REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

AD-A252 577

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
1991	~~THESIS~~/DISSERTATION

1

Transient Fission Gas Behavior in Uranium Nitrude Fuel Under Proposed Space Applications

5. FUNDING NUMBERS

6. AUTHOR(S)

Daniel Lee Deforest, Captain

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

AFIT Student Attending: Texas A&M University

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/CI/CIA- 92-001D

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFIT/CI
Wright-Patterson AFB OH 45433-6583

10. SPONSORING MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for Public Release IAW 190-1
Distributed Unlimited
ERNEST A. HAYGOOD, Captain, USAF
Executive Officer

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

DTIC
ELECTE
JUL 0 9 1992
S A D

92-17913

92

14. SUBJECT TERMS	15. NUMBER OF PAGES
	246
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

TRANSIENT FISSION GAS BEHAVIOR IN URANIUM NITRIDE FUEL UNDER PROPOSED SPACE APPLICATIONS

A Dissertation

by

DANIEL LEE DEFOREST
Captain, USAF
1991

Awarded PhD, Nuclear Engineering
Texas A&M University

Pages: 246
References: 126

In order to investigate whether fission gas swelling and release would be significant factors in a space based nuclear reactor operating under the Strategic Defense Initiative (SDI) program, the finite element program REDSTONE was developed to model 1-D, spherical geometry diffusion equations describing transient fission gas behavior in a single uranium nitride fuel grain. The equations characterized individual bubbles,, limiting calculations to low temperatures, low burnups, or both. Instabilities in the bubble radii calculations forced additional constraints limiting the bubble sizes to minimum and maximum (equilibrium) radii.

Validity was checked against analytical solutions for internal consistency and against experimental studies for external agreement. These checks indicated REDSTONE was satisfactorily accounting for intragranular fission gas swelling, but other non-gaseous sources accounted for up to 1.5% additional swelling.

Calculations were performed to determine the swelling and gas release of a hypothetical SDI scenario, involving 10 years of low power operation at 10 kW followed by a 30 second rise to 1 MW and maintained for 15 minutes. The fuel pellet diametral increase remained less than 1% and release was negligible. Higher temperatures were used to see if operating parameter uncertainties would cause significantly different results. A significant shift to larger bubbles and larger swelling was indicated near 1800°K.

Poorly defined property values were varied and results compared. The largest effects were observed from the irradiation enhanced diffusion coefficient and the resolution parameter. The resolution parameter uncertainty had the largest potential effect on results.

PRINCIPAL REFERENCES

A.C. Marshall and B. McKissock, Report SAND-89-1635C (Sandia National Laboratories, Albuquerque NM, 1989).

R.F. Hilbert , V.W. Storhok, W. Chubb and D.L. Keller, in: Fast Reactor Fuel Element Technology, ed. R. Farmakes (American Nuclear Society, La Grange Park IL 1971) 753.

D.R. Olander, Fundamental Aspects of Nuclear Reactor Fuel Elements, Available NTIS No. TID-26711-P1 (US Energy Research and Development Administration, Springfield VA, 1976).

E.E. Gruber, J. Appl. Phys. 38 (1967) 243.

M.R. Hayns and M.H. Wood, J. Nucl. Mater. 59 (1976) 293.

S. Chandrasekhar, Rev. of Mod. Phys. 15 (1943) 1.

A.D. Brailsford and R.B. Bullough, in: Physical Metallurgy of Reactor Fuel Elements, eds. J.E. Harris and E.C. Sykes (The Metals Society, London, 1975) 148.

G.W. Greenwood, A.J. Foreman and D.E. Rimmer, J. Nucl. Mater. 4 (1959) 305.

R.S. Nelson, J. Nucl. Mater. 25 (1968) 227.

D.S. Burnett, Finite Element Analysis: From Concepts to Applications (Addison-Wesley Publishing Co., Reading MA, 1987).

Accesion For
NTIS CRA&I
DTIC TAB
Unannounced
Justification
By
Distribution /
Availability
Dist | Avail and / or Special
A-1

INSPECTED
DTIC

TRANSIENT FISSION GAS BEHAVIOR IN URANIUM NITRIDE FUEL UNDER PROPOSED SPACE APPLICATIONS

A Dissertation

by

DANIEL LEE DEFOREST

Submitted to the Office of Graduate Studies
Texas A&M University
in partial fulfillment of the requirement for the degree of

DOCTOR OF PHILOSOPHY

December 1991

Major Subject: Nuclear Engineering

TRANSIENT FISSION GAS BEHAVIOR IN URANIUM NITRIDE FUEL UNDER PROPOSED SPACE APPLICATIONS

A Dissertation

by

DANIEL LEE DEFOREST

Approved as to style and content by:

K. L. Peddicord
(Chair of Committee)

Yassin Hassan
(Member)

John Alred
(Member)

Leonard Cornwell
(Member)

John Poston
(Head of Department)

December 1991

ABSTRACT

Transient Fission Gas Behavior in Uranium Nitride Fuel
Under Proposed Space Applications. (December 1991)
Daniel Lee DeForest, B.S., Baylor University;
M.S., Air Force Institute of Technology
Chair of Advisory Committee: Dr. K.L. Peddicord

In order to investigate whether fission gas swelling and release would be significant factors in a space based nuclear reactor operating under the Strategic Defense Initiative (SDI) program, the finite element program REDSTONE (Routine for Evaluating Dynamic Swelling in Transient Operational Nuclear Environments) was developed to model the 1-D, spherical geometry diffusion equations describing transient fission gas behavior in a single uranium nitride fuel grain. The equations characterized individual bubbles, rather than bubble groupings. This limited calculations to those scenarios where low temperatures, low burnups, or both were present. Instabilities in the bubble radii calculations forced the implementation of additional constraints limiting the bubble sizes to minimum and maximum (equilibrium) radii.

The validity of REDSTONE calculations were checked against analytical solutions for internal consistency and against experimental studies for agreement with swelling and release results. These checks indicated that REDSTONE was satisfactorily accounting for the intragranular fission gas portion of swelling, but that other non-gaseous sources accounted for up to 1.5% additional swelling.

A calculation was performed to determine the swelling and gas release associated with a hypothetical SDI scenario. The SDI scenario involved 10 years of low power operation at 10 kW followed by a 30 second rise to 1 MW which was maintained for approximately 15 minutes. The results indicated that the fuel pellet diametral increase would remain less than 1% and release would be negligible, far below that needed for design failure. This same scenario was run at higher temperatures to see if uncertainties in the operating parameters could cause significantly different results. It was apparent that further investigations into the swelling behavior near 1800°K were needed, as results indicated a significant shift to larger bubbles at this temperature.

In an effort to understand the effect of property uncertainties on the solution, several of the properties were varied and the results compared. The largest effects were observed from changes in the irradiation enhanced diffusion coefficient and the resolution parameter. The resolution parameter uncertainty would have the largest potential effect on results, likely resulting in a reduction in the swelling and a corresponding increase in release.

ACKNOWLEDGEMENTS

I wish to thank Dr. K.L. Peddicord for his support and the opportunities he has given me. I would also like to thank all the members of the Advanced Nuclear Fuels Laboratory, past and present. Their comraderie and humor were integral and positive parts of my time at Texas A&M University. And lastly, I give my deepest thanks to my wife, Paula, who stood by me, put up with me, and managed to finish ahead of me. She made the tough times seem easy and I will forever be in her debt.

TABLE OF CONTENTS

Page

ABSTRACT iii

ACKNOWLEDGEMENTS v

TABLE OF CONTENTS vi

LIST OF TABLES xi

LIST OF FIGURES xii

CHAPTER

I INTRODUCTION 1

1.1 Justification 1
1.2 Initial Indications 3
1.3 Use of Diffusion-Only Equations 4
1.4 Use of Finite Elements 5
1.5 Basic Structure of REDSTONE 5
1.6 Objectives 6

II SCOPING CALCULATIONS 7

2.1 Choice of System Parameters 7
2.1.1 Selection of a Nuclear System 7
2.1.2 Selection of a Nuclear Fuel 7
2.2 Boundary Conditions 8
2.3 Fuel Temperature Profile 9
2.4 Fission Gas Source 10
2.5 Steady-State Diffusion Out of the Grain 11
2.6 Heterogeneous Deposition and Nucleation 12
2.7 Transient Regime 13

III FISSION GAS MODELING IN UN 14

3.1 Historical Overview 14
3.1.1 Modeling Studies 14
3.1.2 Experimental Studies 16
3.1.3 Operational Studies 16
3.2 Fission Gas Behavior Regimes 17
3.2.1 Intergranular Fission Gas 17
3.2.2 Intragranular Fission Gas 18

CHAPTER Page

3.3 Experimental Data on UN 19
3.3.1 Overall Status of UN Property Data 19
3.3.2 Diffusion Coefficients 20
3.3.2.1 Xe Diffusion Coefficient 21
3.3.2.2 Irradiation Enhanced Diffusion 26
3.3.2.3 Bubble Diffusion Coefficients 27
3.3.2.4 Self-Diffusion Coefficients 31
3.3.2.5 Vacancy/Interstitial Diffusion Coefficients 32
3.3.3 Miscellaneous Data 35
3.3.3.1 Thermal Equilibrium Concentrations 35
3.3.3.2 Surface Energy 37
3.3.3.3 Lattice Parameter 38
3.3.3.4 Yield Stress 38
3.3.3.5 Thermal Conductivity 38

IV GOVERNING EQUATIONS AND TERMS 39

4.1 Governing Processes 39
4.2 1-D Diffusion Equation in Spherical Coordinates 43
4.3 Gas Atom and Bubble Concentrations 44
4.3.1 Production by Fission 45
4.3.2 Nucleation 47
4.3.3 *Coalescence and Capture* 52
4.3.4 Resolution 60
4.3.5 Diffusion Loss Due to a Temperature Gradient 63
4.4 Bulk Vacancy and Interstitial Concentrations 65
4.4.1 Production by Fission 65
4.4.2 Recombination 66
4.4.3 Dislocation Interaction 67
4.4.4 Bubble Interaction 68
4.5 Bubble Radius Adjustment 72
4.5.1 Relation to Bubble Concentration Equation 72
4.5.2 Relation to Bubble Vacancy Equation 74
4.5.3 Radius Dependent Properties 76
4.5.3.1 Diffusion Coefficient 76
4.5.3.2 Production Rate 76
4.5.3.3 Bubble Pressure 76
4.5.3.4 Resolution Rate 77

V CONCENTRATION FINITE ELEMENT FORMULATION ... 78

5.1 Generalities 78
5.2 Discretization of the Time Derivative 79
5.3 Development of the Galerkin Equations 81
5.4 Incrementalization of the Galerkin Equations 83
5.5 Discretization of the Galerkin Equations 89
5.6 Determination of Interpolating Functions 90

CHAPTER Page

VI BUBBLE RADII FORMULATION....93

6.1 Generalities....93
6.2 Incrementalization of Individual Processes....95
6.2.1 Resolution Loss....95
6.2.2 Resolution Gain....96
6.2.3 Self-Coalescence Loss....97
6.2.4 Coalescence Loss....98
6.2.5 Coalescence Gain....98
6.2.6 Biased Coalescence Loss....99
6.2.7 Biased Coalescence Gain....100
6.2.8 Source Loss....101
6.2.9 Source Gain....102
6.2.10 Interstitial Capture....102
6.2.11 Vacancy Capture....103
6.2.12 Vacancy Emission....103
6.3 Modification of the Diffusion Term....104

VII PROGRAM REDSTONE ORGANIZATION....106

7.1 Overview....106
7.2 Subroutines....108
7.2.1 ASSEMB....108
7.2.2 ASSMBR....108
7.2.3 ASSMBV....108
7.2.4 DIFFUS....109
7.2.5 ELKFG....109
7.2.6 ELKFR....109
7.2.7 ELKFV....109
7.2.8 EQURAD....109
7.2.9 EXPAND....109
7.2.10 ERROR....110
7.2.11 FRTPAG....110
7.2.12 ICOND....110
7.2.13 INPUT....110
7.2.14 INTARS....110
7.2.15 MTXADD....110
7.2.16 OUTPUT....110
7.2.17 PROP....110
7.2.18 RADII....111
7.2.19 RDCARD....111
7.2.20 RESINP....111
7.2.21 RESSAV....111
7.2.22 RNORM....111
7.2.23 SEMBND....111
7.2.24 SETCON....111

CHAPTER Page

7.2.25 SHAPE 111
7.2.26 SOLVE 112
7.2.27 SWELL 112
7.3 Input 112
7.3.1 Card Number 100 113
7.3.2 Card Number 200 113
7.3.3 Card Number 300 113
7.3.4 Card Number 400 114
7.3.5 Card Number 500 114
7.3.6 Card Number 600 114
7.3.7 Card Number 700 114
7.3.8 Card Number 800 114
7.3.9 Card Number 900 115
7.3.10 Card Number 1000 115
7.3.11 Card Number 1100 115
7.4 Output 115

VIII RESULTS 117

8.1 Validation of Program REDSTONE Internals 117
8.1.1 Steady-State Solution 117
8.1.2 Time-Dependent Solution 118
8.2 Validation of Program REDSTONE Results 122
8.2.1 Additional Constraints 122
8.2.2 Comparison with Experimental Studies 125
8.2.3 Comparison with NASA Results 128
8.2.4 Comparison with JAERI Results 130
8.3 Modeling of the Transient Scenario 132
8.4 Variation of Fuel Temperature 136
8.5 Variation of Property Data 141
8.5.1 Xenon Diffusion Coefficient 142
8.5.2 Surface Diffusion Coefficient 144
8.5.3 Irradiation Enhanced Diffusion Coefficient 146
8.5.4 Surface Energy 149
8.5.5 Resolution Parameter 151

IX SUMMARY AND CONCLUSIONS 154

9.1 Summary 154
9.2 Need for Further Work 159
9.3 Conclusions 161

Page

REFERENCES 163

APPENDIX 1 PROGRAM REDSTONE LISTING 173

A1.1 Main Program 174
A1.2 Include Files 213

APPENDIX 2 SAMPLE INPUT 221

APPENDIX 3 SAMPLE OUTPUT 224

VITA 233

LIST OF TABLES

Table Page

1 SP-100 Specifications....8

2 Isotopes of Krypton and Xenon Released in Fission....22

3 Estimated ^{133}Xe Diffusion Coefficients for UC, UN, and UO_2....24

4 Identification of Processes....45

5 Fuel Pellet Characteristics of NASA Study....129

6 Comparison of NASA Results with REDSTONE Results....129

7 Fuel Pellet Characteristics of JAERI Study....131

8 Comparison of JAERI Results with REDSTONE Results....131

9 Transient Scenario Modeling Parameters....133

10 Transient Scenario REDSTONE Results....133

LIST OF FIGURES

Figure Page

1 Interpolation Functions for Two and Three Nodes 92

2 REDSTONE Program Flow. 107

3 Linear Steady-State Comparison with Analytic Solution 118

4 Linear Transient Comparison with Series Solution 119

5 Change in Interpolation Functions with Middle Node Shift 120

6 Comparison of Results with and without Node Shift 121

7 Graduated Node Density Spacing in Fuel Grain 121

8 Comparison of Results with and without Graduated Node Density 123

9 Swelling and Release Percentages after Onset of Transient 134

10 Pre- and Post-transient Concentration Profiles 134

11 Pre- and Post-transient Radii Profiles 135

12 Pre- and Post-transient Bubble Pressure Profiles 136

13 Volume Swelling vs. Pre- and Post-transient Temperatures 137

14 Post-transient Gas Release vs. Temperature 138

15 Pre-transient Concentration Profiles vs. Temperature 138

16 Post-transient Concentration Profiles vs. Temperature 139

17 Pre- and Post-transient Figures-of-Merit 140

18 Scaled Xenon Diffusion Coefficient Gas Release 142

19 Scaled Xenon Diffusion Coefficient Swelling 143

20 Scaled Xenon Diffusion Coefficient Figures-of-Merit 144

21 Scaled Surface Diffusion Coefficient Gas Release 145

22 Scaled Surface Diffusion Coefficient Swelling 145

Figure Page

23 Scaled Surface Diffusion Coefficient Figures-of-Merit....146
24 Scaled Enhanced Diffusion Coefficient Gas Release....147
25 Scaled Enhanced Diffusion Coefficient Swelling....147
26 Scaled Enhanced Diffusion Coefficient Figures-of-Merit....148
27 Surface Energy Gas Release....149
28 Surface Energy Swelling....150
29 Surface Energy Figures-of-Merit....150
30 Scaled Resolution Parameter Gas Release....151
31 Scaled Resolution Parameter Swelling....152
32 Scaled Resolution Parameter Figures-of-Merit....153

CHAPTER I
INTRODUCTION

1.1 Justification

Proposed devices for use in the Strategic Defense Initiative (SDI) program will require at least a megawatt of power and will operate under stringent requirements. In addition to the harsh environmental requirements of space, the operational requirements will be quite severe. These latter requirements are the ability to survive in a low power 'housekeeping' mode for an extended period, perhaps up to ten years, with the ability to come to full power operation within seconds and maintain this power for a minimum of several minutes. Nuclear power is the most viable power option because of the constraints of power/weight ratio and hands-off operation.

Specific operational parameters for a multi-megawatt, SDI reactor are not publicly available. However, one can imagine a possible scenario where a reactor is sitting in 'housekeeping' mode at just enough power to keep the system functional, perhaps 10 kW electric. Upon command, a rapid increase to full power of 1 MW electric would then be executed, where the system would stay for several minutes. Conceivably, the ramp to full power might occur on the order of 30 seconds with the full power mode lasting for 15 minutes. The full power increase from 10 kW to 1MW in just 30 seconds could place severe thermal transients on the fuel. The high power operation could cause a large inventory of fission gas to coalesce into large bubbles inducing unwanted swelling of the fuel, or it could result in quick release of a large quantity of fission gas producing unwanted pressures in the fuel pin.

Style and format of this dissertation follow the Journal of Nuclear Materials.

One of the primary design constraints of an SDI nuclear system will be the amount of pin diameter increase allowed before the system is determined to have 'failed'. An increase in fuel pin diameter occurs either directly from expansion of the fuel against the cladding, or through excess internal pressure from fission gas which is released to the fuel-clad gap. The former case depends strongly on local conditions of the fuel pin since the fission gas is 'trapped' in the fuel at a specific location. In this case, the fission gas behavior both within the fuel grains (intragranular fission gas) and on the grain boundaries (intergranular fission gas) must be determined. The latter case depends less on local conditions since a large quantity of released fission gas will create open pathways out of the fuel and much of the gas that reaches the grain boundaries will be released to the gap where it mixes with gas released from other fuel pin regions. Here, intragranular fission gas behavior is of primary importance since it is implied by release that intergranular gas does not play a significant role. It is important to realize, however, that the actual behavior of the fuel will be some combination of the above two extremes. Nuclear fuel is usually designed to either release the fission gas to the gap or to accommodate the fission gas in the fuel. In either case, it is clear that the intragranular fission gas behavior must first be known in order to properly design the fuel. The importance of considering intergranular fission gas behavior is determined by the specific design of the fuel used.

The effects of thermally expanding fuel (fission gas swelling excluded) and the associated stresses and strains can be determined from macroscopic properties of the fuel and thus are easier to include in a design. However, the effects from fission gas, requiring knowledge of the microscopic properties of the fuel, are harder to determine. For the current scenario, the initial conditions of long term, low power

operation followed by a severe increase in power by two orders of magnitude are unique to this system. A prediction of possible fission gas behavior in such a system has not been previously reported. Potentially, the fission gas could simply be released from the grain to grain boundaries which would become open avenues to the plenum either through cracks or interconnected grain boundary tunnels developed during the transient. On the other hand, the fission gas could be completely retained by the fuel, i.e. no cracks, and expansion of retained fission gas as bubbles could cause severe swelling of the fuel. The severity of this additional swelling would depend on where the fission gas resided during the transient. Intragranular fission gas would be present either as individual gas atoms or in very small bubbles. Intergranular fission gas tends to form much larger, lower density bubbles than found inside the grain. Grain boundaries are not as 'stiff' as the interior of the grain to expansion by excess bubble pressure. As a result, fission gas retained by the fuel on the grain boundaries will cause the most severe swelling of the fuel. When intergranular fission gas is retained, intragranular fission gas plays a lesser role in direct swelling of the fuel. However, knowledge of intragranular fission gas behavior is important as it is a feed for both intergranular bubbles and gas release. Thus, one must first analyze the intragranular fission gas behavior as a prerequisite to determining the effects from other fission gas processes.

1.2 Initial Indications

Scoping calculations based on the system and scenario as outlined above provided some insight into which processes would need to be included in modeling UN fuel. During the low power phase of the scenario, the fuel temperature appears to be cool enough that fission gas atoms are immobile in the lattice. Also, there is

only a small temperature gradient in the fuel due to the higher thermal conductivity of UN. Bubbles that do form will be from random diffusion with additional effects from the high density of fission gas atoms deposited over a ten year period. It is expected that during this phase the fission gas remains in small bubbles and single atoms and within the fuel grain. During the high power phase of the scenario, the fuel temperature and temperature gradient increases so that gas atoms and bubbles become more mobile. However, the time at full power is short enough that any bubble movement is severely limited. Initial calculations indicated that some release from the fuel grain and some bubble size increase might occur, but not to an extent that would cause fuel pin failure. In summary, initial indications were that the scenario being modeled would not cause large swelling or large release from the fuel grain, and that most of the fission gas would be retained in the grain. However, uncertainties in experimentally measured fuel properties for UN will cause corresponding uncertainties in any fission gas results calculated.

1.3 Use of Diffusion-Only Equations

As a result of scoping calculations, the computer program REDSTONE (Routine for Evaluation of Dynamic Swelling in Transient Operational Nuclear Environments), was written. This program is based on the diffusion equation for a 1-D spherical fuel grain. Effects from thermal gradients are included as additional coalescence terms to the diffusion equation and in some cases as simple release fractions to maintain spherical symmetry. This latter approach appears justified as the expected result is that the time at high power will be too short for significant gas movement to destroy spherical symmetry. This is fortunate since detailed modeling of asymmetric effects would cause the subsequent profiles within the fuel grain to

become non-symmetric and necessitate difficult and unwieldy 3-D modeling of the fuel grain.

1.4 Use of Finite Elements

Program REDSTONE was developed as a finite element code based on the 1-D non-linear diffusion equation in spherical coordinates. To be able to analyze the non-linear equation, the equations had to be incrementalized and solved through iteration. This technique has been used previously for structural mechanics. However, the type of non-linearity present in the diffusion equation used here is altogether different from that found in structural mechanics. In structural mechanics the non-linearities occur both as non-constant material properties and as cross products in the displacement gradients. In the formulation of the diffusion equation used here, the material properties are held constant (i.e. they are assumed not to be a function of fission gas atom concentration) and the non-linearities occur as cross products in the concentrations themselves, not in the concentration gradients. Thus the finite element formulation of the diffusion equation which was used in REDSTONE is fundamentally different from that used in structural mechanics. It appears that this approach has not been reported previously.

1.5 Basic Structure of REDSTONE

The decision was made after some investigation into the formulation of the equations that the modeling of vacancies/interstitials within the fuel matrix and bubble radii should be decoupled from the calculation of bubble concentrations. It was found that inclusion of the vacancy/interstitial equations and bubble radii equations with concentration equations would cause such a high degree of non-linearity that an accurate finite element formulation using the methods applied

here could not be developed. The higher non-linearity would have reduced the concentration equations down to an essentially linear form destroying the benefits of using the method in the first place. As a result, the sequence of calculations in REDSTONE is to first calculate concentration changes, keeping vacancy/interstitial concentrations and bubble radii constant. This is followed by calculation of bulk vacancy and interstitial changes, keeping bubble concentrations and radii constant. After calculation of vacancy/interstitial concentrations, radii are recomputed and the entire process repeated until no additional iterations are required in the individual steps of bubble concentration calculation and radii calculation. Once convergence as described above has occurred, results are output, the time is incremented, and the process starts anew until the end of the problem. The calculation of bulk vacancy/interstitial concentrations and bubble radii are separated as the latter calculation is strictly linear and thus more efficient to calculate alone.

