =  0.9998D-01

     e-       =   0.323881D+09 **    O2-     =    0.222354D+03 **
     CS+      =   0.323881D+09 **    CS      =    0.999676D+12 **
     CSO2     =   0.360000D+15 **    O2      =    0.297644D+03 **
```

```
NO. STEPS =    54
    LOW DOSE RATE (RAD) =  0.000D+00
    HIGH DOSE RATE (RAD) =  0.000D+00
```

```
                    CONCENTRATIONS AT TIME =  0.3162D+00

    e-       =   0.102372D+10 **   O2-      =   0.140039D+00 **
    CS+      =   0.102372D+10 **   CS       =   0.998976D+12 **
    CSO2     =   0.360000D+15 **   O2       =   0.519840D+03 **
NO. STEPS =    91
    LOW DOSE RATE (RAD) =  0.000D+00
    HIGH DOSE RATE (RAD) =  0.000D+00
```

```
                    CONCENTRATIONS AT TIME =  0.9998D+00

    e-       =   0.323048D+10 **   O2-      =   0.145215D-05 **
    CS+      =   0.323048D+10 **   CS       =   0.996770D+12 **
    CSO2     =   0.360000D+15 **   O2       =   0.519917D+03 **
NO. STEPS =   189
    LOW DOSE RATE (RAD) =  0.000D+00
    HIGH DOSE RATE (RAD) =  0.000D+00
```

```
                    CONCENTRATIONS AT TIME =  0.3161D+01

    e-       =   0.100823D+11 **   O2-      =   0.145406D-05 **
    CS+      =   0.100823D+11 **   CS       =   0.989918D+12 **
    CSO2     =   0.360000D+15 **   O2       =   0.519721D+03 **
NO. STEPS =   203
    LOW DOSE RATE (RAD) =  0.000D+00
    HIGH DOSE RATE (RAD) =  0.000D+00
```

```
                    CONCENTRATIONS AT TIME =  0.9997D+01

    e-       =   0.288697D+11 **   O2-      =   0.145310D-05 **
    CS+      =   0.288697D+11 **   CS       =   0.971130D+12 **
    CSO2     =   0.360000D+15 **   O2       =   0.519108D+03 **
NO. STEPS =   221
    LOW DOSE RATE (RAD) =  0.000D+00
    HIGH DOSE RATE (RAD) =  0.000D+00
```

```
                    CONCENTRATIONS AT TIME =  0.3161D+02

    e-       =   0.524575D+11 **   O2-      =   0.144800D-05 **
    CS+      =   0.524575D+11 **   CS       =   0.947542D+12 **
    CSO2     =   0.360000D+15 **   O2       =   0.517223D+03 **
NO. STEPS =   242
    LOW DOSE RATE (RAD) =  0.000D+00
    HIGH DOSE RATE (RAD) =  0.000D+00
```

CONCENTRATIONS AT TIME = 0.9997D+02

```
    e-      =   0.553230D+11 **    02-     =   0.143164D-05 **
    CS+     =   0.553230D+11 **    CS      =   0.944677D+12 **
    CSO2    =   0.360000D+15 **    02      =   0.511373D+03 **
NO. STEPS =   268
    LOW DOSE RATE (RAD) =  0.000D+00
    HIGH DOSE RATE (RAD) =  0.000D+00
```

CONCENTRATIONS AT TIME = 0.3161D+03

```
    e-      =   0.553240D+11 **    02-     =   0.138107D-05 **
    CS+     =   0.553240D+11 **    CS      =   0.944676D+12 **
    CSO2    =   0.360000D+15 **    02      =   0.493312D+03 **
NO. STEPS =   279
    LOW DOSE RATE (RAD) =  0.000D+00
    HIGH DOSE RATE (RAD) =  0.000D+00
```

CONCENTRATIONS AT TIME = 0.9996D+03

```
    e-      =   0.553240D+11 **    02-     =   0.123261D-05 **
    CS+     =   0.553240D+11 **    CS      =   0.944676D+12 **
    CSO2    =   0.360000D+15 **    02      =   0.440282D+03 **
NO. STEPS =   283
    LOW DOSE RATE (RAD) =  0.000D+00
    HIGH DOSE RATE (RAD) =  0.000D+00
```

CONCENTRATIONS AT TIME = 0.1000D+04

```
    e-      =   0.553240D+11 **    02-     =   0.123253D-05 **
    CS+     =   0.553240D+11 **    CS      =   0.944676D+12 **
    CSO2    =   0.360000D+15 **    02      =   0.440253D+03 **
NO. STEPS =   283
    LOW DOSE RATE (RAD) =  0.000D+00
    HIGH DOSE RATE (RAD) =  0.000D+00
```

RUN STATISTICS

```
REQUIRED RWORK SIZE    =    112
IWORK SIZE             =     26
NUMBER OF STEPS        =    283
# OF FUNC.- EVALS.     =    365
# OF JACOB.- EVALS     =     50
```

ERROR HALT...ISTATE = 2

BIOGRAPHICAL INFORMATION

Scott Arthur Simonson was born in Iron Mountain Michigan on April 30, 1959. At age three he was moved to Hoffman Estates Illinois where he spent the remainder of his time until he entered the University of Illinois. He received both Bachelors and Masters degrees in Nuclear Engineering in 1981 and 1983, earning Tau Beta Pi honors and the Northwest College and Associated Universities Fellowship in Nuclear Waste Management. The subject of his Masters Degree was the modeling of nuclear waste glass degradation under nuclear waste repository conditions. Upon completion of his graduate degree at the University of Illinois he took employment with Battelle Pacific Northwest Laboratory in Richland, Washington in 1983. He worked for Battelle for approximately three years, investigating various phenomena related to the degradation of materials to be used in nuclear waste repositories. In 1985 he received the Oak Ridge Associated Universities Fellowship in Nuclear Waste Management and acceptance to the Massachusetts Institute of Technology wherein he completed his Ph.D.

September, 1988