1.6 Objectives

The current work attempts to add insight to the modeling of fission gas within UN and answer some of the questions posed by the previously described scenario. The objectives of this study are:

a) to investigate the modeling of fission gas behavior in uranium nitride (UN)

b) to investigate the modeling of fission gas rate equations via a complete finite element formulation

c) the determination of how much intragranular fission gas migrates to the grain boundaries and the amount of intragranular swelling

d) the effect of uncertainties in the currently available property data

CHAPTER II
SCOPING CALCULATIONS

2.1 Choice of System Parameters

2.1.1 Selection of a Nuclear System

Due to the lack of a current established design for an SDI nuclear system as well as the classified nature of the SDI program, a suitable substitute system needed to be chosen for this study. The most obvious substitute was the SP-100 nuclear reactor proposed for electric power requirements of space vehicles and satellites. This reactor system has an established design and it is conceivable that testing of SDI devices would utilize the SP-100 platform first [1]. Currently, the SP-100 is proposed as the most viable power option for systems requiring 10 kW to 1 MW of electric power. The figure-of-merit for failure of the fuel elements in this reactor has also been established. This limit is one percent diametral increase in the fuel element. Essential physical characteristics of SP-100 in the 1 MW configuration are listed in table 1.

2.1.2 Selection of a Nuclear Fuel

Because of the long lifetime required of the SP-100 reactor, as well as the need to have a compact, lightweight, and non-refuelable system, UN was chosen as the nuclear fuel for this reactor. The primary advantages of UN over UO_2 and UC include the higher density of UN, the much higher thermal conductivity of UN over UO_2, and the expected lower fission gas swelling with burnup of UN. The higher density of UN allows for higher fission densities within the fuel and thus a smaller core (and mass) for producing the required amount of power. The higher thermal

Table 1
SP-100 Specifications

Number of Fuel Elements	3055
Fuel	UN
Enrichment	94%
% Theoretical Density	96%
Cladding	Nb1%Zr
Liner	Rhenium
Fuel Element Length	46.46 cm
Fuel Stack Length	7.62 cm
Cladding OD	0.607 cm
Cladding Thickness	0.055 cm
Liner Thickness	0.010 cm
Fuel OD	0.456 cm

conductivity of UN results in lower temperature gradients in the fuel, thus allowing for higher operating temperatures (i.e. higher powers) and reducing internal thermal strain. The fission gas swelling directly affects the lifetime of the reactor and UN's lower propensity for swelling is principally derived from observations from the early 1970's. Researchers who have compared UO_2 with UN have attributed UN's swelling advantage to smaller thermal gradients [2]. Higher thermal gradients (and resulting higher fuel temperatures) in UO_2 introduce aggravations to swelling such as lenticular void growth, increased fission gas and bubble mobilities, and phase changes. Comparison of swelling data for UC with that of UN has suggested that, at identical temperatures, UC swells at a rate greater than both UN and UO_2 [2]. Fission gas swelling is potentially the largest of the factors which contribute to swelling of nuclear fuel, and is additionally the most complex and difficult to predict. Unfortunately, it is also one of the least characterized areas of UN.

2.2 Boundary Conditions

The first task in performing scoping calculations was to determine the temperature and pressure boundary conditions for the fuel element. The computer

code CENTAR (Code for Extended Non-linear Transient Analysis of Extraterrestrial Reactor System) [3] was used for this purpose. CENTAR was developed to model space reactor systems such as the SP-100. However, it does not contain a detailed fuel pin model. Instead a fuel pin is treated simply as a homogeneous material with a thermal conductivity and pin swelling determined from correlations, which are based on existing irradiation data. Because of this, their accuracy in unusual scenarios, such as the one being looked at here, is unknown. Fortunately, the boundary conditions as calculated in a systems code (i.e. coolant temperature and pressure) are not strong functions of the detailed structure of the fuel pins and thus should provide reasonably accurate input to a detailed fuel modeling code.

Calculations using CENTAR with the transient scenario considered here indicated that at 10 kW electric (200 kW thermal and the minimum needed to keep the system operational) the clad surface temperature was approximately 500°K, while at 1 MW (20 MW thermal) the surface temperature jumped to approximately 1500°K.

2.3 Fuel Temperature Profile

Using the calculated outer fuel pin temperatures, the steady-state heat conduction equation was solved analytically in order to find the centerline temperatures and thermal gradients in the fuel. The thermal conductivity was held constant to obtain the analytic solution.

The temperature increases from surface to centerline (T_{CL} - T_S) were 5°K at T_S= 500°K and 300°K at T_S= 1500°K. The maximum thermal gradient is found at the fuel surface. These values are 38°K/cm at T_S=500°K and 2550°K/cm at

T_s=1500°K. Based on these numbers one would expect that biased effects are not significant at 10 kW, but would need to be included at 1 MW. Since the low power operation will determine the pre-transient condition of the fuel, and because the transient time is so short, the following discussion will concentrate on effects present in the low power, low temperature phase of operation.

2.4 Fission Gas Source

In order to model a transient, the pre-transient condition must be known. The first step is the calculation of the amount of fission gas deposited in the grain at 200 kW thermal power over a ten year life. Dividing 200 kW by the total number of fuel pins in the core (3055), an average thermal rod power of 0.065 kW results. Assuming uniform flux within the fuel, the fission rate density will be 1.642×10^{12} fissions/sec-cm^3. Using a grain diameter of 19 μm, obtained from recent investigations involving SP-100 type fuel [4], one calculates that there are 185.7fissions/sec within the grain. Over a ten year period this translates to 5.850×10^{10} fissions in a grain. Multiplying this by a stable fission gas yield of 0.25fission gas atoms per fission [5], one finds that 1.462×10^{10} fission gas atoms are deposited in the grain over ten years, representing a density of 1.293×10^{20} fission gas atoms/cm^3. If one assumes the gas atoms to be evenly dispersed as single gas atoms within the grain, the calculated distance between two gas atoms is 19.8×10^{-7}cm. Comparing this to the diameter of a xenon atom in the lattice (approximately 3×10^{-7} cm), one sees that local effects around each xenon atom could couple with other xenon atoms which could be highly disruptive to the lattice. If disruption is not expected then it must be assumed that fission gas is lost from the grain or that bubbles nucleate and grow from the deposited gas atoms. Since

disruption of the lattice is not expected, diffusion out of the grain and nucleation must be considered.

2.5 Steady-State Diffusion Out of the Grain

An indication of how much gas would be lost from the grain can be obtained by applying the steady-state diffusion equation to the single atoms deposited in the grain. Taking the boundary condition for diffusion calculations to be $C_g = 0$ at the grain boundary, one finds that the steady-state number of fission gas atoms contained within the grain is 3.22×10^{35} atoms. This number is much larger than the 1.462×10^{10} fission gas atoms deposited over ten years, so it is clear that a steady-state equilibrium is not attained. In fact, additional random walk calculations show that each xenon atom only moves a net distance of only 6.4×10^{-9} μm, far below the dimension of one unit cell in the lattice. Calculation of the homogeneous nucleation rate based on the fission gas concentration and thermal diffusivity indicates that this rate does not significantly alter the single gas atom concentration. Ignoring nucleation for the moment and applying the time-dependent form of the diffusion equation [6] one finds, as indicated above, that there is no significant release of fission gas to the grain boundary. Similarly, calculation of the thermal gradient effects show that biased movement of fission gas atoms is insignificant at this temperature. Scoping calculations thus indicate that once the xenon atoms are deposited, they are virtually 'frozen' in the lattice at low power. However, it should be noted that only thermal diffusion of fission gas was considered in the scoping calculations.

2.6 Heterogeneous Deposition and Nucleation

Although manufactured defects can act as heterogeneous nucleation sites, the primary source of defect sites will be the deposited atoms and fission damage created by fission. Accommodation of heterogeneous effects by some investigators has been done by including semi-empirical modifications to their models which account for fission damage and heterogeneous nucleation [7]. Other investigators have justified the use of a straight homogeneous approach. Unlike UO_2, UN has a greater mobility of electrons which allow the energy transferred by a fission fragment to be deposited over a much larger region [8,9]. As a result, the energy transferred to a single lattice atom is lower and much less likely to impart enough energy to dislodge an atom. As a result, lattice damage from fission will be lower, resulting in fewer heterogeneous nucleation sites. A side effect of this property is that the increased electron mobility also prevents a significant thermal spike from being produced by the fission fragment. As a result, the resolution of bubbles should occur primarily via a Nelson knock-on method [10] versus a sputtering or trapping mechanism. Because of this difference in electrical properties, it would be questionable to simply transfer the UO_2 empirical modifications to UN. Unfortunately, modifications tailored to UN have not been developed due to the lack of appropriate experimental fission damage data. Because fission damage is lower in UN, heterogeneous nucleation will occur primarily through a finite probability that a deposited gas atom comes to rest next to an existing gas atom or bubble. Calculations of this probability involve the ratio of nearest neighbor sites for a specific bubble size to the total number of lattice sites available. Lattice damage from fission is not expected to provide a significant source of bubble nucleation.

2.7 Transient Regime

Similar calculations to those already discussed were performed for the transient regime. These calculations indicated that the transient time at high power will be too short for significant movement of fission gas. Thus, it is not expected that interactions between different sized bubbles will play a role during the higher power phase. However, the predicted amount of gas release is dependent on accurate modeling of intragranular fission gas behavior, which in turn depends on several properties of the material. Uncertainties in properties need to be considered in fuel design to determine acceptable safety margins.

CHAPTER III
FISSION GAS MODELING IN UN

3.1. Historical Overview

Fission gas in nuclear fuels has been of interest to the nuclear community since the early 1960's. The theory of fission gas behavior has developed from a simple model of single gas atoms as solid fission products in pseudo-amorphous fuel [11, 12, 13] to the current state-of-the-art of detailed models describing fission gas behavior both within fuel grains and on grain edges and faces. The latter models attempted to account for the known factors which affect fission gas behavior, i.e. resolution from bubbles, heterogeneous and homogeneous bubble nucleation, bubble coalescence, bubble interlinkage, non-equilibrium bubbles, bubble mobility, nonstoichiometric effects, grain growth, impurities of solid fission products, bubble trapping at dislocations and defects, fuel cracking, fission enhancement of diffusion, etc. Unfortunately, the complex synergistic nature of fission gas behavior and the difficulty of studying the fuel at a microscopic level under operating conditions makes identifying and describing all the contributing and competing variables a monumental task.

3.1.1. Modeling Studies

Although some of the existing fuel modeling codes are based on empirical or semi-empirical correlations to describe fission gas swelling and release, these are only useful within the range of operational conditions used to define the correlations and lack the flexibility to perform exploratory analyses in new areas of interest. Exploratory analysis requires accurate mechanistic modeling codes based as much as possible on known theory. It is for this reason that program REDSTONE was

designed from a mechanistic approach. The theoretical basis of the mechanistic equations used in this work, and for the majority of current state-of-the-art mechanistic fuel modeling codes, dates back to Gruber [14] who, in 1966, first assembled a set of equations describing diffusion and interactive fission gas processes in nuclear fuel. Subsequent researchers extended and adapted Gruber's technique as their understanding about additional fission gas processes increased. Many of the extensions applied by subsequent investigators were situation specific and thus had limited usefulness for exploratory analysis. This resulting increase in complexity and specificity in Gruber's method appeared to continue until 1979, when emphasis seemed to shift from extension of Gruber's method to simplification of the method so it could be used efficiently in large integrated fuel modeling codes. The simplifications were primarily of two types, one involving collapsing the individual bubble groups into a single average bubble group and the other involving utilizing an effective diffusion coefficient for single gas atom calculations to account for all effects. These simplifications were developed and verified using calculations from the more detailed fuel modeling codes. For the most part, all were based on UO_2 fuel properties. Both Matthews and Wood [15,16] and Rest [17] have adapted Gruber's approach to their own codes, TRAFIC (TRAnsient Fuel Interpretative Code) and GRASS (Gas Release And Swelling Subroutine). Both these codes have evolved in similar directions and are representative of the mainstream of the current mechanistic approach to fission gas modeling. The processes considered in determining the effective diffusion coefficient or collapsing of bubble groups may or may not be the same in non-UO_2 fuel materials. The UO_2 simplifications might only be in agreement with the specific cases they were verified against, and thus would not be applicable to other fuel materials and/or operating histories. Before

simplifications for non-UO_2 fuel can be applied, initial modeling of fission gas behavior is best done by reversion to the more detailed equations of Gruber's method where the governing equations are more directly related to experimentally measured properties. This constraint affects the direction of the work here since the fuel of choice is UN, for which there are no detailed calculational studies.

3.1.2 Experimental Studies

As with the modeling developments, experimental studies involving nuclear fuels have concentrated essentially on UO_2. In fact, both have, out of necessity, gone hand-in-hand. As a result, the modeling of UO_2 fuel behavior has advanced to a point that is quite detailed, but which cannot be duplicated for other fuel materials due to a lack of appropriate experimental data. Often, material properties necessary for applying current state-of-the-art models are estimated, usually from corresponding UO_2 data. For UN there is some support for substituting UO_2 properties as some researchers have suggested that certain aspects of fission gas behavior in UN are similar to those of UO_2 [2,18]. In addition to the lack of material property data, the lack of experimental studies involving non-UO_2 fuel materials has left unresolved what processes are and are not important in these materials. Both these problems have to be dealt with in the case of UN. The small amount of UN data available is further clouded by the fact that early investigators had problems in producing and maintaining pure, stoichiometric UN [19,20,21]. The state of the fuel being studied was not always well characterized.

3.1.3 Operational Studies

Studies which have dealt with operating scenarios involving power excursions, in addition to concentrating on UO_2, have generally considered fuel

which is operating initially at high powers (and thus high temperatures) and then experiences an overpower situation to very high temperatures where fuel restructuring is likely to occur. The controlling processes involved in these studies are different than those in the investigation here. Micrographs of UN fuel irradiated at relatively low temperatures have indicated that no restructuring of UN fuel occurs at low temperatures [19]. Predominantly low temperatures are expected and thus little or no fuel restructuring is expected to occur. This will result in a different pre-ramp condition of the fuel than in other studies. Matthews [15] indicates that the form of the intragranular gas at the start of a transient is decisive for subsequent release and swelling. No studies were found where the fuel was first operated for an extended period at the low temperatures used here and then experienced a severe overpower transient. Even if such studies were to exist, it is highly probable that they would have concentrated on the macroscopic fission gas release from a specific fuel design, probably using UO_2, and not on the microscopic fission gas behavior of UN fuel.

3.2 Fission Gas Behavior Regimes

3.2.1 Intergranular Fission Gas

For intergranular (grain boundary) fission gas, modeling approaches have not congealed into any single mainstream method, possibly due to the greater effect initial manufacture and operating conditions have on open porosity and grain boundary dynamics. Early investigators considered the intragranular fission gas to be released once it reached the grain boundary, either immediately or once a specified percentage of the grain boundary had been filled with gas [22,23,24,25,26,27]. It was clear from accumulating experimental data, however,

that the dynamics of fission gas behavior on grain boundaries were much more complicated than these simple approaches suggested. In the 1970's, Tucker [28,29,30] looked at the energies involved in the grain boundary dynamics and was able to outline a detailed treatment of grain face and grain edge bubbles. Fission gas release occurred only after grain face bubbles had interconnected to the grain edges and grain edge bubbles were interconnected to the exterior or central void of the fuel. His approach has gained some acceptance and is the approach taken by Matthews [31]. Unlike the dominance of Gruber's method, however, Tucker's theories are not universally used for mechanistic modeling. At least one investigator has treated the grain boundary gas using diffusional rate equations similar to those used in Gruber's method [32]. Since intergranular fission gas was not modeled in the study here, no further elaboration of intergranular fission gas modeling will be made. If necessary, the amount of fission gas released will be determined using a percentage release criteria.

3.2.2 Intragranular Fission Gas

One of the first investigations into the theory of inert gas behavior in nuclear fuels was performed by Greenwood and Speight [33]. They based their theory on the assumptions that bubbles migrate by surface diffusion, there was no resolution of gas from bubbles, the gas in the bubble behaved according to the ideal gas law, and the bubbles were in equilibrium with the surrounding matrix. They also assumed the material to be a perfect, infinite crystal, and that the bubble distribution could be characterized by a mean bubble radius. Gruber [14] extended this work to eliminate the dependence on a mean bubble radius. He proposed using separate rate equations for single gas atoms and each bubble size. These rate equations included interaction

terms between single gas atoms and bubbles to produce the bubble distribution. Other assumptions used by Greenwood and Speight remained assumptions for Gruber's work. Interaction terms included by Gruber (nucleation and coalescence) were based on colloidal coagulation work performed by Chandrasekhar [34]. The method of Gruber provided a basis that described the detailed bubble distribution, yet was easily adapted to specific cases by simply adding additional terms to the rate equations. The specific implementation of Gruber's method used here will be discussed in detail in the chapter covering governing processes.

3.3 Experimental Data on UN

3.3.1 Overall Status of UN Property Data

Available experimental data on UN, especially for the specific form of fuel proposed for the SP-100, is limited and generally not sufficient to fully characterize the microscopic processes occurring within the irradiated material. Advanced fuels, such as UN, were first considered as better alternatives to UO_2, especially for LMFBR (Liquid Metal Fast Breeder Reactor) technology and possible applications in space [19,35,36,37], in the late 1950's to early 1960's. Early investigations of UN and UC, as well as the mixed fuels (U,Pu)N, and (U,Pu)C were performed in the mid to late 1960's. Unfortunately, work on these alternative fuels was overwhelmed by the data being collected on UO_2 and $(U,Pu)O_2$. As a consequence, investigations using UN dropped off in the early 1970's and only picked up again in the mid-1980's as the United States nuclear space program and SDI (Strategic Defense Initiative) came into the spotlight. Most of the experimental data on UN that were obtained in the 1970's utilized experimental techniques and equipment which were not as refined as today, and the effects of nonstoichiometry and fission damage in

UN were not well understood. Unfortunately, recent investigations utilizing UN have emphasized macroscopic observations (i.e. overall release and pin swelling) of particular fuel designs rather than the microscopic properties which would be useful for generalized fission gas modeling. It is difficult, if not impossible, to decipher the microscopic fission gas behavior from the experimental data in these more recent studies. Overall, experimental data available on UN are not sufficient for building comprehensive mechanistic models of microscopic fission gas behavior in UN fuel. As a result, some of the data necessary must be borrowed from UO_2 or UC.

Most of the experimental property data that was available on UN up to 1989 has been compiled into a database, as well as correlated with variables such as temperature and pressure, at Texas A&M University [38]. However, some of the primary data needed for modeling fission gas behavior is not present in this compilation and had to be gleaned from the few isolated investigations in the literature. Included in this data are the xenon diffusion coefficient, the surface diffusion coefficient, and the fission-enhanced diffusion coefficient. No measurements for some of the required data were found. Included were the surface energy in UN, the fission damage dependence of properties in UN, the effect of stoichiometry on UN properties, and the grain boundary energies of UN.

Where possible, the data used in REDSTONE was taken from the compiled database at Texas A&M. Other data were taken from the isolated investigations or taken from corresponding UC or UO_2 data.

3.3.2 Diffusion Coefficients

The diffusion coefficients of U, N, and the fission gases, as well as the surface diffusion coefficient of UN are absolutely essential to modeling fission gas

behavior. Each of the different types of diffusion coefficients are discussed in the following sub-sections.

3.3.2.1 Xe Diffusion Coefficient

Data for fission gas diffusion, whether in UN or other fuels, is typically taken to be the diffusion coefficient of xenon. This is a logical choice since most of the stable gaseous fission products consist of Xe (see table 2). Although some of the short-lived isotopes have half-lives on the order of the transient times considered here, the contribution of these to the total amount of gas already present in the grain at the beginning of the transient will be negligible. Unfortunately, both the Xe diffusion coefficient and the surface diffusion coefficient in UN are poorly documented.

Only a few measurements of the xenon diffusion coefficient in UN have been made. Melehan and Gates [20] measured the in-pile fission gas release from sintered uranium nitride in 1964. Irradiations and gas collection were performed at several temperatures ranging from 540°C to 1700°C. They noted that the fission gas release rate up to 800°C was independent of temperature. This was attributed to a recoil knockout - evaporation release mechanism. Above 800°C, the release rate followed an Arrhenius behavior described by the equation

$$D_{Xe} = 4 \times 10^{-6} \exp\left(-\frac{4.5 \times 10^{-12}}{kT}\right) \quad cm^2 / s \qquad (1)$$

where the activation energy is 4.5×10^{-12} ergs. Their analysis of the release data was based on the theoretical work of Beck [39]. Although an equivalent sphere approach was used, the size of the spheres was determined strictly from the physical dimensions of the fuel element, not from any analysis of grain size. Also, no

Table 2

Isotopes of Krypton and Xenon Released in Fission [5]

Isotope	Half-Life	Percent Yield
^{131}Xe	Stable	3.2
^{132}Xe	Stable	4.7
^{133}Xe	5.3 day	6.6
^{134}Xe	Stable	6.6
^{135}Xe	9.2 hr	5.5
^{136}Xe	Stable	5.9
Total stable xenon yields		20.4
^{83}Kr	Stable	0.4
^{84}Kr	Stable	0.85
^{85}Kr	Stable(10.6 yrs)	0.15
^{85m}Kr	4.4 hr	1.3
^{86}Kr	Stable	1.4
^{87}Kr	78 min	2.5
^{88}Kr	2.8 hr	3.5
Total stable krypton yields		2.8

correction was performed to account for interactions of gas atoms with bubbles or the effect from nonstoichiometric fuel. Post-irradiation examination of the fuel did not reveal evidence of density changes as would be expected from decomposition of UN. However, since a nitrogen overpressure was not maintained, it would be expected that some degradation of the fuel occurred and that the post-irradiation examination was not able to detect it. Any decomposition would affect the resulting calculated diffusion coefficient.

About the same time, Biddle [40] measured the post-irradiation emission of xenon from uranium carbonitride powders containing varying amounts of carbon. The equivalent sphere model developed by Booth [41] was utilized, with the equivalent sphere size being determined from krypton adsorption measurements onto the surface areas. Biddle's analysis did not account for gas atom interactions and no effort appeared to be made to keep the UN stoichiometric during measurement. Resulting diffusion equations for two different surface areas were

$$D_{Xe} = 3 \times 10^{-8} \exp\left(-\frac{4.0 \times 10^{-12}}{kT}\right) \quad \text{cm}^2/\text{s: Surface area} = 1850 \text{ cm}^2/\text{g} \quad (2)$$

and

$$D_{Xe} = 8 \times 10^{-8} \exp\left(-\frac{5.6 \times 10^{-12}}{kT}\right) \quad \text{cm}^2/\text{s: Surface area} = 1000 \text{ cm}^2/\text{g} . \quad (3)$$

These two results appear markedly different. However, results presented for all the fuel materials seem questionable since values for the carbonitrides exhibit just as wide a range with no observable trend as a function of carbon content.

Measurements of xenon diffusion in single crystals of UN were performed by Oi [42]. The evaluation of the diffusion coefficient was based on the measured release fraction and the surface/volume ratio of the bulk crystals. No allowance was made for gas atom interaction and no attempt was made to maintain stoichiometric UN during measurements. The resulting diffusion coefficient was

$$D_{Xe} = 2.05 \times 10^{-4} \exp\left(-\frac{3.57 \times 10^{-12}}{kT}\right) \quad \text{cm}^2/\text{s} . \quad (4)$$

This equation produces values higher than those found by other investigators. This may be the result of a lack of traps for gas atoms in a single crystal. Oi also mentions that the scatter in their data was considerable.

Work carried out in 1970 at BMI [43] measured fission gas release from 97 percent dense compacted UN which had been neutron activated to produce fissions. The resulting data were analyzed using a Hurst [44] model (see table 3). The Hurst model approximates the effects from fission gas precipitation. A nitrogen overpressure was maintained during analysis to keep the UN stoichiometric. The diffusion coefficient for UN was found to be

Table 3

Estimated ^{133}Xe Diffusion Coefficients for UC, UN, and UO_2

Fuel	Estimated Diffusion Coefficient, cm^2/sec 1400°C	1600°C
UC	3×10^{-12}	8×10^{-11}
UN	2×10^{-13}	2×10^{-12}
UO_2	4×10^{-13}	2×10^{-11}

$$D_{Xe} = 4.65\times10^{-4}\exp\left(-\frac{4.9796\times10^{-12}}{kT}\right) \quad cm^2/s. \qquad (5)$$

Ritzman [19] notes that diffusion coefficients calculated using the Hurst model are several orders of magnitude higher than values previously reported due to the inclusion of precipitation of gas atoms. This may explain the lower values obtained by Biddle [40].

Later diffusion work at NASA [45] used a sweep gas facility to measure in-pile krypton release rates from small UN specimens irradiated to eight percent burnup and at high fission rate density. They correlated their data with predictions of a fission gas release model which included atomic diffusion, fission-enhanced diffusion, and direct recoil. Calculation of any one of these contributors depended on the accuracy of the other two. Because of the form of their equation, they were required to calculate the effective surface area for release. This was accomplished by assuming complete interconnection of grain boundary porosity. No correction was made for gas atom interaction, bubble movement, or biased motion, even though bubbles were clearly evident in post-irradiation analysis. As a result, the calculated diffusion equation represents overall release from the fuel irregardless of species. The diffusional contributors to the overall gas release were modeled with the steady-state solution to the diffusion equation in rectangular coordinates, not

with an equivalent sphere model as used previously. The resulting diffusion equation they obtained for atomic diffusion was

$$D_{Kr} = 5.6 \times 10^{-8} \exp\left(-\frac{1.08 \times 10^{-11}}{kT}\right) \quad cm^2/s \qquad (6)$$

The high activation energy present above may be a direct result of neglecting contributions from bubbles and gas atom traps such as grain boundaries (complete interconnection was assumed). Also, no attempt was made to keep the UN stoichiometric, and the small size of the specimens may have caused normally insignificant processes to interfere with release. In addition, it should be noted that the diffusion equation presented above describes krypton diffusion, not that of xenon. Thus some difference might be expected.

Further work at NASA [46] extended the above analysis to include contributions from bubble diffusion. However, other criticisms of the initial study remained. Their adjusted atomic diffusion equation for krypton was

$$D_{Kr} = 2.4 \times 10^{-10} \exp\left(-\frac{2.60 \times 10^{-12}}{kT}\right) \quad cm^2/s \qquad (7)$$

It is unclear if the inclusion of bubble motion was enough to effect the large changes observed in the diffusion equation. The latter activation energy is certainly consistent with those of other investigators. However, the pre-exponential factor appears to be far too low. Differences between the two studies were not discussed sufficiently to explain the change in diffusion equations.

The dependence of diffusion coefficients on the amount of fission damage present has been used as one explanation of the differences obtained by various investigators. Experiments with UC have indicated a definite dependence of the xenon diffusion coefficient on fission damage [47]. Unfortunately, a similar trend

cannot be applied to UN as the amount of fission damage in the preceding study was not specified and fission damage work with UN has not yet been performed. Fission damage dependence is not included in program REDSTONE.

A review of the various fission gas diffusion equations indicates that the latter results at BMI appear to be the most reliable. Not only were steps taken to preserve the integrity of the UN, but account was made for fission gas traps not included by other investigators. Additionally, the form of the fuel used at BMI (sintered, high density, stoichiometric with some fission damage from neutron activation) was similar to that proposed for space reactors. The diffusion equation used in REDSTONE is that derived from the data at BMI.

3.3.2.2 Irradiation Enhanced Diffusion

Irradiation enhanced diffusion of fission gas in nuclear fuels is not well characterized, even for UO_2. Although Dienes and Damask [48] have looked at this problem theoretically, in practice irradiation enhanced diffusion coefficients are obtained strictly from experiment. In these cases, irradiation enhanced diffusion is used to explain an athermal release of fission gas at lower temperatures. For UO_2, the critical temperature for irradiation enhanced diffusion is around 1000°C [49,50]. For UN, the critical temperature appears to be lower around 800°C [20]. Cornell [51] derived an irradiation enhanced diffusion equation based on experimental observations of bubble sizes and concentrations, and obtained an expression that depended on the fission rate and resolution rate. The irradiation enhanced diffusion coefficient is usually expressed in terms of a constant and the fission rate, $\dot{F}$. Typical values for UO_2 range from [49,52]

$$D_E = 2.0 \times 10^{-29} \dot{F} \text{ to } 2.0 \times 10^{-31} \dot{F} \quad \text{cm}^2/\text{s} \qquad (8)$$

where $\dot{F}$ is the fission rate. For UN, investigators at NASA (National Aeronautical and Space Administration) [46] appear to be the only ones to have calculated an irradiation enhanced diffusion rate. The value they found for UN was

$$D_E = 8.22 \times 10^{-31} \dot{F} \quad \text{cm}^2/\text{s} \qquad (9)$$

This study was also discussed in the section on fission gas diffusion which involved measured krypton releases using sweep gas over small specimens of UN. The results were fit to an equation which included contributions from irradiation enhanced diffusion, atomic diffusion, bubble diffusion, and direct recoil. Although it was mentioned that these contributions were interdependent, evaluation of the irradiation enhanced diffusion coefficient was accomplished at low temperatures where other diffusional processes were negligible. The decrease in calculated irradiation enhanced diffusion from the UO_2 value can be explained from the lower amount of fission damage expected in UN [49,53]. Since no other source for irradiation enhanced diffusion in UN exists, the value obtained by NASA is used in REDSTONE.

3.3.2.3 Bubble Diffusion Coefficients

Three mechanisms have been proposed to explain bubble mobilities: vapor transport, volume diffusion, and surface diffusion. Vapor transport occurs from the evaporation of atoms on the hot side of a bubble and deposition on the cold side. Since surface atoms must break completely away from the surface, high activation energies are present, and thus high temperatures are required. For the low temperatures considered here, vapor transport should be negligible. Experimental evidence for UO_2, where the driving forces and temperatures from temperature

gradients are much higher, indicates that this assumption is valid up to moderately high temperatures [54,55].

Volume diffusion occurs from the bulk movement of atoms in the matrix around the bubble. Calculations on the relative effects between surface diffusion and volume diffusion have shown that only when bubbles are larger than about 1mm in diameter [15] is volume diffusion significant. Since only small bubbles are expected, volume diffusion is not considered to be a factor in the study here, which leaves only surface diffusion to explain bubble mobility. By far the most widely accepted model for bubble mobility is the surface diffusion mechanism. Utilizing the surface diffusion model, the bubble diffusion coefficient is

$$D_n = \frac{3\lambda\Omega}{2\pi R_n^4} D_s \qquad (10)$$

where D_n = bubble diffusion coefficient, D_s = surface diffusion coefficient, λ = atomic spacing, Ω = atomic volume, and R_n = bubble radius.

The surface diffusion coefficient for UN was measured by Weaver in 1968 [56] using helium implanted UN. After annealing, bubble sizes and migration distances up a thermal gradient were measured through replication electron microscopy at temperatures ranging from 1258°K to 1858°K and thermal gradients from 75°K/cm to 880°K/cm. Although their original data were not presented, they did calculate a temperature dependent surface diffusion coefficient of

$$D_s = 1.92 \times 10^3 \exp\left(-\frac{2.93 \times 10^{-12}}{kT}\right) \quad cm^2/s \qquad (11)$$

Also observed was that coalescence of bubbles did not occur as the bubbles traveled up the thermal gradients. Weaver attributed this to the existence of large stress fields surrounding nonequilibrium bubbles. This latter phenomena has been

theoretically investigated by Ronchi [57]. The surface diffusion equation obtained by Weaver is used in program REDSTONE.

If one uses the above equation to calculate a bubble diffusion coefficient for a bubble the size of one xenon atom, one gets a value several orders of magnitude higher than the xenon diffusion coefficient. Even further, Cornell [51] through observations of irradiated fuel postulated that very small bubbles are immobile due to being trapped at dislocations, and thus cannot be described by a diffusion coefficient as given above. However, Nichols [58] looked at much of the data on bubble diffusion and concluded that small bubbles move at rates far greater than those assumed by Cornell, though still not at rates predicted from theory. Lower small bubble mobility is probably a factor since small bubbles in UO_2 have been observed to be faceted rather than spherical [59, 60], and thus would not be governed by the same enthalpies as those of larger, purely surface-diffusion controlled bubbles. This presents a complication, however, since it is general practice to treat all bubbles as being controlled by the same mechanism. So that small bubbles are accurately modeled, bubble diffusion coefficients should exhibit a smooth transition from single xenon atoms to the larger, surface diffusion controlled bubbles. Various theories of bubble mobility, some based on other diffusion mechanisms, have been proposed to explain experimental observations and all claim good agreement with the data [61, 62]. No predominant theory has emerged. Gruber [63] and Chien [32] have adjusted small bubble diffusion coefficients by considering the surface atom jump distance when it is greater than the circumference of the bubble. Ronchi [64] adjusts bubble diffusion coefficients based on a transition from xenon atom to a bubble size large enough so that atomic strains around the bubble are completely relaxed. Because both methods give similar

results, Gruber's approach is applied in REDSTONE due to its easier implementation.

Gruber's approach was to apply a 'rolling stone' model for surface diffusion. Robertson [65] has postulated that this model is a more accurate description of surface diffusion. The classic model of surface diffusion is that it occurs one interatomic distance at a time via the movement of vacancies on the surface. In the 'rolling stone' model, the movement of surface atoms occurs when an atom pops out of the surface and 'rolls' along the surface to an existing vacancy. The distance of the 'roll', i.e. the mean jump distance, will span 10-1000 interatomic distances. Gruber shows that this model produces results consistent with existing data.

To illustrate the approach, a bubble of radius 'R_n' is assumed with a surface atom mean jump distance of λ_s, assumed to be constant and independent of surface orientation and curvature. The net movement of the bubble, λ_n, can be calculated from simple geometry from the arc length, λ_s, and the bubble radius, R_n, i.e.

$$\lambda_n = 2R_n \sin\left(\frac{\lambda_s}{2R_n}\right) \qquad (12)$$

For $\lambda_s << \pi R_n$, the above becomes $\lambda_n = \lambda_s$. For $\lambda_s > \pi R_n$, λ_n is periodic. Because this result yields unrealistic behavior, Gruber assumed for $R_n < \lambda_s/\pi$ that $\lambda_n = 2R_n$. The correction for bubble diffusion can then be represented by

$$D_n^{new} = D_n\left(\frac{\lambda_n}{\lambda_s}\right)^2 \qquad (13)$$

Substituting the appropriate expressions for the correction factor into the above equation gives

$$D_n^{new} = \frac{6\lambda\Omega D_s}{\pi\lambda_s^2 R_n^2}\sin^2\left(\frac{\lambda_s}{2R_n}\right) \qquad R_n > \frac{\lambda_s}{\pi} \qquad (14)$$

$$D_n^{new} = \frac{6\lambda\Omega D_s}{\pi\lambda_s^2 R_n^2} \qquad R_n \le \frac{\lambda_s}{\pi} \qquad (15)$$

The only additional unknown is the jump distance, λ_s. Gruber calculated λ_s by applying his modifications to the data of Gulden [61], which indicated that bubbles smaller than 37 Å appeared to follow an R_n^{-2} dependence rather than the R_n^{-4} dependence expected from surface diffusion. Gruber was able to predict the trends in Gulden's data using surface diffusion and his modifications. He was also able to determine λ_s from the data to be approximately 112 Å. Since data on UN to perform a similar analysis are not available, it is assumed in REDSTONE that λ_s= 112 Å for UN.

3.3.2.4 Self-Diffusion Coefficients

Experimental data on U and N diffusion in UN are more abundant than for fission gas diffusion. The available data have been surveyed and correlated with nitrogen partial pressure, temperature, and grain size in a database recently compiled at Texas A&M University [38]. Since the work involved in compiling the database considered data quality and experiment reliability, it was decided to simply use the correlated values from the Texas A&M University database. An extensive review of the available self-diffusion data will not be presented here. Interested readers are referred to the above reference.

The correlated equations for self-diffusion of U and N in UN are

$$D_U = 2.2146 \times 10^{-11} P^{0.6414} \exp\left(-\frac{1.1029 \times 10^{-12}}{kT}\right) \quad \text{cm}^2/\text{s} \qquad (16)$$

and

$$D_N = 2.251 \times 10^{-5} P^{0.4134} g^{-0.737+2.179\times10^{-4}T} \exp\left(-\frac{2.6527 \times 10^{-12}}{kT}\right) \quad \text{cm}^2/\text{s} \qquad (17)$$

where P = nitrogen partial pressure (atm), T = temperature (K), and g = average grain size (μm).

3.3.2.5 Vacancy/Interstitial Diffusion Coefficients

The diffusion of vacancies and interstitials is poorly documented for nuclear fuels. Previous investigators adopted values from other materials. Because it is the vacancies that cause bubble growth, several investigators who have considered the effects of vacancies have not included interstitials. Further, only a few investigators who have incorporated vacancy effects have published the actual diffusion data used. Additionally, vacancy effects are often incorporated only implicitly by utilizing creep properties of the fuel, not by a set of coupled rate equations. As a result, there is slightly more documentation concerning vacancies than interstitials.

For interstitials, both Hayns [66] and Dienes and Damask [48] use the same value for the migration energy, $Q_i^m = 3.20\times10^{-13}$ ergs, where Q_i^m is the energy of migration for interstitials. Hayns took his value from austenitic steel while Dienes and Damask took theirs from a representative metal 'such as copper'. Since the two values agree, it may be that Dienes and Damask's representative metal was not all copper, but rather a conglomeration of data available. This is a reasonable hypothesis since Dienes and Damask's work predates the others discussed by

14years. Hayns used a pre-exponential factor of 10^{-3}, while Dienes and Damask used a theoretical approach to arrive at 10^{-2}. However, several values needed to arrive at the latter value had to be contrived. Thus, it cannot be automatically assumed that the theoretical value is better than other values for which the origin is unknown. Straalsund and Guthrie [67] assumed a pre-exponential factor of one for all their diffusion equations and used an interstitial migration energy value, reportedly from austenitic steel, which was different from the above, i.e. Q_i^m=2.07 $\times 10^{-13}$ ergs. The reason for the discrepancy in values is unknown. However, they mention that many of the parameters needed are highly uncertain, and thus some of the data involves reasonable estimates. Because it cannot be determined which values best apply to UN, and given the agreement between Hayns, and Dienes and Damask, the diffusion of interstitials in REDSTONE was chosen to be represented by

$$D_i = 10^{-3} \exp\left(-\frac{3.20 \times 10^{-13}}{kT}\right) \quad \text{cm}^2/\text{s}. \qquad (18)$$

None of the above works which considered interstitials were concerned with fission gas modeling. Esteves, et al. [68] explicitly considered vacancies (but not interstitials) in modeling fission gas behavior in UO_2. Their form of the diffusion equation for vacancies was derived by taking the pre-exponential factor from uranium self-diffusion and a vacancy migration energy equal to the vacancy formation energy, i.e.

$$D_v = 4 \times 10^{-7} \exp\left(-\frac{2.44 \times 10^{-12}}{kT}\right) \quad \text{cm}^2/\text{s} \qquad (19)$$

where $Q_v^m = Q_v^f = 2.44 \times 10^{-12}$ ergs. The energy of formation was taken as one-half the activation energy for self-diffusion. The basis for their pre-exponential factor is questionable since Olander [5] notes that

$$D_v = D_U C_v \Omega \ . \qquad (20)$$

Thus, one would not expect the self-diffusion pre-exponential factor and the vacancy pre-exponential factor to be the same. Additionally, it would be expected, as is the case with the other investigators reviewed, that the migration energy would be approximately 80% to 90% of the formation energy. Other investigators took their values from other metals. Dienes and Damask [48] again used values from a 'representative metal'. Their theoretically based pre-exponential factor was one and the vacancy migration was $Q_v^m = 1.28 \times 10^{-12}$ ergs. As will be seen, this value is much lower than others. Because of the previously mentioned uncertainties in the source of Dienes and Damask's data, their values for vacancy diffusion were not considered for use in REDSTONE. Straalsund and Guthrie [67], Hayns [66], and Brailsford and Bullough [69] all use values derived from steel. Straalsund and Guthrie assumed a pre-exponential factor of one, and used a migration energy of $Q_v^m = 2.05 \times 10^{-12}$ ergs. Hayns, and Brailsford and Bullough both used a pre-exponential factor of 0.6. However their migration energies were slightly different with Hayns using $Q_v^m = 2.24 \times 10^{-12}$ ergs, and Brailsford and Bullough using $Q_v^m = 2.08 \times 10^{-12}$ ergs. The discrepancy between these last three migration energies could not be resolved, although it is possible that the value used by Straalsund and Guthrie involves a 'reasonable estimate'.

The vacancy migration energy used in REDSTONE is an average of these three energies. The pre-exponential factor was taken as 0.6, i.e.

$$D_v = 0.6\exp\left(-\frac{2.12\times10^{-12}}{kT}\right) \quad \text{cm}^2/\text{s}. \qquad (21)$$

3.3.3 Miscellaneous Data

Non-diffusional experimental data is also required to adequately model fission gas behavior. The more important ones are discussed below.

3.3.3.1 Thermal Equilibrium Concentrations

The equilibrium concentrations of vacancies and interstitials, although insignificant in an irradiation environment, are needed to specify the boundary conditions of bubbles, dislocations, and grain boundaries. The equilibrium concentrations are governed through thermal emission via

$$C_v^e = \frac{1}{\Omega}\exp\left(-\frac{Q_v^f}{kT}\right) \qquad (22)$$

and

$$C_i^e = \frac{1}{\Omega}\exp\left(-\frac{Q_i^f}{kT}\right) \qquad (23)$$

where Ω = atomic volume, Q_v^f = vacancy energy of formation, and Q_i^f = interstitial energy of formation. As with diffusion data, the energies of formation are poorly documented. This is particularly true for interstitials. Olander [5] notes that, unlike vacancy formation, a variety of interstitial sites are available and thus it is harder to determine the interstitial energy of formation. Again, investigators have used data available from similar metals. Straalsund and Guthrie [67] used a value from austenitic steel, $Q_i^f = 6.9\times10^{-12}$ ergs, while Dienes and Damask [48] used a value

from their 'representative metal', $Q_i^f = 6.4\times10^{-12}$ ergs. Olander notes further that the energy of formation for interstitials is probably higher than that for vacancies in all solids. In practice for nuclear fuels, the equilibrium concentration of interstitials is taken as zero due to the high formation energy [5,48,70,71]. This same practice is used in REDSTONE, i.e. $C_i^e = 0$.

For vacancies, the equilibrium concentration is retained. Hayns [66] and Brailsford and Bullough [69] used the same values from steel, $Q_v^f = 2.56\times10^{-12}$ ergs. Straalsund and Guthrie [67] used energies of formation in the range $Q_v^f = 2.6\times10^{-12}$ to 2.8×10^{-12} ergs. Dienes and Damask [48] used values from their 'representative metal', $Q_v^f = 1.92\times10^{-12}$ ergs. Like their value of the vacancy migration energy, this value is much lower than that used by other investigators. Because of this discrepancy, this value was not considered for use in REDSTONE. As discussed earlier, Esteves, et al. [68] took half of the activation energy for self-diffusion in UO_2 as the formation energy for vacancies [72], i.e. $Q_v^f = 2.44\times10^{-12}$ ergs. Another fission gas modeling work, Ting, et al. [73], modeled UC using a vacancy formation energy provided by DonnerandSchiile[74],$Q_v^f = 2.71\times10^{-12}$ ergs.

All the values above, except for that of Dienes and Damask, fall in the same general range. Since one value does not appear any more applicable than another, the vacancy energy of formation used in REDSTONE is an average of the above values, with the exception of Dienes and Damask's value, i.e. $Q_v^f = 2.60\times10^{-12}$ ergs, with

$$C_v^e = \frac{1}{\Omega}\exp\left(-\frac{2.60\times10^{-12}}{kT}\right) \quad \text{cm}^{-3} . \qquad (24)$$

3.3.3.2 Surface Energy

The surface energy is important for determining bubble sizes and excess pressures. No available data on the surface energy of UN were found. Surface energy data is available concerning the other two nuclear fuels, UO_2 and UC. An early experimental study [75] found the free surface energy of UO_2 to be 640ergs/cm ^{2} and the free surface energy of UC to be 1000 ergs/cm^2. These numbers are very approximate, with stated uncertainties on the order of 20 to 30 percent. The free surface energy of UO_2 was calculated from theory by Benson [76] to be 1030 erg/cm^2. Warner and Nichols [77] estimated the surface energy to be approximately 1000 ergs/cm^2. About the same time, Eberhart [78] found critical surface energies of UO_2 to lie between 600 and 1650 ergs/cm^2 depending on whether an argon or hydrogen atmosphere was present. Later, Maiya [79] performed experiments of the growth of grain boundary grooves and determined an average free surface energy of 626 ergs/cm^2. Clearly, a consistent agreement on the value of free surface energy in UO_2 has not been reached. Most often, the value used in fuel modeling codes is 1000 ergs/cm^2. For UC, reported surface energy values have varied just as much. Initial studies set the free surface energy at 730ergs/cm ^{2} [80] to 780 ergs/cm^2 [81]. However, further work indicated that the free surface energy of UC was as high as 2000 erg/cm^2 [82,83]. Although some fuel modeling codes have used the lower values for UC, the spread of available data suggests that the use of 1000 ergs/cm^2 would not be unreasonable. Since no reported data for free surface energy of UN have been found, and consistent values are not available for either UO_2 or UC, the value for free surface energy of UN used in REDSTONE is 1000 ergs/cm^2.

3.3.3.3 Lattice Parameter

As with self-diffusion, the lattice parameter of UN has also been correlated with temperature at Texas A&M University [38]. The equation

$$a = 4.879 + 3.264 \times 10^{-5} T + 6.889 \times 10^{-9} T^2 \qquad \text{Å} \qquad (25)$$

is used in REDSTONE where T = temperature (K).

3.3.3.4 Yield Stress

The yield stress is used in REDSTONE to determine the minimum sizes of bubbles. The yield stress for UN was approximated from the stress-strain curves of Werner and Blank [84]. Values consistent with the plastic behavior were read from the curves and then correlated with a least squares fit. The resulting correlation is

$$\sigma_y = -9.256 \times 10^{8} T + 1.735 \times 10^{12} \qquad \text{Pa}. \qquad (26)$$

The values produced with such a correlation are very crude. Fortunately, bubble sizes change very little with large changes in pressure near this stress. Thus, inaccuracies in the yield stress should have minimal effect on the program results. The main effect of the yield stress is to keep the bubble sizes from becoming too small.

3.3.3.5 Thermal Conductivity

The thermal conductivity of UN has been correlated with temperature and porosity at Texas A&M University [38]. This correlation was used in determining the temperature profile in the fuel during scoping calculations. The thermal conductivity of porous UN is given by

$$k_p = 1.864 T^{0.0361} \exp(-2.14P) \qquad \text{W / m - k} \qquad (27)$$

where T = temperature (K) and P = volume fraction porosity.

CHAPTER IV
GOVERNING EQUATIONS AND TERMS

4.1 Governing Processes

The equation which is the basis of program REDSTONE, the diffusion equation, can be considered a balance of production and loss rates of individual species, be they gas atoms, gas bubbles, vacancies, or interstitials. Considering first individual gas atoms, a balance of production and losses within the grain typically would consider the following:

RATE OF CHANGE OF CONCENTRATION OF SINGLE GAS ATOMS	=	PRODUCTION BY • FISSION • RESOLUTION FROM INTRAGRANULAR GAS BUBBLES • RESOLUTION FROM INTERGRANULAR GAS BUBBLES • RELEASE FROM CRYSTALLINE DEFECTS	-	LOSS BY • DIFFUSION OUT OF THE GRAIN • NUCLEATION OF GAS BUBBLES • GAS BUBBLE *CAPTURE* • CRYSTALLINE DEFECT CAPTURE • GRAIN GROWTH

(28)

For UN fuel and the operating scenario to be considered here, some of the above processes can be eliminated from consideration. Based on scoping calculations, the amount of intergranular gas would be so small that resolution effects from these bubbles would be negligible. Assuming no grain boundary resolution also provides for the worst case situation where diffusion out of the grain is maximized. Within the UN fuel grain, the effect of crystalline defects on gas atoms will be small, in the case here assumed negligible, due to the lower amount of fission damage expected in UN [9]. The dominance of fission gas bubbles will provide for the major capture defect [85]. The effect of fission damage defects is

somewhat accounted for as the Xe diffusion coefficient used was measured in irradiated, fission damaged material [43]. Loss by grain growth can also be eliminated, since the temperatures of interest during the long term operation are too low for significant grain growth to occur [20,37]. During the short term at high power, the time is too short for any significant grain growth. After eliminating the above terms, a simplified equation can be written:

RATE OF CHANGE OF CONCENTRATION OF SINGLE GAS ATOMS	=	PRODUCTION BY • FISSION • RESOLUTION FROM INTRAGRANULAR GAS BUBBLES	-	LOSS BY • DIFFUSION OUT OF THE GRAIN • NUCLEATION OF GAS BUBBLES • GAS BUBBLE CAPTURE	(29)

Biased effects are implicitly included in the above. Modeling of biased migration effects can likewise be simplified since biased migration would be minimal in UN at low temperatures and at high temperatures the time available severely limits atom and bubble movement. Specific simplifications to biased effects will be discussed in sections dealing with those effects.

For gas bubbles, a similar production-loss equation can be written:

RATE OF CHANGE OF CONCENTRATION OF GAS BUBBLES OF 'N' ATOMS	=	PRODUCTION BY • FISSION • NUCLEATION OR COALESCENCE OF SMALLER BUBBLES • RESOLUTION FROM THE NEXT LARGER INTRAGRANULAR GAS BUBBLES • RELEASE FROM CRYSTALLINE DEFECTS	-	LOSS BY • DIFFUSION OUT OF THE GRAIN • GAS ATOM CAPTURE AND BUBBLE COALESCENCE • RESOLUTION • CRYSTALLINE DEFECT CAPTURE • GRAIN GROWTH	(30)

The same arguments, as for single gas atoms, can be used to eliminate effects from crystalline defects and grain growth. The bubble rate equation can thus be simplified:

RATE OF CHANGE OF CONCENTRATION OF GAS BUBBLES OF 'N' ATOMS	=	PRODUCTION BY • FISSION • NUCLEATION OR COALESCENCE OF SMALLER BUBBLES • RESOLUTION FROM THE NEXT LARGER INTRAGRANULAR GAS BUBBLES	-	LOSS BY • DIFFUSION OUT OF THE GRAIN • GAS ATOM CAPTURE AND BUBBLE COALESCENCE • RESOLUTION

(31)

Vacancies and interstitials can be treated likewise with an overall equation:

RATE OF CHANGE OF CONCENTRATION OF VACANCIES AND INTERSTITIALS	=	PRODUCTION BY • FISSION • RELEASE FROM BUBBLES • RELEASE FROM CRYSTALLINE DEFECTS • RELEASE FROM GRAIN BOUNDARIES	-	LOSS BY • DIFFUSION TO GRAIN BOUNDARIES • *RECOMBINATION* • BUBBLE CAPTURE • CRYSTALLINE DEFECT CAPTURE

(32)

Unlike gas atoms and bubbles, effects from crystalline defects are included as they provide the driving force for bubble growth [5,86]. Matthews and Wood [15] note that the effect of crystalline defects as vacancy sources will be negligible compared to the grain boundary for grains less than 100μm. However, vacancy effects from crystalline effects are important for determining the vacancy/interstitial balance, and thus vacancy emission from defects is included. Release of interstitials from sinks are considered negligible since the equilibrium concentration of interstitials in UO_2, and presumably UN, is negligible [5,87]. Release of vacancies from bubbles and

grain boundaries are not incorporated explicitly, but as boundary conditions when solving equations for the overall concentrations. Thus, the simplified equation for vacancies and interstitials is:

RATE OF CHANGE OF CONCENTRATION OF VACANCIES AND INTERSTITIALS	=	PRODUCTION BY • FISSION	-	LOSS BY • DIFFUSION TO GRAIN BOUNDARIES • RECOMBINATION • BUBBLE CAPTURE • DEFECT CAPTURE

(33)

The rate equation describing bubble vacancy adjustment is a modified version of the simplified bubble rate equation:

RATE OF CHANGE OF CONCENTRATION OF VACANCIES ASSOCIATED WITH GAS BUBBLES OF 'N' ATOMS	=	PRODUCTION BY • NUCLEATION OR COALESCENCE OF SMALLER BUBBLES TIMES THE SUM OF VACANCIES ASSOCIATED WITH CONTRIBUTING BUBBLES • RESOLUTION FROM THE NEXT LARGER INTRAGRANULAR GAS BUBBLES TIMES VACANCIES ASSOCIATED WITH NEXT LARGER BUBBLE • BUBBLE CAPTURE OF VACANCIES	-	LOSS BY • DIFFUSION OUT OF THE GRAIN TIMES VACANCIES ASSOCIATED WITH GAS BUBBLE • GAS ATOM CAPTURE AND BUBBLE COALESCENCE TIMES VACANCIES ASSOCIATED WITH GAS BUBBLE • RESOLUTION TIMES VACANCIES ASSOCIATED WITH GAS BUBBLE • BUBBLE CAPTURE OF INTERSTITIALS

(34)

The actual number of vacancies in a single bubble is obtained by dividing the solution to the above equation by the bubble concentration.

The form of the equation used in REDSTONE is the calculation of bubble radii, not the calculation of bubble vacancies. The bubble radius depends directly on the number of vacancies. The radius equation can thus be related to the above

equation describing bubble vacancies. Many of the properties discussed in earlier chapters also depend on the bubble radius. The bubble radius, in turn, indirectly depends on the fuel properties. Calculating the bubble radius directly allows one to update these properties concurrently, improving the stability of calculation. The specifics of converting from the bubble vacancy equation to the radius equation will be covered in later sections and chapters.

4.2 1-D Diffusion Equation in Spherical Coordinates

The diffusion equation for a single species (n=1 for atoms, n>1 for bubbles, n=v,i for vacancies, interstitials respectively) can be written in general terms as

$$\frac{\partial C_n}{\partial t} = \nabla\bullet\left(D_n \nabla C_n\right) + \alpha_n + \beta_n C_n + \gamma_n C_n^2 + \sum_{\substack{k=1 \\ k\neq n}}^{N} \rho_{nk} C_n C_k + \sum_{\substack{k=1 \\ k\neq n}}^{N} \eta_k C_k + \sum_{\substack{k=1 \\ k\neq n}}^{N}\sum_{\substack{j=k \\ j\neq n}}^{N} \varepsilon_{kj} C_k C_j \qquad (35)$$

where C_n = the unknown quantity of n'th species (concentration or vacancies per bubble), D_n = diffusion coefficient, α_n = constant factors, β_n = constant multipliers of the unknown quantity, γ_n = constant multipliers of the square of the unknown quantity, ρ_{nk} = constant multipliers of products of the unknown quantity and k'th species, ε_{kj} = constant multipliers of products of k'th and j'th species, η_k = constant multipliers involving the k'th species. Note that the 'j' index in the double summation term begins at 'k' rather than '1'. Since there is no distinction between contributions from 'C_kC_j' or 'C_jC_k', starting the summation at '1' would wrongly include a second contribution from this term.

For 1-D spherical coordinates the gradient term, which describes the diffusion loss out of the grain, becomes

$$\nabla \bullet \left(D_n \nabla C_n\right) = \frac{1}{r^2}\frac{\partial}{\partial r}\left(D_n r^2 \frac{\partial C_n}{\partial r}\right) . \qquad (36)$$

Since the diffusion coefficient is held constant during calculations, this equation becomes

$$\nabla \bullet \left(D_n \nabla C_n\right) = \frac{D_n}{r^2}\frac{\partial}{\partial r}\left(r^2 \frac{\partial C_n}{\partial r}\right) . \qquad (37)$$

The specific multipliers of the overall equation are determined by the individual processes. These processes are listed in table 4 by species type and which multiplier is affected. Individual processes are discussed in more detail by equation type below.

4.3 Gas Atom and Bubble Concentrations

With the exception of biased diffusion out of the grain, the processes governing the change in gas atom and bubble concentrations are described by

$$\begin{aligned}\frac{\partial C_n}{\partial t} = {} & D_n \nabla^2 C_n - K_n C_n + K_{n-1} C_{n-1} - b_n C_n N_n + b_{n+1} C_{n+1} N_{n+1} - 32\pi R_n D_n C_n^2 \\ & - \sum_{\substack{k=1 \\ k \neq n}}^{N} \left[4\pi\left(R_n + R_k\right)\left(D_n + D_k\right) - \pi\left(R_n + R_k\right)^2 \left|V_n - V_k\right|\right] C_n C_k \\ & + \sum_{\substack{k=1 \\ k \neq n}}^{n/2} \left[4\pi\left(R_{n-k} + R_k\right)\left(D_{n-k} + D_k\right) - \pi\left(R_{n-k} + R_k\right)^2 \left|V_{n-k} - V_k\right|\right] C_{n-k} C_k \end{aligned} \qquad (38)$$

where each of the terms are in order of appearance: diffusion, source loss from producing the next larger bubble, source gain from the next smaller bubble, resolution loss to the next smaller bubble, resolution gain from the next larger bubble, self-coalescence loss, coalescence loss from random and biased effects, and

Table 4

Identification of Processes

	Single Atoms	Bubbles	Vacancies and Interstitials	Bubble V[illegible]cy Adjustn[illegible]
α	• TOTAL FISSION GAS PRODUCTION RATE		• TOTAL VACANCY AND INTERSTITIAL PRODUCTION RATE • EFFECT OF BUBBLE OVERPRESSURE ON VACANCY CAPTURE	
β	• FRACTION OF PRODUCTION ATOMS DEPOSITED IN EXISTING BUBBLES • LOSS BY RANDOM DIFFUSION OUT OF THE GRAIN	• PRODUCTION BY FISSION (PROBABILISTIC) • LOSS BY RANDOM DIFFUSION OUT OF THE GRAIN	• LOSS FROM CAPTURE BY BUBBLES AND DEFECTS	• LOSS DUE TO DIFFUSION, RESOLUTION RATE, AND COALESCENCE RATE OF BUBBLE
γ	• LOSS BY NUCLEATION OF GAS BUBBLES	• LOSS BY ISO-BUBBLE COALESCENCE		
ρ	• LOSS BY GAS BUBBLE CAPTURE	• LOSS BY GAS ATOM CAPTURE AND BUBBLE COALESCENCE	• LOSS FROM RECOMBINATION	
ε		• PRODUCTION BY NUCLEATION OR COALESCENCE OF SMALLER BUBBLES		• GAIN FROM COALESCENCE OF BUBBLES
η	• PRODUCTION BY RESOLUTION FROM INTRAGRANULAR GAS BUBBLES	• PRODUCTION BY RESOLUTION FROM THE NEXT LARGER INTRAGRANULAR GAS BUBBLES		• GAIN FROM RESOLUTION OF BUBBLES

coalescence gain from random and biased effects. Each will be discussed in the following sections.

4.3.1 Production by Fission

The production of gas atoms by fission can be represented by

$$K = Y_{FG}\dot{F} \qquad (39)$$

where Y_{FG} = yield of fission gas atoms per fission and $\dot{F}$ = fission rate. Both the yield and fission rate are taken as constant within a time step and spatially. This latter constraint is consistent with the fast neutron flux expected in the reactor core.

The yield of fission gas atoms per fission was taken as 0.25 [5] from consideration of the stable isotopes of krypton and xenon.

Researchers generally have considered the fission gas source to contribute only to single gas atoms. However, scoping calculations revealed that if gas atoms are immobile, then the final density of atoms is too great to be realistically supported by the lattice. Because of the high density, there is a distinct possibility that deposition of gas atoms in latter stages of operation would have a finite probability of forming new bubbles or increasing the size of existing bubbles. Such a process can be modeled by considering the overall fission gas source to be divided up into separate components for single atoms and each bubble size. Apportionment is determined by calculating the ratio of nearest neighbor lattice sites for a particular bubble size to the total number of lattice sites available within a unit volume. Lattice sites, whether filled or unfilled, are assumed to always exist and no calculation is done to alter the dimensions of the fuel grain. Since the number of lattice sites remains constant, the total number of lattice sites in a unit volume can be calculated once at the beginning of calculations, i.e.

$$N_L = \frac{1}{a_o^3} \qquad (40)$$

where a_o is the lattice constant. The number of nearest neighbor lattice sites for a single gas atom in a face-centered-cubic (FCC) structure is 12 [5]. The number of nearest neighbor lattice sites for a bubble can be calculated from the surface area of the bubble and the (average) density of lattice sites on the surface. When multiplied by the bubble concentration, C_n, this gives the total nearest neighbor sites for a bubble size in a unit volume. The bubble surface area is

$$A_n = 4\pi R_n^2 \qquad (41)$$

where R_n is the radius of bubble with 'n' atoms. In calculating the density of lattice sites on a bubble surface, most investigators consider only the (100) crystal direction. A more accurate density can be calculated by considering the three principal directions (100), (110), and (111) and taking an average. Lattice site surface densities for each direction are:

$$\frac{2}{a_o^2} \qquad \frac{2}{\sqrt{2}a_o^2} \qquad \frac{5}{\sqrt{3}a_o^2} \qquad (42)$$

(100) (110) (111)

The average of these three directions is $2.10032\ a_o^{-2}$. Combining with the bubble surface area, the number of nearest neighbor sites to a single bubble is given by

$$\text{Nearest Neighbors} = 8.40128\pi R_n^2 a_o^{-2} \ . \qquad (43)$$

Multiplying by the bubble concentration and fission gas source, and dividing by the total number of lattice sites, the source term for bubble one atom larger than 'n' is given by

$$K_n = \beta_n C_n = 8.40128\pi R_n^2 a_o C_n Y_{FG} \dot{F} \ . \qquad (44)$$

Correspondingly, the source for single atoms deposited in the grain is given by

$$K_o = \alpha_1 - \sum_{n=1}^{N} \beta_n C_n = Y_{FG}\dot{F} - \sum_{n=1}^{N} 8.40128\pi R_n^2 a_o C_n Y_{FG} \dot{F} \ . \qquad (45)$$

4.3.2 Nucleation

Historically, nucleation of bubbles has been treated homogeneously with modifying terms added to account for heterogeneous effects [88, 89, 90]. Heterogeneous nucleation has periodically been explicitly applied in an effort to explain anomalies in observed trends [91]. Heterogeneous nucleation was utilized in the BUBL code [87]. However, this code also assumed gas release from the grain occurred entirely from bubble motion and bubble interactions occurred only at

dislocations. Single gas atoms in the matrix only served to nucleate and increase the size of bubbles on dislocation lines. Robertson [92] makes the observation that applied heterogeneous nucleation is on a scale so fine that little difference exists between the two approaches in practice. This is supported by the calculations of Dollins [93] which indicated that a gas atom is several orders of magnitude more likely to meet another gas atom than a dislocation. Also, Warner and Nichols [94] note that the parameter-fitting used to adjust models to actual experimental data minimizes any large discrepancies between the two models. In REDSTONE, homogeneous nucleation is applied with the only heterogeneous effect being the probabilistic production discussed in the previous section. This assumption appears legitimate since fission damage will be lower in UN than in UO_2 and crystalline defect concentration will be negligible compared to the bubble density.

Typically, a stable nucleus of a bubble is considered to be a diatomic cluster of gas atoms [88]. All researched modeling efforts have assumed this, and this will be assumed to be the case here. The modeling of bubble nucleation will introduce a loss term into the single gas atom equation and a gain term into the equation describing diatomic gas bubbles. Gain of diatomic bubbles through nucleation is described by an equation given by

$$\text{Gain Nucleation Rate} = k_{11}C_1^2 \qquad (46)$$

where C_1 = gas concentration and k_{11} = rate constant. The nucleation loss term for single gas atoms is

$$\text{Loss Nucleation Rate} = -2k_{11}C_1^2 \ . \qquad (47)$$

The equation is multiplied by two because two gas atoms are removed for every diatomic cluster formed.

Following a purely theoretical approach, Olander [5] derived the rate constant for single vacancy processes, k_{vv}, in terms of a combinatorial number and basic properties of the fuel and gas atom, i.e.

$$k_{vv} = \frac{z_{vv}\Omega D_v}{a_o^2} \qquad (48)$$

where D_v = diffusion coefficient of vacancies, z_{vv} = a to-be-determined combinatorial number, Ω = volume per lattice site = $a_o^3/4$ for a FCC lattice structure, and a_o = lattice parameter. The combinatorial number as calculated by Olander is based on the number of nearest neighbors, the species jump frequency, and the species concentration. For a FCC structure, the calculated combinatorial number is 84. In considering interactions between xenon atoms, Olander states that the combinatorial number is probably higher than an atomic value (such as for vacancies) because evidence suggests that the xenon atoms migrate as a complex involving the xenon atom and several vacancies. Researchers have estimated that a single gas atom is combined with two to four vacancies, with the most probable being a neutral trivacancy in UO_2 [95,96]. For UN, there have not been any studies to determine the optimum number of vacancies associated with a gas atom. Since a neutral set of vacancies seems most logical, a gas atom in UN will be assumed to consist of the atom and a neutral divacancy. Olander does not calculate the corresponding combinatorial number for a xenon-vacancy complex nor does he give an indication of how much larger this number should be than the atomic value. If the number of occupied nearest neighbors to the complex are the controlling factor, as indicated by Olander's analysis, then a divacancy complex in UN would have 18 versus 12 nearest neighbor sites. Based on this change alone, the combinatorial number becomes 126. However, this is an overly simple adjustment and there are

probably other mitigating factors, such as the effect of divacancies associated with the jumping atom as well as whether a gas atom jumping into a nearest neighbor to a vacancy constitutes nucleation, which might reduce the combinatorial number. Other researchers have successfully utilized a single vacancy mechanism (combinatorial number = 84) to describe xenon migration [88]. Because of the lack of evidence concerning xenon-vacancy complexes in UN and thus the calculation of a combinatorial number, a combinatorial of 84 will be assumed. In order to convert from the vacancy form of the rate constant to a rate constant for xenon, Olander states that in the case of an atomic species other than a vacancy, all one needs to do is replace the vacancy diffusion coefficient in the above equation with the appropriate atomic diffusion coefficient. Interestingly, had the appropriate atomic diffusion coefficient been used in Olander's development of the above equation the vacancy fraction, x_v, would have appeared in the denominator of the rate constant. This happens because of Olander's definition of the diffusion coefficient for an atomic species, i.e.

$$D_x = a_o^2 x_v w \qquad (49)$$

versus the definition for the vacancy diffusion coefficient, i.e.

$$D_v = a_o^2 w \qquad (50)$$

where w = the jump frequency of the species. One possible reason for this discrepancy is that an atomic species interacting with another species (gas atom, dislocation, bubble, grain boundary) will essentially 'see' a vacancy fraction of one due to the vacancies associated with the secondary interacting species. Certainly, the vacancy fraction would be expected to be greater than in the bulk matrix. If a vacancy fraction of one is not the case, then assuming it to be so will give the

maximum rate of interaction. Making appropriate substitutions and notation changes, the rate equation becomes

$$k_{11} = \frac{z_{11}\Omega D_1}{a_o^2} \qquad (51)$$

where D_1 = diffusion coefficient of single xenon atoms and $z_{11} = z_{vv} = 84$. The rate constant for nucleation is given by

$$k_{11} = 21 a_o D_1 \qquad (52)$$

leaving a nucleation rate of

$$\text{Nucleation Rate} = 42 a_o D_1 C_1^2 \qquad (53)$$

where k_{11} was multiplied by two to account for both gas atoms being mobile [5].

Most recent researchers have expressed the nucleation rate from a less theoretical approach, based on the work of Chandrasekhar [34] who developed rate equations for use in colloidal chemistry. Rest [17] expresses the nucleation term as

$$\text{Nucleation Rate} = 16\pi F_1 D_1 C_1^2 \qquad (54)$$

where F_1 = probability that two gas atoms which come together actually stick and R_1=radius of volume occupied by the intragranular gas atom. Hayns and Wood [97] use a similar expression that does not include the sticking factor and is lower by exactly a factor of two.

An effective radius of a gas atom, R_1, can be calculated by assuming an atomic volume of $\Omega = a_o^3/4$ and equating this with the volume of a sphere, $\frac{4}{3}\pi R_1^3$, giving

$$R_1 = a_o \sqrt[3]{\frac{3}{16\pi}} = 0.69336 a_o \ . \qquad (55)$$

This gives a reaction rate (with F_1=1) based on Chandresakhar's work of

$$\text{Nucleation Rate} = 34.8521 a_o D_1 C_1^2 \qquad (56)$$

which is approximately 20% less than that of Olander (45% if a combinatorial number of 126 is used). Esteves [68] has compared Chandresakhar's equations to more exact (and complex) theoretical rate equations [98] and concluded that both are of comparable accuracy. Given the relative closeness of the two results, the degree of uncertainty currently present in fuel modeling, and the conclusions of the above study, it is reasonable to utilize Chandresakhar's equations for bubble nucleation. Since they are also more efficient and straightforward in fuel modeling codes, their use in REDSTONE seems more than justified. It will be seen later that this has advantages as bubble coalescence and atom capture can be treated similarly. A sticking factor is not used in program REDSTONE as its use by Rest appears to be as a fitting parameter to adjust his code to actual data.

The previous discussion of nucleation was based on theory involving random effects. In a temperature gradient, atoms and bubbles will experience an additional force pushing them in a single direction at a speed proportional to the temperature gradient and their size. For nucleation, no additional biased terms are needed since all gas atoms at a particular location are pushed at the same velocity. Only random effects will contribute to the nucleation of bubbles.

4.3.3 Coalescence and Capture

Bubble coalescence and gas atom capture are expressed, as for nucleation, through a rate equation. In this case two concentrations must be considered, one for the bubble and one for either the gas atom or other bubble, i.e.

$$\text{Coalescence Rate} = k_{kn} C_k C_n \qquad (57)$$

where C_k, C_n = concentrations of bubbles with 'k' and 'n' atoms respectively and k_{kn}= rate constant for the reaction. Again, analysis of this equation will be

approached first from theoretical considerations and then from the equations of Chandrasekhar, the form typically used.

Theoretical analysis of the rate constant has been performed by Olander [5]. The method of analysis he uses depends on the predominant process controlling the reaction. Coalescence of reacting species (bubbles, atoms, or vacancies) which are of comparable size, and thus comparable mobilities, is considered to be reaction rate controlled, as was the nucleation rate developed in the previous section. When one of the species is much larger than the other, the process is diffusion limited due to the establishment of a concentration gradient of the more mobile species around the larger species. Between these two extremes a combination of these two processes exists. Interactions between gas atoms and bubbles will be treated as reaction rate controlled as bubble sizes are expected to be small in the scenario here. Interactions between bubbles will be treated likewise. Interactions between bubbles and vacancies/interstitials are treated as diffusion controlled, both theoretically and in practice, as vacancies and interstitials will have much higher diffusivities than the bubbles involved in the interaction. Diffusion controlled reactions will be discussed in the section dealing with vacancy-bubble interaction.

In general, bubble interaction rates based on reaction rate control are cumbersome to calculate from purely theoretical concerns. Partly, this is because one must calculate analogous terms to the jump distance and jump frequency used for atomic species. The difficulty in calculating these terms involves the more general question of what constitutes the formation of the coalesced bubble. Although the time required for a single coalescence is usually considered to be negligible, it is not instantaneous and involves the movement of many atoms from start to finish. This complication manifests itself in the calculation of the

combinatorial number, z_{kn}, used in the rate constant. However, the interaction of a single gas atom (treated as an atomic species) and a stationary bubble (gas atom capture) is simple enough that a combinatorial number can be calculated using a modification of Olander's analysis of vacancy-vacancy interactions. In the following analysis gas atom capture by a bubble will be considered, with the modification that the vacancy fraction will be taken as one.

The rate constant for gas atom capture, analogous to that for nucleation, is given by

$$k_{1n} = \frac{z_{1n}\Omega D_1}{a_o^2} \qquad (58)$$

where Ω = atomic volume, a_o = lattice constant, z_{1n} = combinatorial number, and D_1= diffusion coefficient of the single atom. Because only D_1 is present in the above equation, it is implicitly assumed that the bubble is stationary. In order to calculate a rate constant for two mobile species, it is necessary to calculate k_{n1} and add this to k_{1n}. It is for this reason that the nucleation rate constant in the previous section was multiplied by a factor of two (k_{11}+k_{11}). As discussed previously, the calculation of k_{n1} is complex as it requires the probability of a bubble 'jumping' next to a gas atom. The bubble has been assumed stationary for this reason. However, by considering only k_{1n}, an acceptable value for comparison can still be obtained since the magnitude of k_{n1} will be smaller than k_{1n} due to the gas atom being the more mobile species.

The development of the rate constant, following Olander, begins by first considering the rate of formation of bubbles containing 'n+1' atoms through capture of a single gas atom, i.e.

$$\text{Interaction Rate} = P_{n+1}C_1 \qquad (59)$$

where P_{n+1} = probability that a gas atom jumps into a site nearest neighbor to the bubble. A gas atom that jumps into a nearest neighbor site is assumed to become part of the bubble. The probability P_{n+1} depends on the number of nearest neighbors as well as the probability per second that a gas atom jumps into a particular site. The number of nearest neighbors to a bubble has previously been calculated to be

$$\text{Nearest Neighbors} = 8.40128\pi R_n^2 a_o^{-2} \quad . \qquad (60)$$

Thus, the probability per second that a gas atom jumps into a nearest neighbor site is given by

$$P_{n+1} = 8.40128\pi R_n^2 a_o^{-2} P_x \qquad (61)$$

where P_x = probability per second that a gas atom jumps into a particular site. The probability P_x can be obtained by considering the number of sites next to a specific nearest neighbor site which are not themselves nearest neighbors. One must also consider the probability that a particular site is occupied by a gas atom and the jump frequency of the gas atom in a particular direction.

The number of sites next to a nearest neighbor will range between four and seven. Four sites are present when considering a planar bubble surface (as for a large bubble), while seven sites are present in the case of nucleation. The probability that a particular site is occupied by a gas atom is given by $C_1\Omega$. The jump frequency of gas atoms in the matrix can be obtained from theory for the diffusion coefficient ($x_v = 1$), i.e.

$$D_1 = a_o^2 w \qquad (62)$$

where w = jump frequency of xenon atoms. Rearranging, one gets the jump frequency in terms of measurable quantities,

$$w = \frac{D_1}{a_o^2} \quad . \qquad (63)$$

Putting these quantities together gives

$$P_{n+1} = \left(8.40128\pi\frac{R_n^2}{a_o^2}\right)\left(4C_1\Omega\right)\left(\frac{D_1}{a_o^4}\right) = 33.6051\pi\frac{R_n^2}{a_o^4}D_1C_1\Omega \ . \qquad (64)$$

The rate of gas atom capture for bubbles of 'n' atoms is then

$$\text{Capture Rate} = 33.6051\pi\frac{R_n^2}{a_o^4}D_1C_1\Omega C_n \qquad (65)$$

giving a rate constant of

$$k_{1n} = \frac{z_{1n}\Omega D_1}{a_o^2} = 33.6051\pi\frac{R_n^2}{a_o^4}D_1\Omega \qquad (66)$$

and a combinatorial number of

$$z_{1n} = 33.6051\pi\frac{R_n^2}{a_o^2} \ . \qquad (67)$$

The value for a_o for UN is approximately 4.889×10^{-8} cm. Substituting this into the rate equation gives

$$\text{Capture Rate} = 1.7184\times10^{8}\pi R_n^2 D_1C_1C_n \ . \qquad (68)$$

As with nucleation, many investigators [17,97] handle random gas atom capture following Chandrasekhar [34], i.e.

$$\text{Rate} = 4\pi\left(D_1 + D_n\right)\left(R_1 + R_n\right)C_1C_n \ . \qquad (69)$$

The form of this equation is identical to that used for bubble coalescence, and similar to that for bubble nucleation. It thus provides a uniform theoretical treatment of gas atom and bubble interactions. Both bubble and gas atom motion are included, i.e. neither species is assumed stationary. If the bubble is assumed stationary (D_n=0) the rate equation becomes

$$\text{Rate} = 4\pi D_1\left(R_1 + R_n\right)C_1C_n \ . \qquad (70)$$

This equation differs from that derived using Olander's approach through a linear versus quadratic dependence on bubble radius. An inspection of the equation based on Chandrasekhar reveals that as the bubble becomes much larger than a single atom, the theoretical form of the diffusion limited reaction is achieved, i.e.

$$\text{Diffusion Limited Rate} = 4\pi R_n D_1 C_1 C_n \quad . \qquad (71)$$

The equations of Chandrasekhar, at first glance, appear vastly different from the reaction limited equations based on a purely theoretical treatment (with its associated assumptions). A more direct comparison can be made by first considering nucleation and substituting for R_1 and R_n, which are both approximately 3.072×10^{-8} cm. In this case, the theoretical equation (with D_n assumed to be zero) becomes

$$\text{Nucleation Rate} = 1.62\times10^{-7}\pi D_1 C_1 C_n \qquad (72)$$

while the equation according to Chandrasekhar becomes

$$\text{Nucleation Rate} = 2.46\times10^{-7}\pi D_1 C_1 C_n \quad . \qquad (73)$$

The two rates are comparable and certainly within any error associated with the assumptions made. If, instead, a bubble of 25×10^{-8} cm radius is assumed for R_n, then the rates become

$$\text{Theoretical Rate} = 1.07\times10^{-5}\pi D_1 C_1 C_n \qquad (74)$$

and

$$\text{Chandresakhar Rate} = 1.12\times10^{-6}\pi D_1 C_1 C_n \quad . \qquad (75)$$

The difference between rates is now an order of magnitude and increasing as bubble size increases. Another shortcoming of the theoretical analysis was that the bubble was assumed to be stationary, at least relative to the gas atom, which is certainly not the case for analysis just presented. Thus the accuracy of the theoretical treatment

as applied is questionable, even for the regime where reaction rate control would be dominate. It should be apparent from the above analysis that the theoretical approaches for bubble and gas atom interactions are not applicable over the full range of bubble sizes. The approach using the equations of Chandrasekhar does seem to accommodate the full range, with acceptable agreement for reaction rate control and convergence to the theoretical limit for diffusion control. The conclusion can be made that the theoretical approach as applied by Olander is inaccurate for the full range of bubble sizes and too difficult to use. The approach taken from the work of Chandrasekhar provides a viable alternative to describe interactions over the entire range of bubble sizes. The reaction rate for interaction of two bubbles, according to Chandrasekhar, would be

$$\text{Rate} = 4\pi\left(D_k + D_n\right)\left(R_k + R_n\right)C_k C_n \qquad (76)$$

which is also the same form resorted to by Olander in applying coalescence to his own program and is the form chosen for use in REDSTONE. For self-coalescence loss, this rate is multiplied by two since two species of 'n' size are removed for each reaction.

For biased effects due to a temperature gradient, additional terms are required for any two species of different size. Bubbles of the same size would not interact for the same reason there is no biased nucleation. The force experienced by a bubble in a temperature gradient is due to the difference in surface atom energies between the 'hot' and 'cold' sides of the bubble. The atoms on the hot side will have more energy allowing more atoms to escape from the surface and travel along the inside of the bubble. Some of these atoms will come to rest on the cold side where they will lose their energy and remain. The result will be a net flux of surface atoms to the cold side of the bubble and a migration of the bubble up the

temperature gradient. Biased bubble motion is treated as if a force is acting on the bubble as a whole, and is proportional to the bubble size and the temperature gradient within the bubble. Because of the lower thermal conductivity of the gas within the bubble, the temperature gradient within the bubble is higher than in the bulk matrix. For a spherical bubble, the effective temperature gradient on the bubble, ∇T_b, is related to the temperature gradient in the bulk matrix, ∇T, by

$$\nabla T_b = \frac{3}{2}\nabla T \ . \qquad (77)$$

The velocity of a bubble in a temperature gradient is given by the Nernst-Einstein equation

$$V_n = \frac{D_s F_n}{kT} \qquad (78)$$

where k = Boltzman constant, D_s = Surface diffusion coefficient, and the force, F_n, is given by

$$F_n = \left(\frac{2\pi R_n^3}{a_o^3}\right)\frac{Q_s^*}{T}\nabla T_b \qquad (79)$$

where Q_s^* is the activation energy for surface atoms breaking loose from the surface. Although D_s has been measured for UN, Q_s^* has not. However, Olander [5] notes that the value recommended by Maiya [79] for UO_2 is approximately 80% of the heat of vaporization. This result is reasonable considering that jumps of surface atoms involve atoms which almost leave the surface permanently. Thus, an estimate of Q_s^* for UN can be obtained by assuming it to be 80% the heat of vaporization . For UN, this value has been reported as 1.1250×10^{-11} ergs [99], corresponding to a Q_s^* of 9.0003×10^{-12} ergs.

In a given temperature gradient, it is the size of the bubble that determines the velocity. The coalescence rate will therefore depend directly on the relative velocities of the two bubbles, as well as the cross section for interaction, i.e.

Biased Coalescence Rate = $\pi(R_k + R_n)^2 |V_n - V_k| C_k C_n$. (80)

4.3.4 Resolution

The rate of resolution of gas atoms from bubbles is commonly treated by applying a resolution parameter, b, to the bubble concentration, i.e.

Resolution Rate = bC_nN_n (81)

where C_n = concentration and N_n = number of gas atoms in the bubble.

Experimental measurements [100,101] have placed the resolution parameter for UO_2 in the range $1.8\times10^{-4} < |b| < 3.6\times10^{-4}$. Theoretically, the resolution parameter has been described by a variety of different methods based on various theories. One theory of resolution [102] attributes the process to the production of a thermal spike from a passing fission fragment in the vicinity of a gas bubble. The thermal spike creates temporary disorder in the lattice surrounding the bubble completely mixing fission gas atoms with matrix atoms. The simplest model of this theory [103] proposes that all bubbles intersected by a fission fragment track are entirely destroyed. Obviously, this model works best for small bubbles. The resolution parameter in this case is given by

$b = 2\pi R_n^2 \mu_{ff} \dot{F}$ (82)

where R_n = radius of bubble with 'n' atoms, μ_{ff} = distance travelled by a fission fragment during slowing down from birth energy, and $2\pi R_n^2$ = the bubble cross section for annihilation. Because whole bubbles are destroyed in the above theory, the loss rate of bubbles due to resolution is given by bC_n rather than bC_nN_n.

A modification of the above model [104] suggests that the cross section for annihilation is not the bubble cross section, but a cross section determined by a cylinder of disorder created along the fission fragment track. The cross section for annihilation would then become $2\pi(R_n^2 - R_f^2)$, where R_f is the radius of the damage cylinder. Complete annihilation of a bubble would occur if it was totally within this cylinder. For bubbles larger than this cross section, a passing fission fragment would simply decrease the size of the bubble.

Another proposed theory of resolution [105] is that a passing fission fragment blasts off chunks of the matrix onto the opposite wall of a bubble through a pressure pulse, trapping fission gas in the process. Full conditions for the resolution processes above have been discussed by Blank and Matzke [8]. However, the above theories probably do not describe resolution in a fuel like UN, which has a higher thermal and electrical conductivity than UO_2 [8,106]. For UN the resolution constant is expected to be at least an order of magnitude less.

Other resolution models [10,107,108], based on theory attributed to Nelson, propose a knock-on process whereby the fission fragment physically knocks individual gas atoms out of the gas bubble into the matrix. In this theory, the resolution parameter is given by [5]

$$b = \int_{E_{ff}^{max}}^{T_{min}} \phi(E_{ff})\sigma(E_{ff},T)dE_{ff} = 2\sigma(E_{ff}^{max},T_{min})\ln\left(\frac{E_{ff}^{max}}{T_{min}}\right)\mu_{ff}\dot{F} \qquad (83)$$

where E_{ff} = the energy of a fission fragment at some point in the slowing down process, E_{ff}^{max} = the maximum energy of a fission fragment, $\phi(E_{ff})$ = the energy spectrum of the fission fragment flux, T_{min} = minimum energy required to completely redissolve the gas atom, $\sigma(E_{ff},T)$ = transfer cross section for

energy T to $T+dT$ from a fragment of energy E_{ff}. This resolution parameter is highly dependent on the minimum energy needed for resolution. This minimum energy will increase with increasing temperature since gas atom mobility at higher temperatures will make it easier for an escaped gas atom to diffuse back to the bubble. Olander calculates the above resolution parameter using values of E_{ff}^{max}=67MeV, μ_{ff} = 6 micrometers, and two values of T_{min}, 300 eV and 1 keV. The calculated values are $b=1.1\times10^{-19}F$ and $b=4.1\times10^{-19}F$ for T_{min}=300 eV and 1 keV, respectively. These calculated values are about an order of magnitude lower than experimental values in UO_2. Nelson [10] tried to correct for this by estimating the additional effect of resolution from secondary knock-ons. However, in a fuel such as UN, it is not clear that secondary knock-ons would be produced as easily as in UO_2. For UC, the theoretical single knock-on resolution rate has been shown to be a satisfactory estimate of the experimental resolution rate [9]. Since UN has similar electrical properties, the same conclusion is made for UN. Thus, values of the resolution parameter used in REDSTONE come from the theoretical calculated values of Olander. Resolution at low temperatures (<500 K) utilizes the T_{min}=300eV value, while resolution at high temperatures (> 1500 K) utilizes the T_{min}=1 keV value, with prorated values between these two extremes.

The above resolution parameter is all that is required for small bubbles. For large bubbles, the energy imparted to a gas atom in the center of the bubble may not be enough to force the atom past other gas atoms and out of the bubble. Only gas atoms located within a distance, d, of the surface will have enough energy to be ejected from the bubble. Nelson estimates this distance to be about 15 Å, increasing with decreasing bubble gas density according to the relationship,

$$d = \frac{1/B}{\rho_n} \qquad (84)$$

where B = the van der Waals constant for xenon and ρ_n = the gas bubble density. From the van der Waals equation of state and the proportion of the outer shell of thickness, d, to the total bubble volume, a resolution efficiency for large bubbles is obtained, i.e.

$$\eta_n = 1 - \left\{1 - 15\left[\frac{1}{R_n} + \frac{1}{B}\left(\frac{kT}{2\gamma}\right)\right]\right\}^3 \qquad (85)$$

where R_n = the bubble radius (Å), B = the van der Waals constant, k = Boltzman constant, and γ = surface energy. Thus the resolution rate for bubbles with R_n>15 Å is

$$\text{Resolution Rate} = \eta_n b C_n N_n \ . \qquad (86)$$

Since resolution introduces single gas atoms back into the matrix, the summation of all resolutions of bubbles provides a source term for single gas atoms.

4.3.5 Diffusion Loss Due to a Temperature Gradient

When the grain is subjected to a temperature gradient, gas atoms and bubbles will be forced in a single direction without regard to the symmetry of the grain. For long term operation, this introduces serious complications to the modeling of the fuel grain, i.e. one must develop additional equations and methods to handle the 3-D problem. Because of the difficulty of modeling the 3-D equations, unique solving methods are required. Fortunately, the scenario to be modeled here introduces few biased effects as a result of 1) the essentially negligible temperature gradient during long term operation, and 2) the short duration of operation when a temperature gradient becomes important. The simplifications to the 3-D case to be made here are designed to retain as much of the biased effects as possible, but still maintain

1-D symmetry in the grain. Biased effects have already been introduced into the coalescence terms. In this section, a method will be described for accounting for gas and bubble loss to the grain boundary due to the temperature gradient.

In relation to a spherical fuel grain, the effect of a biased force will be to shift all bubbles of the same size a distance 'h' of a time period Δt. This distance is determined by the velocity of the bubbles. In the case of different sized bubbles, a separate 'h' would be associated with each size. Other investigators [15,109,110] had tackled this asymmetry problem by simply calculating the amount of overlap between two identical spheres separated by a distance 'h'. The fraction of this volume to the total volume of one of the spheres was taken to be the fraction of fission gas retained in the fuel grain. This appears to be the only method used by 1-D codes to model transient release of fission gas from the grain. The volume of the overlapping regions is given by [111]

$$V = \frac{\pi h^2}{6}\left(3R_g - \frac{h}{2}\right) \qquad (87)$$

where R_g = grain radius. Dividing this by the volume of the sphere, $\frac{4}{3}\pi R_g^3$, and subtracting from one gives the fraction released, i.e.

$$\text{Release Fraction} = 1 - \frac{h^2\left(6R_g - h\right)}{16R_g^3} \,. \qquad (88)$$

The distance 'h' is determined by multiplying the bubble velocity by the time step. The bubble velocity used is the average across the fuel grain for a particular bubble size. A sphere the size of the fuel grain is then shifted to determine the percentage loss of bubbles of that size. The percentage loss is applied to all nodes in the grain. Each bubble size is considered separately, but elements and nodes are not. This

relatively simplistic approach was used because of the difficulty of accounting for asymmetric movement of bubbles from element to element and node to node.

4.4 Bulk Vacancy and Interstitial Concentrations

The processes governing bulk vacancy change in the fuel grain are described by

$$\frac{\partial C_v}{\partial t} = D_v \nabla^2 C_v + K_v - \alpha C_v C_i - Z_v D_v \rho_d C_v - \sum_{k=1}^{N} 4\pi R_n D_v C_n \left\{ C_v - C_v^e \exp\left[-\frac{\Omega}{kT}\left(P_n - \frac{2\gamma}{R_n} - P_o \right) \right] \right\} \quad (89)$$

where the terms are in order of appearance: diffusion, source, loss to dislocations, and loss to bubbles. The corresponding equation for interstitials is

$$\frac{\partial C_i}{\partial t} = D_i \nabla^2 C_i + K_i - \alpha C_v C_i - Z_i D_i \rho_d C_i - \sum_{k=1}^{N} 4\pi R_n D_i C_n C_i \quad (90)$$

where each of the terms are analogous to those in the equation for vacancies. Effects from grain boundaries are accounted for through appropriate boundary conditions, not as explicit terms in the equations. Each of the above terms will be discussed in the following sections.

4.4.1 Production by Fission

The production of vacancies and interstitials via fission has not been experimentally investigated for UN. For UO_2, researchers have estimated a production rate of $10^4 \dot{F}\Omega$ to $5\times10^5 \dot{F}\Omega$ [112, 113] with better agreement to experiment being obtained for values close to $5\times10^5 \dot{F}\Omega$. Since production of fission damage in UN is expected to be lower than in UO_2, one would expect that production of vacancies and interstitials would also be lower. An estimate of the production rate in UN can be obtained by comparing experimental values for

fission-enhanced diffusion in both UN and UO_2. Since fission-enhanced diffusion is attributed to the vacancies produced from fission, such a comparison should provide a basis to estimate vacancy production in UN from the corresponding value for UO_2. Fission-enhanced diffusion was discussed in section 2.3.2.2. In that section, it was seen that the diffusion rate in UO_2 was one to two orders of magnitude larger than that in UN. Assuming the vacancy and interstitial production rates in UN are the same order of magnitude lower, the production rates for UN can be estimated to be

$$K_v = K_i = 5 \times 10^3 \dot{F}\Omega \ . \qquad (91)$$

4.4.2 Recombination

Olander [5] has given a detailed discussion of the interactions between point defects, expressing rate constants in terms of combinatorial numbers. His approach, discussed in previous sections, assumed that defects must jump into nearest neighbor positions to interact. However, Gibson [114] has shown that vacancies have a large sphere of influence extending beyond simple nearest neighbors. Olander does not specify what the actual sphere of influence for recombination is, except to suggest that the combinatorial number is approximately 10 times larger (approximately 100 atom volumes) than the nearest neighbor treatment (12 atom volumes). Straalsund and Guthrie [67] use a sphere of influence ranging between 12 and 500 atom volumes. Dollins and Jurisch [71] use a combinatorial number of 5040, equivalent to an atom volume of about 500 atoms. Using this number, the loss rate due to recombination is

$$\text{Recombination Rate} = \alpha C_v C_i = \frac{5040\Omega\left(D_i + D_v\right)}{a_o^2} C_v C_i \qquad (92)$$

where a_o is the atomic jump distance. Although it is clear that a simple nearest neighbor treatment is insufficient for recombination, the exact sphere of influence to use is unclear. For program REDSTONE, the loss rate utilized by Dollins and Jurisch is adopted since it does provide for a larger sphere of influence.

4.4.3 Dislocation Interaction

Although dislocations are assumed to have a negligible effect on bubbles, they are included for vacancies and interstitials because studies have indicated that the slight bias dislocations have for interstitials provides the driving force for void growth [5,67,115]. One of the difficulties in including dislocations is determining their types and densities. To complicate matters, both of these can change over time with dose and temperature. Theoretical treatments have attempted to account for different dislocation types and applied them to the behavior in structural materials such as steel [115,116]. However, dislocation properties are not well documented for nuclear fuels, especially under irradiation conditions. Thus, for nuclear fuels, the applied dislocation treatment has been much more generalized, with dose and temperature effects not included. In fact, Rest [117] leaves out dislocations completely. Typically, the interaction terms are described by

$$\text{Vacancy - Dislocation Interaction Rate} = Z_v \rho_d D_v \left[C_v - C_v^e \exp\left(\frac{\Omega}{kT} P_o \right) \right] \qquad (93)$$

and

$$\text{Interstitial - Dislocation Interaction Rate} = Z_i \rho_d D_i C_i \qquad (94)$$

where thermal emission of interstitials is considered negligible and Z_v, Z_i = numbers characterizing capture volumes associated with unit length, ρ_d = overall dislocation density, C_v, C_i = concentrations, D_v, D_i = diffusion coefficients, and C_v^e = thermal

(equilibrium) concentration. Because a good correlational predictor of dislocation types and densities for nuclear fuels is not available, this same set of equations is used in REDSTONE. The obvious limitations of this approach are that the dislocation density is held constant and only one type of dislocation is considered.

The numbers Z_v and Z_i are typically on the order of unity. However, because of the larger strain field surrounding an interstitial, Z_i is greater than Z_v by one to two percent. This difference is provides the driving force for steady-state void growth. Consistent with other researchers, the values of Z_v and Z_i in REDSTONE are taken to be 1.00 and 1.02 respectively.

The dislocation density was adopted from other researchers as well. Values of 10^8 cm/cm^3 to 10^{11} cm/cm^3 have been used [67,118,119]. Since UN is expected to have fewer dislocations than other corresponding materials in an irradiation environment, the lower value of 10^8 cm/cm^3 is used in REDSTONE.

4.4.4 Bubble Interaction

Vacancy and interstitial diffusion rates are much greater than corresponding bubble diffusion rates. As a result, the rate at which vacancies and interstitials interact with a bubble can be calculated via diffusion equations in the vicinity of a single stationary bubble (diffusion limited reaction), given by

$$\frac{\partial C_v}{\partial t} = \frac{D_v}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial C_v}{\partial r}\right) + K_v \qquad (95)$$

and

$$\frac{\partial C_i}{\partial t} = \frac{D_i}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial C_i}{\partial r}\right) + K_i \qquad (96)$$

with the boundary conditions

$$\left.\begin{aligned} C_v &= C_v^e \exp\left[-\frac{\Omega}{kT}\left(P_n - \frac{2\gamma}{R_n} - P_o\right)\right] \\ C_i &= C_i^e = 0 \end{aligned}\right\} \; r = R_n \qquad (97)$$

and

$$\left.\begin{aligned} C_v &= C_v^b \\ C_i &= C_i^b \\ \frac{dC_v}{dr} &= 0 \\ \frac{dC_i}{dr} &= 0 \end{aligned}\right\} \; r = R_c \qquad (98)$$

plus initial conditions, where C_v,C_i = vacancy and interstitial concentrations, D_v,D_i=vacancy and interstitial diffusion coefficients, K_v,K_i = vacancy and interstitial production rates, R_c = capture volume associated with the bubble, and R_n= bubble radius. The effects from non-equilibrium bubbles have been included as boundary conditions for vacancies. A similar boundary equation would normally be present for interstitials except that the interstitial equilibrium concentration has been taken as zero. Capture effects from dislocations are not included in these equations as they are already included in the overall bulk concentration equations and bubbles are the dominant sink on vacancies [93]. Also, since it is desirable to come up with an analytic expression, recombination is not included here so that the equations remain uncoupled. Recombination is also included in the calculation of the bulk vacancy and interstitial concentrations. Solutions to the resulting equations are still not easily obtained in finite form because of the time derivative. Additionally, establishment of the initial condition for the time variable involves knowing the exact profile around each bubble size at the beginning of each time step. This

requirement is not practical since it involves modeling vacancy profile changes for all bubbles. A greatly simplified solution can be obtained by taking a quasi-steady-state approach, i.e. setting the left hand side of the above equations to zero. Such an approach assumes that the vacancies and interstitials equilibriate around a bubble much faster than the bubble changes in size. This assumption is reasonable since one can argue that after the initial establishment of bubble profiles, all bubbles would have vacancy and interstitial profiles established around them. Subsequent coalescences, resolutions, etc would not change these profiles much from their equilibrium positions. Re-equilibriation of vacancies/interstitials would occur rapidly since only minor changes would be required. In support of this, Matthews [15] has concluded that a steady-state treatment of vacancies is appropriate in most situations. With this simplification, the equations become

$$0 = \frac{D_v}{r^2}\frac{d}{dr}\left(r^2\frac{dC_v}{dr}\right) + K_v \qquad (99)$$

and

$$0 = \frac{D_i}{r^2}\frac{d}{dr}\left(r^2\frac{dC_i}{dr}\right) + K_i \qquad (100)$$

with boundary conditions as before and no initial conditions.

Olander notes that the solutions to these equations exhibit two distinct regions within the capture volume, each dominated by a different process. In the inner region a diffusion loss term is dominant and production effects can be neglected. In the outer region diffusion loss is negligible and only production need be considered. Thus approximate solutions in the inner region can be found by solving for the equations

$$\frac{1}{r^2}\frac{\mathrm{d}}{\mathrm{d}r}\left(r^2\frac{\mathrm{d}C_\mathrm{v}}{\mathrm{d}r}\right)=0 \qquad (101)$$

and

$$\frac{1}{r^2}\frac{\mathrm{d}}{\mathrm{d}r}\left(r^2\frac{\mathrm{d}C_\mathrm{i}}{\mathrm{d}r}\right)=0\ . \qquad (102)$$

The outer region is actually described by the overall production/loss equations describing bulk vacancy/interstitial concentrations. The concentrations change little in the outer region and quickly attain the values in the bulk matrix.

Usually, the capture radius, R_c, is taken to be much greater than the bubble radius, R_n. Assuming this to be true here, the flux boundary conditions at $r = R_c$ given earlier can be replaced with concentration boundary conditions at $r = \infty$, i.e.

$$\left.\begin{aligned} C_\mathrm{v} &= C_\mathrm{v}^\mathrm{b} \\ C_\mathrm{i} &= C_\mathrm{i}^\mathrm{b} \end{aligned}\right\}\ r=\infty \qquad (103)$$

where a 'b' superscript has been added to denote concentrations far away from bubbles. These boundary conditions essentially convert the capture volume to an infinite medium. The solutions to these greatly simplified equations are,

$$C_\mathrm{v} = C_\mathrm{v}^\mathrm{e} + \left(C_\mathrm{v}^\mathrm{b} - C_\mathrm{v}^\mathrm{e}\right)\left(1-\frac{R_\mathrm{n}}{r}\right) \qquad (104)$$

and

$$C_\mathrm{i} = C_\mathrm{i}^\mathrm{b}\left(1-\frac{R_\mathrm{n}}{r}\right)\ . \qquad (105)$$

The resulting reaction rate for a single bubble is given by

$$\text{Reaction Rate} = \left(-4\pi R_\mathrm{n}^2\right)\ J \qquad (106)$$

where J is the flux across the bubble surface given by

$$J = -D\left(\frac{\mathrm{d}C}{\mathrm{d}r}\right)_{R_\mathrm{n}}\ . \qquad (107)$$

Making the appropriate substitutions, the net single bubble interaction rates become

$$\text{Vacancy Interaction Rate} = 4\pi R_n D_v \left\{ C_v - C_v^e \exp\left[-\frac{\Omega}{kT}\left(P_n - \frac{2\gamma}{R_n} - P_o \right) \right] \right\} \qquad (108)$$

and

$$\text{Interstitial Interaction Rate} = 4\pi R_n D_i C_i \qquad (109)$$

where the 'b' superscript has been dropped from the bulk concentrations. To get the interaction rates within a unit volume, one needs to multiply by the concentration of bubbles in the volume, i.e.

$$\text{Vacancy Interaction Rate} = 4\pi R_n D_v C_n \left\{ C_v - C_v^e \exp\left[-\frac{\Omega}{kT}\left(P_n - \frac{2\gamma}{R_n} - P_o \right) \right] \right\} \qquad (110)$$

and

$$\text{Interstitial Interaction Rate} = 4\pi R_n D_i C_i C_n \,. \qquad (111)$$

Previous investigators have used the forms above for use in transient modeling [71].

4.5 Bubble Radius Adjustment

A bubble vacancy adjustment equation was described in section 4.1 in terms of the various processes involved. In this section, this equation will be related to the concentration equation, and then the conversions needed to directly describe bubble radius changes will be discussed. Many of the relationships discussed here have already been discussed previously.

4.5.1 Relation to Bubble Concentration Equation

Ignoring for the moment the affect of the bulk vacancy concentration on the growth of fission gas bubbles, the resulting equation for bubble vacancy growth is related to the bubble concentration equations by noting

RATE OF CHANGE OF NUMBER OF VACANCIES ASSOCIATED WITH BUBBLE OF 'N' ATOMS	=	RATE OF GAIN OF BUBBLES OF 'N' ATOMS THROUGH PROCESSES INVOLVING BUBBLES OF 'K' ATOMS	×	NUMBER OF VACANCIES PER BUBBLE OF 'K' ATOMS	
	+	RATE OF GAIN OF BUBBLES OF 'N' ATOMS THROUGH PROCESSES INVOLVING BUBBLES OF 'K' ATOMS AND BUBBLES OF 'J' ATOMS	×	NUMBER OF VACANCIES PER BUBBLE OF 'K' ATOMS + NUMBER OF VACANCIES PER BUBBLE OF 'J' ATOMS	(112)
	-	RATE OF LOSS OF BUBBLES OF 'N' ATOMS	×	NUMBER OF VACANCIES PER BUBBLE OF 'N' ATOMS	

Stated in words, this conversion is simply the loss terms in the original concentration equation multiplied by the number of vacancies per bubble for the bubble being lost, plus the gain terms multiplied by the sum of the appropriate number of vacancies per bubble for the bubbles being gained. To complete the equation, one must add the effects from bubble-vacancy and bubble-interstitial interactions developed in section 4.4.4. This equation describes the rate of change of the <u>total</u> vacancy concentration associated with bubbles of 'n' atoms in a unit volume. The total vacancies associated with bubbles of 'n' atoms, C_{nv}, is equal to $C_n \times N_{nv}$, where C_n is the bubble concentration and N_{nv} is the number of vacancies per bubble of 'n' atoms. In REDSTONE, the bubble concentration is held constant during bubble volume adjustment calculations. The resulting equation with all terms included is

$$\frac{\partial C_{nv}}{\partial t} = \left(D_n \nabla^2 C_n\right) N_{nv} - K_n C_n N_{nv} + K_{n-1} C_{n-1} N_{(n-1)v}$$
$$- b_n C_n N_n N_{nv} + b_{n+1} C_{n+1} N_{n+1} N_{(n+1)v} - 32\pi R_n D_n C_n^2 N_{nv}$$
$$- \sum_{\substack{k=1 \\ k \neq n}}^{N} \left[4\pi \left(R_n + R_k\right)\left(D_n + D_k\right) - \pi \left(R_n + R_k\right)^2 \left|V_n - V_k\right| \right] C_n C_k N_{nv}$$
$$+ \sum_{\substack{k=1 \\ k \neq n}}^{n/2} \left[4\pi \left(R_{n-k} + R_k\right)\left(D_{n-k} + D_k\right) - \pi \left(R_{n-k} + R_k\right)^2 \left|V_{n-k} - V_k\right| \right] C_{n-k} C_k \left(N_{kv} + N_{(n-k)v}\right)$$
$$- 4\pi R_n D_i C_i C_n + 4\pi R_n D_v C_n \left\{ C_v - C_v^e \exp\left[-\frac{\Omega}{kT} \left(P_n - \frac{2\gamma}{R_n} - P_o \right) \right] \right\} \qquad (113)$$

where each of the terms have been discussed previously.

4.5.2 Relation to Bubble Vacancy Equation

The bubble vacancy rate equation presented in the previous section is not the exact form used in REDSTONE. The equation in REDSTONE describes the rate of change of the radius of a bubble, which involves a conversion of the bubble vacancy equation. The conversion is made by noting that vacancies in a bubble are related to the radius by

$$N_{nv} = \frac{4\pi R_n^3}{\Omega} \qquad (114)$$

where Ω is the atomic volume for one vacancy. Applying this conversion and rearranging somewhat, produces the following equation for radius change,

$$C_n \frac{\partial R_n}{\partial t} = \left(D_n \nabla^2 C_n\right)\frac{R_n}{3} - K_n C_n \frac{R_n}{3} + K_{n-1} C_{n-1} \frac{R_{n-1}^3}{3R_n^2}$$
$$- b_n C_n N_n \frac{R_n}{3} + b_{n+1} C_{n+1} N_{n+1} \frac{R_{n+1}^3}{3R_n^2} - 32\pi R_n D_n C_n^2 \frac{R_n}{3}$$
$$- \sum_{\substack{k=1 \\ k \neq n}}^{N} \left[4\pi\left(R_n + R_k\right)\left(D_n + D_k\right) - \pi\left(R_n + R_k\right)^2 \left|V_n - V_k\right|\right] C_n C_k \frac{R_n}{3}$$
$$+ \sum_{\substack{k=1 \\ k \neq n}}^{n/2} \left[4\pi\left(R_{n-k} + R_k\right)\left(D_{n-k} + D_k\right) - \pi\left(R_{n-k} + R_k\right)^2 \left|V_{n-k} - V_k\right|\right] C_{n-k} C_k \left(\frac{R_k^3}{3R_n^2} + \frac{R_{n-k}^3}{3R_n^2}\right)$$
$$+ \frac{\Omega}{R_n} D_i C_i C_n + \frac{\Omega}{R_n} D_v C_n \left\{C_v - C_v^e \exp\left[-\frac{\Omega}{kT}\left(P_n - \frac{2\gamma}{R_n} - P_o\right)\right]\right\}. \qquad (115)$$

In addition to this conversion, several of the properties depend on bubble radius. These include the diffusion coefficients, production rates, bubble pressures, and resolution rates. Each of these dependencies are reviewed in following sections. These should be considered in the formulation of the finite element problem. Unfortunately, inclusion of these adds even more non-linearity to the problem. Preliminary investigations indicated that only inclusion of bubble pressure dependencies made a noticeable difference in the convergence behavior. Additionally, calculational stability could be adversely affected by allowing the terms used to calculate concentrations to change when calculating radii. Fortunately, bubble pressure is only used in the vacancy interaction terms which were not used to calculate concentrations. Disregarding the other radius dependent property terms and keeping terms common with the concentration equation constant during the radius calculation simplifies the finite element formation of the radius equation considerably. The details of the formulation will be discussed in Chapter VII.

4.5.3 Radius Dependent Properties

Following is a short review of the radius dependence of fuel properties used in program REDSTONE.

4.5.3.1 Diffusion Coefficient

The diffusion coefficient has been defined previously in section 3.3.2.2 as

$$D_n^{new} = \frac{6\lambda\Omega D_s}{\pi\lambda_s^2 R_n^2}\sin^2\left(\frac{\lambda_s}{2R_n}\right) \qquad R_n > \frac{\lambda_s}{\pi} \qquad (116)$$

$$D_n^{new} = \frac{6\lambda\Omega D_s}{\pi\lambda_s^2 R_n^2} \qquad R_n \le \frac{\lambda_s}{\pi} \qquad (117)$$

where the included terms have been defined previously. The diffusion coefficient is seen to have an R_n^2 dependence modified for larger bubbles by the $\sin^2()$ term which also depends on bubble radius.

4.5.3.2 Production Rate

The production rate of bubbles through deposition of new fission gas depends on the fraction of the bubble surface area and the total atomic sites, i.e.

$$K_n = \frac{8.40128\pi\Omega}{a_o^2} R_n^2 \qquad (118)$$

where the included terms have been defined previously. This equation indicates an R_n^2 dependence.

4.5.3.3 Bubble Pressure

The bubble pressure is described by the van der Waals reduced equation of state,

$$P_n = \frac{N_n kT}{\frac{4}{3}\pi R_n^3 - N_n B} \qquad (119)$$

where the included terms have been defined previously. The bubble pressure is seen to have a more complex radius dependence. The equation is still in a form, however, that can be dealt with effectively.

4.5.3.4 Resolution Rate

For bubbles with radius greater than 15 Å the resolution parameter is described by

$$b_n = b\eta_n = b\left(1 - \left\{1 - 15\left[\frac{1}{R_n} + \frac{1}{B}\left(\frac{kT}{2\gamma}\right)\right]\right\}^3\right) \qquad (120)$$

where the included terms have been defined previously. For smaller bubbles, there is no radius dependence. The resolution parameter for larger bubbles is again seen to have a more complex radius dependence.

CHAPTER V
CONCENTRATION FINITE ELEMENT FORMULATION

5.1 Generalities

Detailed modeling of the fission gas behavior within a fuel grain must take into account not only the individual gas atoms but also all possible gas bubble sizes. In reality, the number of bubble sizes modeled is either capped at some reasonable level or the bubbles are grouped into one or more 'average' bubble sizes which exhibit properties which are supposedly an average of the group. This latter approach has been the preferred method in recent years as it greatly reduces storage and running time requirements. Normally, the average group values are calibrated to agree with release results rather than swelling results, i.e. bubble group values are based on a bubble containing the average number of gas atoms in the group and bubble volume is not conserved. Programming the collapse of bubble sizes into groups is only done after multi-bubble size calculations are performed, confirming the collapsed results. This has not been done for UN. Since the study here concerns calculating both the release and the swelling, the decision was made to model individual bubble sizes rather than groups. Scoping calculations had indicated that the number of different bubble sizes formed would be small and therefore manageable within the storage and time constraints of the program. Consideration of the gas atoms as well as 'N' bubble groups results in 'N+1' non-linear equations that must be solved simultaneously. Because of the complexity and non-linearities of these equations, numerical methods must be used to solve them.

The diffusion equation can be expressed in the following general form

$$\frac{\partial C_n}{\partial t} = D_n \nabla^2 C_n + \alpha_n + \beta_n C_n + \gamma_n C_n^2 + \sum_{\substack{k=1 \\ k \neq n}}^{N} \rho_{nk} C_n C_k + \sum_{\substack{k=1 \\ k \neq n}}^{N} \eta_k C_k + \sum_{\substack{k=1 \\ k \neq n}}^{N} \sum_{\substack{j=k \\ j \neq n}}^{N} \varepsilon_{kj} C_k C_j \quad (121)$$

This is the same equation presented in Chapter III except that the diffusion coefficient is constant here. This is also the form of the equation used in REDSTONE for bubble and vacancy/interstitial concentrations. The form used for bubble volume adjustment is somewhat different due to higher non-linearities and will be discussed in the next chapter.

5.2 Discretization of the Time Derivative

Note that in the general equation the concentration appears as a function of both time and position. A finite element formulation could be performed incorporating both these variables, but this would require nodal discretization in time as well as position. Such a discretization would require a single solution of a set of equations describing all concentrations at all nodal positions and at all discretized times. Such a formulation would be unmanageable for any reasonable problem. Instead, the time dependence is usually treated with finite difference techniques and finite elements are used for the remaining position variables. In the formulation of the current problem the partial derivative in time is replaced with a backward Euler approximation so that only a finite element formulation with position need be done.

A backward Euler approximation given by

$$\frac{{}^{t+\Delta t}\partial C_n}{\partial t} \approx \frac{{}^{t+\Delta t}C_n - {}^{t}C_n}{\Delta t} \quad (122)$$

was chosen as it does not impose a maximum time step size and thus is unconditionally stable. The notation used is that the unknown quantity is at time 't+Δt', and the last calculated known quantity is at time 't'. The unconditionally stable feature is important when dealing with non-linear problems because divergence of time stepping schemes is not always apparent. In other words, a non-linear problem using a central difference or forward Euler time stepping scheme may eventually converge even though the solution became unstable somewhere in the process. As a result, an answer can be obtained which is totally meaningless. Although a backward Euler scheme provides the stability needed in a non-linear problem, it has disadvantages verses both the other schemes.

Against a central difference scheme, i.e.

$$\frac{{}^{t+\Delta t}\partial C_n}{\partial t} \approx \frac{{}^{t+2\Delta t}C_n - {}^{t}C_n}{2\Delta t} \qquad (123)$$

which is second order accurate, the backward Euler scheme is only first order accurate. However, this trade-off is more than acceptable in ligh of the poor accuracy of experimental data in UN which must be input into the calculation. The advantage the forward Euler scheme,

$$\frac{{}^{t+\Delta t}\partial C_n}{\partial t} \approx \frac{{}^{t+2\Delta t}C_n - {}^{t+\Delta t}C_n}{\Delta t} \qquad (124)$$

over the backward Euler scheme is not accuracy, but has to do with the efficiency of the resulting program. The forward Euler scheme is an explicit method. As a result, for a linear problem time stepping involves only updates to the force vector The use of a forward Euler scheme allows one to calculate the stiffness matrix only once at the beginning of the problem because the non-time dependent stiffness terms remain constant in a linear problem. If a backward Euler is used, the stiffness

matrix must be updated at every time step due to the dependence of this scheme on ${}^{t+\Delta t}C_n$. However, in a non-linear problem, such is the case here, there are terms in the stiffness matrix which must be updated at every time step anyway. Thus the advantages of the forward Euler and central difference schemes evaporate, and given the choice, the backward Euler is the preferred scheme to use for non-linear problems.

After discretizing the time variable, the time derivative can simply be incorporated into the already defined generic constants, i.e. since ${}^{t}C_n$ is known

$$\alpha_n^{new} = \alpha_n^{old} + \frac{{}^{t}C_n}{\Delta t} \qquad (125)$$

and

$$\beta_n^{new} = \beta_n^{old} + \frac{1}{\Delta t} \; . \qquad (126)$$

Thus in the formulation the time derivative is incorporated into the generic constants, leaving

$$0 = D_n \nabla^2 C_n + \alpha_n + \beta_n C_n + \gamma_n C_n^2 + \sum_{\substack{k=1 \\ k \neq n}}^{N} \rho_{nk} C_n C_k + \sum_{\substack{k=1 \\ k \neq n}}^{N} \eta_k C_k + \sum_{\substack{k=1 \\ k \neq n}}^{N} \sum_{\substack{j=k \\ j \neq n}}^{N} \varepsilon_{kj} C_k C_j \; . \qquad (127)$$

With the time derivative discretized, the problem can now be formulated in finite elements using the Galerkin method.

5.3 Development of the Galerkin Equations

Transformation of the original governing equation to the Galerkin equations is presented here. In later sections, the Galerkin equations will be tailored to the problem and individual terms will be identified.

First the general equation is set equal to a residual term,

$$\langle R\rangle = D_n\nabla^2 C_n + \alpha_n + \beta_n C_n + \gamma_n C_n^2 + \sum_{\substack{k=1\\k\neq n}}^{N}\rho_{nk}C_nC_k + \sum_{\substack{k=1\\k\neq n}}^{N}\eta_k C_k + \sum_{\substack{k=1\\k\neq n}}^{N}\sum_{\substack{j=k\\j\neq n}}^{N}\varepsilon_{kj}C_kC_j \ . \tag{128}$$

The 'N_G' Galerkin residual equations ('N_G' will be defined later) are

$$\int_\Omega \langle R\rangle \phi_i d\Omega = 0 \qquad i = 1,2,.....N_G \tag{129}$$

where ϕ_i is a weighting function which will later be identified as the interpolation functions. The above provides for 'N_G' equations per bubble group (degrees of freedom). Thus the total number of equations will turn out to be 'N_G' times the maximum number of bubble groups. It will be necessary to utilize integration by parts to reduce the order of differentiation in the above equation. This manipulation is easier if the exact physical geometry is first substituted into the integral equation, i.e. 1-D spherical coordinates. Thus,

$$d\Omega = 4\pi r^2 dr \tag{130}$$

and

$$\nabla^2 C_n = \frac{1}{r^2}\frac{d}{dr}\left(r^2\frac{dC_n}{dr}\right) \tag{131}$$

so that

$$\int_\Omega D_n\nabla^2 C_n\phi_i d\Omega = 4\pi\int_R D_n\frac{d}{dr}\left(r^2\frac{dC_n}{dr}\right)\phi_i dr \ . \tag{132}$$

The 4π is a constant that will appear in every integral and thus will have no effect on the equations. It can safely be dropped. Applying integration by parts,

$$\int_R D_n\frac{d}{dr}\left(r^2\frac{dC_n}{dr}\right)\phi_i dr = \left[\phi_i D_n r^2\frac{dC_n}{dr}\right]_{R_{in}}^{R_{out}} - \int_R D_n r^2\frac{dC_n}{dr}\frac{d\phi_i}{dr}dr \ . \tag{133}$$

Note that the second order derivative in the original equation has been reduced to a first order derivative. The first term on the RHS of the equation is a boundary term to be evaluated at the boundaries of the element. For the problem being looked at here, the explicit boundary conditions are that the gradient in concentration is zero at the center of the grain ($r = 0$) and that the concentration is specified on the outer boundary. Additionally, inter-element boundary conditions require continuity of concentration and concentration gradient. For the inside element which contains a node at $r = 0$, the boundary term reduces to zero at $r = 0$ and thus requires only evaluation at the outer boundary of the element. Upon assembly of the equations and application of the specified concentrations, the elemental boundary terms cancel out except where a non-zero concentration gradient is specified on a physical boundary [120]. In the problem developed here, there are no specified concentration gradients and thus integration by parts effectively yields

$$\int_R D_n \frac{d}{dr}\left(r^2 \frac{dC_n}{dr}\right)\phi_i dr = -\int_R D_n r^2 \frac{dC_n}{dr}\frac{d\phi_i}{dr} dr \qquad (134)$$

with the resulting equations, which are all equal to zero, becoming

$$\int_R \left(-D_n \frac{dC_n}{dr}\frac{d\phi_i}{dr} + \phi_i \left[\alpha_n + \beta_n C_n + \gamma_n C_n^2 + \sum_{\substack{k=1 \\ k\neq n}}^{N} \rho_{nk} C_n C_k + \sum_{\substack{k=1 \\ k\neq n}}^{N} \eta_k C_k + \sum_{\substack{k=1 \\ k\neq n}}^{N}\sum_{\substack{j=k \\ j\neq n}}^{N} \varepsilon_{kj} C_k C_j \right] \right) r^2 dr \,. \qquad (135)$$

5.4 Incrementalization of the Galerkin Equations

It is desirable that the above equations can be solved by proven inversion matrix solving techniques. To do this the equation needs to be put into the form '$[\mathbf{K}]\{\mathbf{C}\}=\{\mathbf{F}\}$' where $[\mathbf{K}]$ is the stiffness matrix, $\{\mathbf{C}\}$ is the vector containing the unknown values, and $\{\mathbf{F}\}$ is the force or load vector. This form implies that

'[K]{C}' should be a set of linear equations, i.e. [K] is constant with {C} being the unknowns removed from the original equations. It is clear that putting the Galerkin equations as presented above into the form '[K]{C}={F}' will not result in linear equations. However, several methods of dealing with this situation are available. Some of these are reviewed below. As will be seen, these are natural extensions of each other.

The simplest solution method is to assign all the constant and non-linear terms to {F} and the coefficients to the linear terms to [K]. The non-linear terms used in {F} would be based on values from the previous iteration, and the stiffness matrix [K] would be constant during the iterations, i.e.

$$[\mathbf{K}]\ {}^{t+\Delta t}\{\mathbf{C}\}^{i} = {}^{t+\Delta t}\{\mathbf{F}\}^{i-1} \qquad (136)$$

where the 't' and 't+Δt' denote the previous time and the current time, respectively and the 'i-1' and 'i' denote the previous iteration and the current iteration, respectively. Because all the non-linearities are based entirely on previous values, the convergence rate for this method could be too slow, depending on the degree of non-linearity. In fact, very strong non-linearities might prevent convergence altogether. This method does have the advantage that [K] is constant and thus does not have to be updated at every iteration. Unfortunately, in the problem to be solved here the constants in [K] change anyway (although not at every iteration) due to explicit calculation of some of the coefficients at each time step.

Another method, which is a variation of the above, is to move all the non-linearities from {F} on the RHS to the LHS while factoring out one C_n, C_k, or C_j (whichever is applicable) to produce {C} and a [K] which is partially dependent on the unknown values. Thus {F} would remain constant and [K] would now have to

be updated at every iteration with values from the previous iteration. The resulting set of equations are denoted by

$$ {}^{t+\Delta t}[\mathbf{K}]^{i-1} \; {}^{t+\Delta t}\{\mathbf{C}\}^{i} = \{\mathbf{F}\} \; . \qquad (137) $$

As in the first method, the convergence rate could be too slow since the driving force towards convergence has simply been switched from {**F**} to [**K**]. Additionally, [**K**] must now be recalculated at every iteration. Both of these methods require that the calculation of the unknowns at the time of interest, ${}^{t+\Delta t}C_n^{(i)}$, depend on the previously calculated values at a previous time or iteration, ${}^{t+\Delta t}C_n^{(i-1)}$ or ${}^{t}C_n$.

A final method, and the one chosen for the current problem, is to utilize the incrementalization of C_n and cause the unknown values to be ΔC_n. An incremental dependence is achieved by defining

$$ {}^{t+\Delta t}C_n^{i} = {}^{t+\Delta t}C_n^{i-1} + \Delta C_n \qquad (138) $$

where ΔC_n = the unknown change from the previous iteration and ${}^{t+\Delta t}C_n^{(0)} = {}^{t}C_n$. Thus if traditional Newton iteration is employed, the last calculated value from the previous time period is all that is needed to start the iteration. The resulting equations are

$$ {}^{t+\Delta t}[\mathbf{K}]^{i-1} \; {}^{t+\Delta t}\{\Delta\mathbf{C}\}^{i} = {}^{t+\Delta t}\{\mathbf{F}\}^{i-1} \; . \qquad (139) $$

For an incremental, non-iterative method, the same equations are used except that there is only one iteration performed and thus the value from the previous time period, i.e. '0'th iteration, is always required. This incrementalization has the added advantage that the convergence can be easily observed since $\Delta C_n \rightarrow 0$ as convergence is achieved. The biggest advantage, however, is that the convergence is now driven by changing terms in both [**K**] and {**F**}. This accelerates convergence as well as

provides additional stability for strong non-linearities. In fact, this third implementation can be considered to be a combination of the first two methods, incorporating the best of both (at least in the problem to be solved here). Again [**K**] must be recalculated at each iteration, but the gains in stability and convergence rate are considered to be worth the increase in calculations.

It should be obvious from the above discussions that the values at time 't+Δt' are always dependent on previously iterated values at time 't+Δt' except for the very first iteration, where the values from time 't' should be used. With this understood, the writing of the basic equations can be simplified by dropping the time superscript. Additionally, it is apparent that if the unknowns are incrementalized as defined above, then only the 'i-1' designator appears. Thus the iteration superscript can also be dropped with the understanding that no superscript refers to the last known value, i.e. the 'i-1'th iteration. Except for a few obvious places, both these superscripts will be dropped for further illustrations, i.e. $C_n = {}^{t+\Delta t}C_n^{(i-1)}$.

Using this modified notation, the incrementalized concentration terms can now be substituted into the Galerkin equations. After some rearrangement the Galerkin equations become

$$\int_R \left\{ \left[-D_n \frac{d}{dr}\frac{d\phi_i}{dr} + \phi_i \left(\beta_n + 2\gamma_n C_n + \sum_{\substack{k=1 \\ k\neq n}}^{N} \rho_{nk} C_k \right) \right] \Delta C_n \right.$$

$$\left. + \phi_i \left[\sum_{\substack{k=1 \\ k\neq n}}^{N} \rho_{nk} C_n + \sum_{\substack{k=1 \\ k\neq n}}^{N} \eta_k + \sum_{\substack{k=1 \\ k\neq n}}^{N} \sum_{\substack{j=k \\ j\neq n}}^{N} \varepsilon_{kj} C_j \right] \Delta C_k + \phi_i \left[\sum_{\substack{k=1 \\ k\neq n}}^{N} \sum_{\substack{j=k \\ j\neq n}}^{N} \varepsilon_{kj} C_k \right] \Delta C_j \right\} r^2 dr$$

$$+ \int_R \left\{ \phi_i \left[\gamma_n (\Delta C_n)^2 + \sum_{\substack{k=1 \\ k\neq n}}^{N} \rho_{nk} \Delta C_n \Delta C_k + \sum_{\substack{k=1 \\ k\neq n}}^{N} \sum_{\substack{j=k \\ j\neq n}}^{N} \varepsilon_{kj} \Delta C_k \Delta C_j \right] \right\} r^2 dr =$$

$$\int_R \left\{ D_n \frac{dC_n}{dr}\frac{d\phi_i}{dr} - \phi_i \left[\alpha_n + \beta_n C_n + \gamma_n C_n^2 + \sum_{\substack{k=1 \\ k\neq n}}^{N} \rho_{nk} C_n C_k + \sum_{\substack{k=1 \\ k\neq n}}^{N} \eta_n C_k + \sum_{\substack{k=1 \\ k\neq n}}^{N} \sum_{\substack{j=k \\ j\neq n}}^{N} \varepsilon_{kj} C_k C_j \right] \right\} r^2 dr \quad (140)$$

Note that the unknown quadratic (non-linear) terms have been separated out on the LHS. Except for these terms the above equation would easily fit into the form '$[\mathbf{K}]\{\Delta\mathbf{C}\}=\{\mathbf{F}\}$'. Inspection of the above equations will reveal that the LHS controls the rate of convergence while the RHS controls the accuracy of the result. Additionally, as convergence is approached the quadratic terms become negligible very quickly. Thus it can safely be assumed that dropping the quadratic terms will only slow the convergence rate and will have no effect on the resulting solution. The equation can further be simplified by noting that the 'k' and 'j' indices in the double summation terms of the LHS can be interchanged without affecting the equation. Making these changes yields the equation

$$\int_R \left\{ \left[-D_n \frac{d}{dr}\frac{d\phi_i}{dr} + \phi_i \left(\beta_n + 2\gamma_n C_n + \sum_{\substack{k=1 \\ k\neq n}}^{N} \rho_{nk} C_k \right) \right] \Delta C_n \right.$$

$$\left. + \phi_i \left[\sum_{\substack{k=1 \\ k\neq n}}^{N} \rho_{nk} C_n + \sum_{\substack{k=1 \\ k\neq n}}^{N} \eta_k + \sum_{\substack{k=1 \\ k\neq n}}^{N} \sum_{\substack{j=k \\ j\neq n}}^{N} \varepsilon_{kj} C_j + \sum_{\substack{j=1 \\ j\neq n}}^{N} \sum_{\substack{k=j \\ k\neq n}}^{N} \varepsilon_{jk} C_j \right] \Delta C_k \right\} r^2 dr =$$

$$\int_R \left\{ D_n \frac{dC_n}{dr}\frac{d\phi_i}{dr} - \phi_i \left[\alpha_n + \beta_n C_n + \gamma_n C_n^2 + \sum_{\substack{k=1 \\ k\neq n}}^{N} \rho_{nk} C_n C_k + \sum_{\substack{k=1 \\ k\neq n}}^{N} \eta_n C_k + \sum_{\substack{k=1 \\ k\neq n}}^{N} \sum_{\substack{j=k \\ j\neq n}}^{N} \varepsilon_{kj} C_k C_j \right] \right\} r^2 dr \qquad (141)$$

The '[**K**]', '{ΔC}', and '{**F**}' terms are now readily identifiable,i.e.

$$[\mathbf{K}]\{\Delta\mathbf{C}\} = \{\mathbf{F}\} \Rightarrow \begin{bmatrix} K_{11} \cdots K_{1N} \\ \vdots \ddots \vdots \\ K_{N1} \cdots K_{NN} \end{bmatrix} \begin{Bmatrix} \Delta C_1 \\ \vdots \\ \Delta C_N \end{Bmatrix} = \begin{Bmatrix} F_1 \\ \vdots \\ F_N \end{Bmatrix} \qquad (142)$$

where

$$K_{nn} = \int_R \left\{ -D_n \frac{d}{dr}\frac{d\phi_i}{dr} + \phi_i \left(\beta_n + 2\gamma_n C_n + \sum_{\substack{k=1 \\ k\neq n}}^{N} \rho_{nk} C_k \right) \right\} r^2 dr \qquad (143)$$

and

$$K_{nk} = \int_R \left\{ \phi_i \left(\rho_{nk} C_n + \eta_k + \sum_{\substack{j=k \\ j\neq n}}^{N} \varepsilon_{kj} C_j + \sum_{\substack{j=n \\ j\neq k}}^{N} \varepsilon_{jk} C_j \right) \right\} r^2 dr \; . \qquad (144)$$

The rest of the matrix terms not identified are filled out in a similar manner. The terms above are repeated for every weighting function, ϕ_i. The final stiffness matrix [**K**] will consist of 'N_G× N' rows. Discretization of the problem allows one to determine the weighting functions and causes the stiffness matrix to become square with 'N_G× N' columns.

5.5 Discretization of the Galerkin Equations

So that the Galerkin equations can be formulated in a form useful for solving real problems, the specific problem must be discretized. Discretization is a necessary prerequisite to determining weighting functions and methods to calculate values such as 'C_n' and 'ΔC_n'. First, the problem is attacked in a general form by assuming physical elements with 'N_G' nodes per element. Normally, the first and last nodes (in a 1-D problem) determine the physical boundaries of the element. Within each element, the values at the nodes will be known or calculated. Between nodes, values will have to be interpolated using weighting functions (akainterpolation functions), i.e. the ϕ_i's which were used above. The requirements for interpolating functions are that ϕ_i equals one at the i'th node and equals zero at all other nodes. There are no requirements governing the behavior of the weighting functions between nodes, and continuity across element boundaries is automatically taken care off by the formation of the Galerkin equations themselves. So that the nodal values can be differentiated from the interpolated values, a subscript modification will be introduced which refers to the local (elemental) node referred to, i.e. $C_{n:i}$ will refer to a nodal concentration where 'n' indicates the bubble group, as before, and 'i' indicates the node number. The interpolated values are related to the nodal values by

$$C_n = \sum_{i=1}^{N_G} \phi_i C_{n:i} \qquad (145)$$

and

$$\Delta C_n = \sum_{i=1}^{N_G} \phi_i \Delta C_{n:i} \qquad (146)$$

and

$$r_n = \sum_{i=1}^{N_G} \phi_i r_{n:i} \quad . \qquad (147)$$

These values are substituted into the previously developed Galerkin equations. Because of the interpolation of ΔC_n, every degree of freedom in an element generates 'N_G' unknowns to be determined in the discretized Galerkin equations. The result is a [**K**] matrix that is square of dimensions '$N_G \times N$' as well as {Δ**C**} and {**F**} vectors of length '$N_G \times N$'. Note that because of the first order gradient term, the weighting functions must be continuous and once differentiable. Had iteration upon the original governing equation been formulated instead of a Galerkin formulation, the weighting functions would have had to be twice differentiable.

Once the elemental matrices and vectors have been formed, they are assembled into a global matrix and vectors which are solved using common solving routines such as Gaussian elimination or direct substitution-iteration. For the problem here, Gaussian elimination was chosen as the solving method.

5.6 Determination of Interpolating Functions

Interpolation functions are used to interpolate between nodes in a problem. When elements are defined within a problem, the interpolation functions need only interpolate within an element. Values between elements are accounted during assembly into the global matrix through the boundary nodes connecting elements. In theory, one would want interpolation functions which exactly fit the profile of the unknown being modeled. In practice, this cannot be done since one must know the solution to know the profile. However, a general idea of the solution profile is usually known, and appropriate interpolation functions can be chosen. The only criteria for the i'th interpolation function is that it is has a specified value (usually one) at the i'th node and is zero at all other elemental nodes. Other constraints can

be imposed, such as requiring orthogonal functions, but are not required. Since most problem solutions are continuous and smoothly varying, the most often used interpolation functions are based on simple polynomials. More complex functions have been used when the problem solution warrants it. However, it becomes increasingly more difficult to meet the nodal criteria mentioned above with more complex functions. Another method used to keep the interpolation functions simple is to keep the number of elemental nodes low. Thus, typically, investigators use only two or three nodes per element per dimension. The polynomial interpolation functions for both these sets are presented in fig. 1. The specific characteristics of the problem solution here are best modeled using three nodes per element in the outermost element. For this reason, program REDSTONE was programmed to use three nodes per element for all elements.

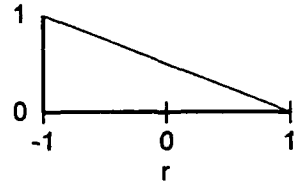
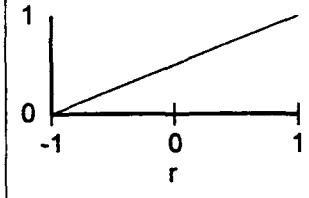
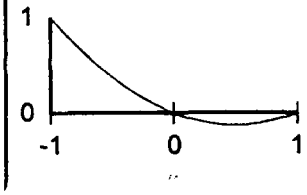
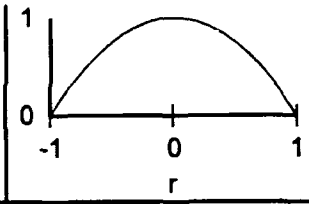
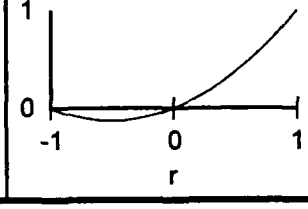
2 Node Element	$h_1 = \frac{1}{2}(1 - r)$	
	$h_2 = \frac{1}{2}(1 + r)$	
3 Node Element	$h_1 = \frac{1}{2}(1 - r) - \frac{1}{2}(1 - r^2)$	
	$h_2 = (1 - r^2)$	
	$h_3 = \frac{1}{2}(1 + r) - \frac{1}{2}(1 - r^2)$	

Fig. 1 Interpolation Functions for Two and Three Nodes

CHAPTER VI
BUBBLE RADII FORMULATION

6.1 Generalities

The finite element formulation for the bubble radii equations involves more detail than for the concentration equations due to higher non-linearities and the fact that some of the fuel properties depend on bubble radius. Specific dependencies were discussed in Chapter IV and included the resolution parameter, source term, diffusion coefficient, pressure, and bubble velocity. It was also mentioned in that chapter that all but one of the dependencies, pressure, could be left out of the actual finite element implementation in program REDSTONE. The bulk of the formulation method for bubble radii remains the same as for concentrations. In fact, the main difference is the apportionment of terms in the governing equation to the correct finite element terms. In Chapter IV, use was made of a generic governing equation including constant, linear, and quadratic terms. This equation was then incrementalized before being put into Galerkin residual equations. Because of the higher non-linearities, the generic equation in Chapter IV is not sufficient for describing the bubble radii equations.

In this chapter, rather than approach the formulation using a general equation, each individual process will be looked at separately. These processes are the same ones previously discussed. Each process will be incrementalized, leaving only constant and linear terms to deal with. Two incrementalizations are presented, one with full radius dependence incorporated and one with the radius dependent properties and concentration equation terms held constant.. Except for pressure, this latter form was the one used in program REDSTONE. Once incrementalized, the

methods used in Chapter IV to build the Galerkin equations, the stiffness matrix, and force vector are the same needed for bubble radii equations. The incrementalized bubble radii terms need only be put into residual Galerkin equations, with linear terms being put into the stiffness matrix and constant terms being put into the force vector. Only the diffusion gradient term needs to be specifically looked at in terms of the formulation of the residual Galerkin equations. This occurs because the bubble radius, which now modifies the diffusion term, is also a function of position in the grain.

Extensive use is made of expansion in Taylor series which have been truncated after first order terms, i.e.

$$f(x+\Delta x) \approx f(x) + \Delta x \frac{f'(x)}{1!} \; . \qquad (148)$$

The purpose of using Taylor series is to isolate unknowns in linear, first order terms, yet keep the non-linear behavior. In practice, the formulation method used here could have been used for the concentration formulation with exactly the same results.

The bubble radii adjustment equation was presented in Chapter III. It is reprinted here for reference,

$$
\begin{aligned}
C_n \frac{\partial R_n}{\partial t} = {} & \left(D_n \nabla^2 C_n\right)\frac{R_n}{3} - K_n C_n \frac{R_n}{3} + K_{n-1} C_{n-1} \frac{R_{n-1}^3}{3R_n^2} \\
& - b_n C_n N_n \frac{R_n}{3} + b_{n+1} C_{n+1} N_{n+1} \frac{R_{n+1}^3}{3R_n^2} - 32\pi R_n D_n C_n^2 \frac{R_n}{3} \\
& - \sum_{\substack{k=1 \\ k \neq n}}^{N} \left[4\pi\left(R_n + R_k\right)\left(D_n + D_k\right) - \pi\left(R_n + R_k\right)^2 \left|V_n - V_k\right|\right] C_n C_k \frac{R_n}{3} \\
& + \sum_{\substack{k=1 \\ k \neq n}}^{n/2} \left[4\pi\left(R_{n-k} + R_k\right)\left(D_{n-k} + D_k\right) - \pi\left(R_{n-k} + R_k\right)^2 \left|V_{n-k} - V_k\right|\right] C_{n-k} C_k \left(\frac{R_k^3}{3R_n^2} + \frac{R_{n-k}^3}{3R_n^2}\right) \\
& - \frac{\Omega}{R_n} D_i C_i C_n + \frac{\Omega}{R_n} D_v C_n \left\{C_v - C_v^e \exp\left[-\frac{\Omega}{kT}\left(P_n - \frac{2\gamma}{R_n} - P_o\right)\right]\right\} .
\end{aligned}
\tag{149}
$$

As programmed in REDSTONE, the radii which are changed during radii calculations are the ones at the very ends of the terms and the vacancy/interstitial interaction terms. Other radii remain constant during calculations.

6.2 Incrementalization of Individual Processes

6.2.1 Resolution Loss

Resolution loss is described by

$$\text{Resolution Loss} = \tfrac{1}{3} b_n C_n R_n N_n \tag{150}$$

where the resolution parameter, b_n, for bubbles with R_n>15 Å is given by

$$b_n = b\eta_n = b\left\{1 - \left(X^3 - \frac{3X^2Y}{R_n} + \frac{3XY^2}{R_n^2} - \frac{Y^3}{R_n^3}\right)\right\} \tag{151}$$

where

$$X = 1 - \frac{Y}{B}\left(\frac{kT}{2\gamma}\right) \tag{152}$$

$$Y = 15 \times 10^8 \ . \tag{153}$$

Combining this radius dependency with the loss term, incrementalizing R_n, and applying Taylor series gives

$$\text{Loss} = \tfrac{1}{3} b_n C_n N_n R_n + \tfrac{1}{3} b C_n N_n \left\{ 1 - \left(X^3 - \frac{3X^2Y}{R_n} + \frac{3XY^2}{R_n^2} - \frac{2Y^3}{R_n^3} \right) \right\} \Delta R_n \quad . \qquad (154)$$

Note that the use of incrementalization and Taylor series have converted the resolution loss term into the last known loss term plus the correction term for the new time step. The form above would only be applied for bubbles with R_n>15 Å. For smaller bubbles, b_n has no radius dependence. Such a discrepancy could introduce oscillations into the radius calculations if a bubble has a radius approximately equal to 15 Å. The corresponding incrementalized loss term for small bubbles is

$$\text{Loss} = \tfrac{1}{3} b_n C_n N_n \left(R_n + \Delta R_n \right) \qquad (155)$$

which is considerably simplified. This is also the form used if one leaves out the radius dependence in the resolution parameter altogether and is the form used in program REDSTONE.

6.2.2 Resolution Gain

Resolution gain is somewhat more difficult to handle than loss because of the mixture of dependencies on two different bubble radii. However, the basic approach for incrementalization is the same. The gain term is given by

$$\text{Resolution Gain} = \tfrac{1}{3} b_k C_k N_k \frac{R_k^3}{R_n^2} \qquad (156)$$

where the 'n+1' subscript has been replaced by a 'k' for simplification. Using the same methods as for resolution loss, one arrives at (for R_k>15 Å),

$$\text{Gain} = \tfrac{1}{3} b_k C_k N_k \frac{R_k^3}{R_n^2} + bC_k N_k \left\{ \left(1 - X^3\right) \frac{R_k^2}{R_n^2} + 2X^2 Y \frac{R_k}{R_n^2} - XY^2 \frac{1}{R_n^2} \right\} \Delta R_k$$
$$+ bC_k N_k \left\{ \tfrac{2}{3} \left(X^3 - 1\right) \frac{R_k^3}{R_n^3} - 2X^2 Y \frac{R_k^2}{R_n^3} + 2XY^2 \frac{R_k}{R_n^3} - \tfrac{2}{3} Y^3 \frac{R_k^3}{R_n^5} \right\} \Delta R_n \quad . \qquad (157)$$

For smaller bubbles and retaining the radius dependence only partially,

$$\text{Gain} = \tfrac{1}{3} b_k C_k N_k \left[\frac{R_k^3}{R_n^2} + \frac{3R_k^2}{R_n^2} \Delta R_k - \frac{2R_k^3}{R_n^3} \Delta R_n \right] \qquad (158)$$

which is the form used in REDSTONE.

6.2.3 Self-Coalescence Loss

Loss through coalescence of bubbles with same size bubbles is given by

$$\text{Self - coalescence Loss} = \tfrac{32}{3} \pi R_n^2 D_n C_n^2 \qquad (159)$$

The diffusion coefficient radius dependence can be expressed by

$$D_n = \frac{D_o}{R_n^2} \qquad (160)$$

then the loss term becomes

$$\text{Self - coalescence Loss} = \tfrac{32}{3} \pi D_o C_n^2 \qquad (161)$$

where

$$D_o = \frac{6\lambda \Omega D_s}{\pi \lambda_s^2} \sin^2\left(\frac{\lambda_s}{2R_n}\right) \qquad R_n > \frac{\lambda_s}{\pi} \qquad (162)$$

$$D_o = \frac{6\lambda \Omega D_s}{\pi \lambda_s^2} \qquad R_n \le \frac{\lambda_s}{\pi} \quad . \qquad (163)$$

The sine term was not incorporated explicitly into the formulation even though it depends on R_n. This was done because of problems with using Taylor series for a periodic function and because of the conditional application of the term. However, the sine term is still accounted for in the last known solution incorporated into the

force vector. Note also that without the sine term, the resulting incrementalized loss term has no dependence on bubble radius.

If, instead, the diffusion coefficient dependence is left out and the original term from the concentration equation is held constant, then the incremental solution becomes

$$\text{Self - coalescence Loss} = \frac{32}{3}\pi R_n D_n C_n^2 \left[R_n + \Delta R_n \right] \qquad (164)$$

where only the last term in brackets changes during calculation. This is the form used in REDSTONE.

6.2.4 Coalescence Loss

Loss through coalescence of bubbles with different size bubbles is given by

$$\text{Coalescence Loss} = \frac{4}{3}\pi R_n \left(R_n + R_k \right)\left(D_n + D_k \right) C_n C_k \ . \qquad (165)$$

Applying methods as before with full radius dependence included yields,

$$\begin{aligned} \text{Coalescence Loss} = & \frac{4}{3}\pi R_n \left(R_n + R_k \right)\left(D_n + D_k \right) C_n C_k \\ & + \frac{4}{3}\pi C_n C_k \left[D_k \left(2R_n + R_k \right) - D_n R_k \right] \Delta R_n \\ & + \frac{4}{3}\pi C_n C_k R_n \left[D_n - D_k \left(1 + 2\frac{R_n}{R_k} \right) \right] \Delta R_k \end{aligned} \qquad (166)$$

and with radius dependence partially left out,

$$\text{Coalescence Loss} = \frac{4}{3}\pi \left(R_n + R_k \right)\left(D_n + D_k \right) C_n C_k \left[R_n + \Delta R_n \right] \qquad (167)$$

where only the last term in brackets changes during calculation.

6.2.5 Coalescence Gain

Gain through coalescence of two smaller bubbles is given by

$$\text{Coalescence Gain} = \frac{4}{3}\pi \left(R_j + R_k \right)\left(D_j + D_k \right) C_j C_k \frac{1}{R_n^2}\left(R_j^3 + R_k^3 \right) \qquad (168)$$

where the 'n-k' subscript has been replaced with 'j' for convenience. Applying methods as before with full radius dependence included yields,

$$\begin{aligned}\text{Coalescence Gain} = {} & \tfrac{4}{3}\pi\left(R_j + R_k\right)\left(D_j + D_k\right)C_jC_k\frac{1}{R_n^2}\left(R_j^3 + R_k^3\right) \\ & - \tfrac{8}{3}\pi C_jC_k\frac{1}{R_n^3}\left(R_j^3 + R_k^3\right)\left(R_j + R_k\right)\left(D_j + D_k\right)\Delta R_n \\ & + \tfrac{4}{3}\pi C_jC_k\frac{1}{R_n^2}\left[\frac{D_k}{R_k}\left(2R_k^4 - R_j^3R_k + R_jR_k^3 - 2R_j^4\right)\right. \\ & \qquad \left. + D_j\left(4R_k^3 + R_j^3 + 3R_k^2R_j\right)\right]\Delta R_k \\ & + \tfrac{4}{3}\pi C_jC_k\frac{1}{R_n^2}\left[\frac{D_j}{R_j}\left(2R_j^4 - R_k^3R_j + R_kR_j^3 - 2R_k^4\right)\right. \\ & \qquad \left. + D_k\left(4R_j^3 + R_k^3 + 3R_j^2R_k\right)\right]\Delta R_j \end{aligned} \qquad (169)$$

and with radius dependence partially left out,

$$\begin{aligned}\text{Coalescence Gain} = {} & \tfrac{4}{3}\pi\left(R_j + R_k\right)\left(D_j + D_k\right)C_jC_k \\ & \times\left[\frac{R_k^3 + R_j^3}{R_n^2} - 2\frac{R_k^3 + R_j^3}{R_n^3}\Delta R_n + \frac{3R_k^2}{R_n^2}\Delta R_k + \frac{3R_j^2}{R_n^2}\Delta R_j\right]\end{aligned} \qquad (170)$$

where only the last term in brackets changes during calculation.

6.2.6 Biased Coalescence Loss

Loss through biased coalescence with another bubble is given by

$$\text{Biased Coalescence Loss} = \tfrac{1}{3}\pi R_n\left(R_n + R_k\right)^2\left|V_n - V_k\right|C_nC_k \ . \qquad (171)$$

Bubble velocity depends on radius with the dependency,

$$V_n = V_oR_n^3 \ . \qquad (172)$$

Unfortunately, incrementalization of the velocity requires splitting the terms into constant and correction pieces which are then placed at different locations in the

stiffness matrix and force vector. Keeping the absolute value incorporated becomes extremely difficult. Because of this, the velocity terms are kept constant in both the radius dependent and radius independent formulations. The absolute value then resides in the force vector during calculations. However, incorporating the other radius dependent terms into the formulation gives,

$$\begin{aligned}\text{Biased Coalescence Loss} = &\tfrac{1}{3}\pi R_n\left(R_n + R_k\right)^2\left|V_n - V_k\right|C_nC_k \\ &+\tfrac{1}{3}\pi\left(3R_n^2 + 4R_nR_k + R_k^2\right)\left|V_n - V_k\right|C_nC_k\Delta R_n \\ &+\tfrac{1}{3}\pi\left(2R_n^2 + 2R_nR_k\right)\left|V_n - V_k\right|C_nC_k\Delta R_k \ . \end{aligned} \qquad (173)$$

Holding radius dependent terms from the concentration equation constant yields the form used in REDSTONE,

$$\text{Biased Coalescence Loss} = \tfrac{1}{3}\pi\left(R_n + R_k\right)^2\left|V_n - V_k\right|C_nC_k\left[R_n + \Delta R_n\right] \qquad (174)$$

where only the last term in brackets changes during calculation.

6.2.7 Biased Coalescence Gain

Gain through biased coalescence of two smaller bubbles is given by

$$\text{Biased Coalescence Gain} = \tfrac{1}{3}\pi\left(R_j + R_k\right)^2\left|V_j - V_k\right|C_jC_k\frac{1}{R_n^2}\left(R_j^3 + R_k^3\right) \qquad (175)$$

where the 'n-k' subscript has again been replaced with 'j' for convenience. The arguments for keeping the velocity terms constant in biased coalescence loss also apply here. Applying methods as before to the rest of the radius dependence in the above yields

$$\text{Biased Coalescence Gain} = \tfrac{1}{3}\pi\left(R_j + R_k\right)^2\left|V_j - V_k\right|C_jC_k\frac{1}{R_n^2}\left(R_j^3 + R_k^3\right)$$
$$-\tfrac{2}{3}\pi\left(R_j + R_k\right)^2\left|V_j - V_k\right|C_jC_k\frac{1}{R_n^3}\left(R_j^3 + R_k^3\right)\Delta R_n$$
$$+\frac{\pi}{3R_n^2}\left(5R_k^4 + 8R_k^3R_j + 3R_k^2R_j^2 + 2R_kR_j^3 + 2R_j^4\right)\left|V_j - V_k\right|C_jC_k\Delta R_k$$
$$+\frac{\pi}{3R_n^2}\left(5R_j^4 + 8R_j^3R_k + 3R_j^2R_k^2 + 2R_jR_k^3 + 2R_k^4\right)\left|V_j - V_k\right|C_jC_k\Delta R_j \qquad (176)$$

and holding the radius dependent terms carried over from the concentration equation constant yields

$$\text{Biased Coalescence Gain} = \tfrac{1}{3}\pi\left(R_j + R_k\right)^2\left|V_j - V_k\right|C_jC_k$$
$$\times\left[\frac{R_j^3 + R_k^3}{R_n^2} - 2\frac{R_j^3 + R_k^3}{R_n^3}\Delta R_n + \frac{3R_k^2}{R_n^2}\Delta R_k + \frac{3R_j^2}{R_n^2}\Delta R_j\right] \qquad (177)$$

where only the last term in brackets changes during calculation.

6.2.8 Source Loss

Loss through deposition of another fission gas atom in the bubble is given by

$$\text{Source Loss} = \tfrac{1}{3}K_nC_nR_n \qquad (178)$$

where the radius dependence of the source term is

$$K_n = K_oR_n^2 \ . \qquad (179)$$

Inclusion of this into the incremental formulation gives

$$\text{Source Loss} = \tfrac{1}{3}K_nC_nR_n + K_nC_n\Delta R_n \qquad (180)$$

or if the source radius dependence is left out,

$$\text{Source Loss} = \tfrac{1}{3}K_nC_n\left[R_n + \Delta R_n\right] \qquad (181)$$

where only the last term in brackets changes during calculation.

6.2.9 Source Gain

Gain through deposition of another fission gas atom in the next smaller bubble is given by

$$\text{Source Gain} = \tfrac{1}{3} K_k C_k \frac{R_k^3}{R_n^2} \qquad (182)$$

where the 'n-1' subscript has been replaced by 'k' for convenience. The radius dependence of the source term is as stated in the last section. The full radius dependent formulation is

$$\text{Source Gain} = \tfrac{1}{3} K_k C_k \frac{R_k^3}{R_n^2} - \tfrac{2}{3} K_k C_k \frac{R_k^3}{R_n^3} \Delta R_n + \tfrac{5}{3} K_k C_k \frac{R_k^2}{R_n^2} \Delta R_k \qquad (183)$$

with the partially radius independent formulation being

$$\text{Source Gain} = \tfrac{1}{3} K_k C_k \left[\frac{R_k^3}{R_n^2} - 2 \frac{R_k^3}{R_n^3} \Delta R_n + 3 \frac{R_k^2}{R_n^2} \Delta R_k \right] \qquad (184)$$

where only the term in brackets changes during calculation.

6.2.10 Interstitial Capture

A decrease in radius due to capture of interstitials is given by

$$\text{Interstitial Capture Decrease} = \Omega D_i C_i C_n \frac{1}{R_n} . \qquad (185)$$

The incremental formulation of this term is

$$\text{Interstitial Capture Decrease} = \frac{\Omega D_i C_i C_n}{R_n} \left[1 - \frac{\Delta R_n}{R_n} \right] . \qquad (186)$$

This is the form used in REDSTONE. There is not a partially radius dependent form as this term did not appear in the original concentration equation.

6.2.11 Vacancy Capture

An increase in radius due to capture of vacancies is given by

$$\text{Vacancy Capture Increase} = \Omega D_v C_v C_n \frac{1}{R_n} . \qquad (187)$$

The incremental formulation of this term is similar to that for interstitial capture, i.e.

$$\text{Vacancy Capture Increase} = \frac{\Omega D_v C_v C_n}{R_n}\left[1 - \frac{\Delta R_n}{R_n}\right] . \qquad (188)$$

Again, there is not a partially radius dependent form as this term did not appear in the original concentration equation.

6.2.12 Vacancy Emission

A decrease in radius due to emission of vacancies is given by

$$\text{Vacancy Emission Decrease} = \Omega D_v C_n \frac{1}{R_n} C_v^e \exp\left\{\frac{\Omega}{kT}\left(P_o + \frac{2\gamma}{R_n} - P_n\right)\right\} . \qquad (189)$$

The incremental formulation of this term is quite different from the previous two terms because of the exponential which is dependent on both bubble radius and bubble pressure. The bubble pressure has a radius dependence given by

$$P_n = \frac{X}{R_n^3 - Y} \qquad (190)$$

where

$$X = \frac{4 N_n kT}{3\pi} \qquad (191)$$

and

$$Y = \frac{4 N_n B}{3\pi} . \qquad (192)$$

After incrementalization and then Taylor series expansion, one obtains

$$\text{Vacancy Emission Decrease} = \frac{\Omega D_v C_n}{R_n} C_v^e \exp\left\{\frac{\Omega}{kT}\left(P_o + \frac{2\gamma}{R_n} - P_n\right)\right\} \times \left[1 - \left(\frac{1}{R_n} + \frac{2\Omega\gamma}{kTR_n^2} - \frac{3\Omega X R_n^2}{kT\left(R_n^3 - Y\right)^2}\right)\right] \Delta R_n \qquad (193)$$

which is the form used in REDSTONE.

6.3 Modification of the Diffusion Term

The diffusion term for bubble radius adjustment is given by

$$\text{Diffusion Loss} = \frac{R_n D_n}{3r^2} \frac{d}{dr}\left(r^2 \frac{dC_n}{dr}\right) . \qquad (194)$$

Before incrementalizing this term, it will be useful to first incorporate it into a Galerkin formulation. Holding the diffusion coefficient constant, the Galerkin residual equations for this term are given by

$$\text{Diffusion Loss} = \frac{4\pi}{3} D_n \int_R R_n \frac{d}{dr}\left(r^2 \frac{dC_n}{dr}\right) \phi_n dr . \qquad (195)$$

As before, the 4π can be dropped. The above term is different from the formulation in the previous chapter by the presence of R_n. Applying integration by parts gives

$$\text{Diffusion Loss} = \left[\frac{1}{3} D_n R_n \phi_i r^2 \frac{dC_n}{dr}\right]_{R_{out}}^{R_{in}} - \frac{D_n}{3} \int_R r^2 \frac{dC_n}{dr} \frac{d\left(R_n \phi_i\right)}{dr} dr . \qquad (196)$$

The first term in brackets is evaluated at the element boundaries. Like the concentration gradient, the concentration gradient times the bubble radius must be continuous and smoothly varying across the boundary. There can be no jumps in bubble size at a boundary. As a result, this term cancels out everywhere except on physical boundaries where the gradient is specified. Since the gradient is never

specified in the problem here, this term can be eliminated entirely. The second term involves the gradient of a product. Expanding this out gives for the diffusion loss term

$$\text{Diffusion Loss} = -\frac{D_n}{3}\int_R R_n r^2 \frac{dC_n}{dr}\frac{d\phi_i}{dr}dr - \frac{D_n}{3}\int_R \phi_i r^2 \frac{dC_n}{dr}\frac{dR_n}{dr}dr \ . \qquad (197)$$

The result of the R_n has been to add an extra term to be included into the formulation. These terms can now be incrementalized and incorporated using the techniques presented in Chapter V.

CHAPTER VII
PROGRAM REDSTONE ORGANIZATION

7.1 Overview

Program REDSTONE is organized as shown in fig. 2. The flow of program control involves three separate and one overall iteration cycles. The three separate iteration cycles involve calculations of bubble concentrations, bulk vacancy and interstitial concentrations, and bubble radii. The overall iteration cycle is completed when no additional iterations are required in the bubble concentration or bubble radii steps during any one overall iteration. Time stepping is done using time steps and time periods. Periods define a set of operational conditions for a length of time. Time steps are user specified equal interval divisions within the time period to facilitate calculational and/or output requirements. Transient analysis is accomplished by stepping between the initial and final period conditions based on the number of time steps in a period. Within REDSTONE is built-in the capability to 'restart' from a previously saved analysis. This allows one to set up a specific initial profile for subsequent exploratory analyses. A complete program listing can be found in Appendix 1.

Input to the program is accomplished through an input file with data in free format and identified by position on a line and a unique line number. Comment lines are allowed and identified by an '*' or 'C' at the beginning of the line. Optionally, a restart input file is also used to input starting conditions. Output of the program consists of readable output at user-specified intervals, and a character based file for input to separate plotting/data analysis schemes. Additionally, after each

period, specific parameters are saved in machine readable form for use in subsequent restart analyses using program REDSTONE.

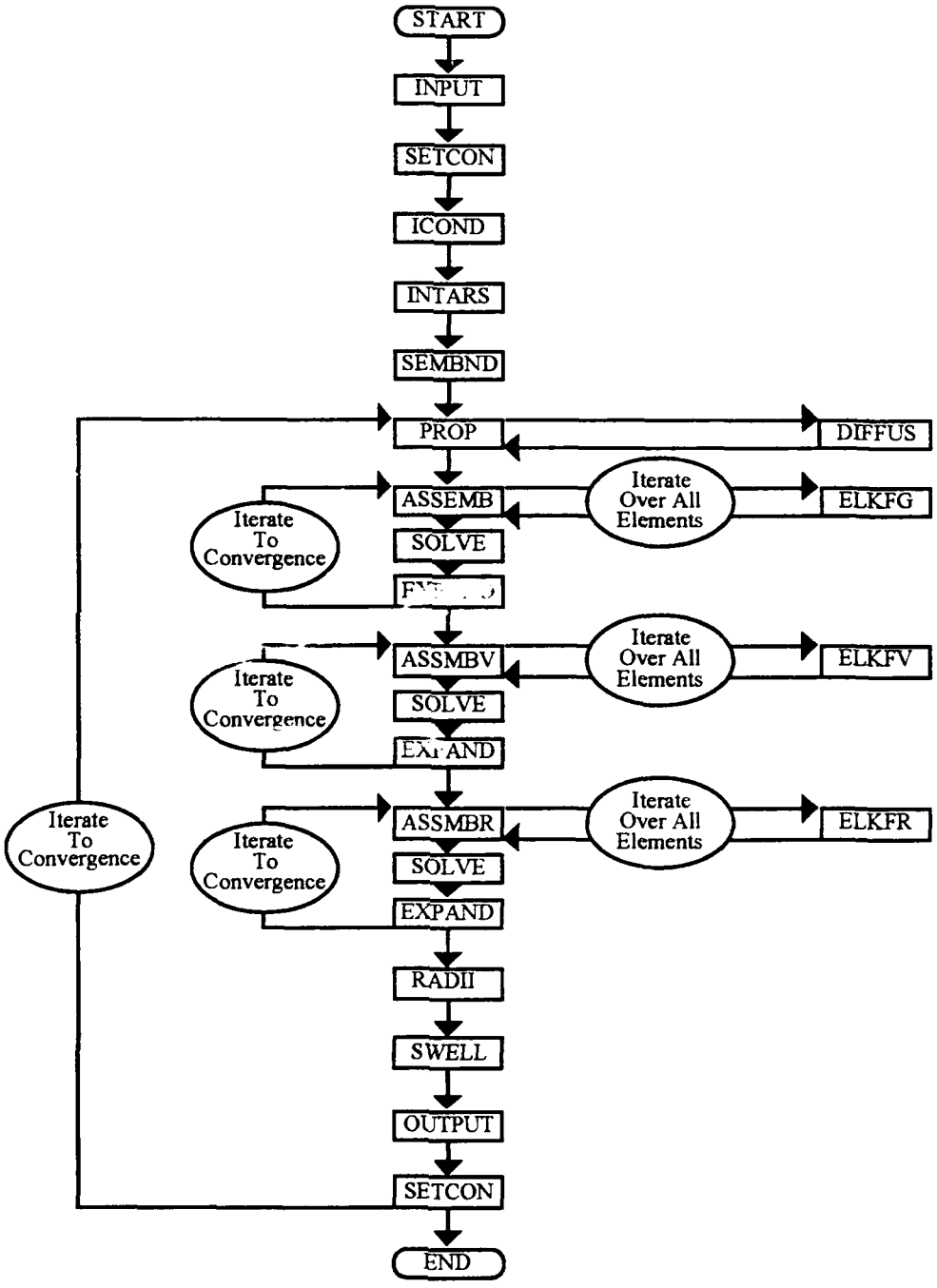


Fig. 2 REDSTONE Program Flow.

7.2 Subroutines

Program REDSTONE is made up of 27 subroutines and functions. Each of these are listed below with a short description of their functions.

7.2.1 ASSEMB

Assembles the global matrix and force vector to be solved for bubble concentrations. The global stiffness matrix is produced in a banded, non-symmetrical form with specified concentration rows and columns (those on the grain surface) eliminated but still accounted for. The bandwidth of the global matrix is twice the number of bubble sizes times the number of nodes in an element.

7.2.2 ASSMBR

Assembles the global matrix and force vector to be solved for bubble radii. Again, the global stiffness matrix is produced in a banded, non-symmetrical form with specified radii rows and columns (those on the grain surface and gas atoms) eliminated but still accounted for. The bandwidth of the global matrix is twice the number of bubble sizes minus one times the number of nodes in an element. The number of bubble sizes is reduced by one because the radius of single gas atoms is not variable.

7.2.3 ASSMBV

Assembles the global matrix and force vector to be solved for bulk vacancy and interstitial concentrations. The global stiffness matrix is produced in a banded, non-symmetrical form with specified radii rows and columns (those on the grain surface) eliminated but still accounted for. The bandwidth of the global matrix is twice the number of nodes per element times two.

7.2.4 DIFFUS

Calculates diffusion coefficients for single gas atoms, bubbles, vacancies, and interstitials.

7.2.5 ELKFG

Assembles the elemental stiffness matrices and force vectors of bubble concentrations for subsequent integration into the global matrix and vector by ASSEMB. The governing equations are incorporated in this subroutine.

7.2.6 ELKFR

Assembles the elemental stiffness matrices and force vectors of bubble radii for subsequent integration into the global matrix and vector by ASSMBR. The governing equations are incorporated in this subroutine.

7.2.7 ELKFV

Assembles the elemental stiffness matrices and force vectors of vacancy and interstitial concentrations for subsequent integration into the global matrix and vector by ASSMBV. The governing equations are incorporated in this subroutine.

7.2.8 EQURAD

Calculates the equilibrium radius of a given bubble size with temperature and gas atom content as inputs.

7.2.9 EXPAND

Adds the iterative solutions into the respective change vectors describing the change over a time period.

7.2.10 ERROR

Prints the user specified error in the output.

7.2.11 FRTPAG

Prints the REDSTONE banner in the output.

7.2.12 ICOND

Calculates arrays describing connectivity between bubble groups at elemental nodes with bubble groups at global nodes. Used by ASSEMB and ASSMBR.

7.2.13 INPUT

Assigns data read in from the input file to respective variables.

7.2.14 INTARS

Calculates arrays describing correlations of all bubble groups to those which are not specified. Used by ASSEMB and ASSMBR.

7.2.15 MTXADD

Performs matrix addition on arrays.

7.2.16 OUTPUT

Prints results of calculations in character format.

7.2.17 PROP

Calculates fuel, fission gas, and bubble properties at the beginning of each new time step. Also calculates initial bubble radii.

7.2.18 RADII

Calculates new bubble radii based on calculation results.

7.2.19 RDCARD

Reads a line of input data from the input file and parses it.

7.2.20 RESINP

Reads the restart input file when restart is specified.

7.2.21 RESSAV

Saves data in machine readable form to a restart file after each time period. Allows subsequent runs to start at a specific point in time and with an already calculated concentration and radii profile.

7.2.22 RNORM

Calculates the norm of a vector.

7.2.23 SEMBND

Calculates the half-bandwidth of the bubble concentration stiffness matrix. Similar values for vacancy/interstitial concentrations and bubble radii are calculated in assembly routines.

7.2.24 SETCON

Adds the time period changes into the known values and sets changes to zero.

7.2.25 SHAPE

Calculates finite element shape functions.

7.2.26 SOLVE

Solves banded, non-symmetrical matrices and stores the result in the force vector. Does not check for positive definiteness.

7.2.27 SWELL

Calculates gas released and swelling using calculated results.

7.3 Input

Input to program REDSTONE is done through 80 column character records residing in an external file. An example can be found in Appendix 2. Each record is first scanned for a '*' or 'C' in column one to see if the record is a comment. If so, parsing is stopped and the next line read in for processing. If the line is not a comment it is first parsed into words (denoted by a breaking space). The first word is checked to see if it is a valid identifying line number. An invalid line number causes program execution to stop with an error. Numbers are used in the first word of a line to denote the type of data present. The numbers used are grouped by hundreds with subgroups by tens. Thus a three (to four) digit number is required with the first two (or three) digits being significant. The last digit can be used at the user's discretion. Once the data type has been determined, a check is performed to insure the right amount of data is present on the line. If so, the individual words are assigned to the respective variables and the next line read in for processing. After processing all lines, the assigned data is printed to the output file. The specific types of data input are reviewed below by identifying line number.

7.3.1 Card Number 100

Contains user specified header information. This information is not parsed, but is printed as is on the output file. Only the '1' in the line number is significant, leaving numbers 100-199 to be used by the user for headers.

7.3.2 Card Number 200

Can contain a variety of data depending on the line number digit in the tens place. An identifier for whether British or metric units are used is identified by '210'. The number of Gauss points to use for integration is identified by '220'. A 0/1 flag for whether the input deck is reprinted on the output is denoted by '230'. A 0/1 flag to indicate whether a restart file is to be read is denoted by '240'. Currently, the type of units identifier is not used. Program input and output are in cgs units. Both the '2' and the second digit in the line number are significant.

7.3.3 Card Number 300

Contains the number of elements, the number of bubble sizes per node, the number of nodes per element, the number of specified concentrations, the number of time periods, and the maximum number of iterations to allow. Currently, the number of nodes per element is hard coded into much of the program for three nodes per element, so '3' should be the only value for this variable. The maximum number of iterations is applied individually for each iteration cycle, not as an aggregate for all iterations. Only the '3' in the line number is significant.

7.3.4 Card Number 400

Previously used to input the connectivity array of local node numbers to global node numbers. These are now calculated in the program and thus number 400 is no longer used.

7.3.5 Card Number 500

Contains the location of nodes in centimeters from the center of the fuel grain. A separate line is used for each node. Only the '5' in the line number is significant.

7.3.6 Card Number 600

Contains specified concentration information. Each specified concentration occupies one line. Each line contains a number identifying the bubble size, the node of the specified concentration, and the value of the concentration in cgs units. Only the '6' in the line number is significant.

7.3.7 Card Number 700

Contains the radius of the fuel grain in centimeters. Only the '7' in the line number is significant.

7.3.8 Card Number 800

Previously used to input material type identifiers for multiple material types (i.e., different enrichments, etc). Currently much of the program is hard coded for only one material type. As a result only one material type is allowed and this line number is no longer used.

7.3.9 Card Number 900

Contains the fuel porosity, enrichment, and convergence limit to use. The porosity and enrichment are fractions less than one. The porosity is not currently used since it is an intergranular property. Only the '9' in the line number is significant.

7.3.10 Card Number 1000

Contains the time length of periods in seconds, the number of time steps to take in a period, and the interval in time steps which to output results. One line is used for each time period. Only the '10' in the line number is significant.

7.3.11 Card Number 1100

Contains period operational data, one line for each period. Each line contains the initial power density in Watts per cubic centimeter, the initial temperature in degrees Kelvin, the initial temperature gradient in degrees Kelvin per centimeter, the initial pressure in Pascals, the final power density, the final temperature, the final temperature gradient, and the final pressure. Only the '11' in the line number is significant.

7.4 Output

Program REDSTONE produces several forms of output. The primary output is a listing of calculational results at user specified intervals. An example can be found in Appendix 3. The first part of this listing is a 'REDSTONE' banner followed by the user input header statements. This is optionally followed by a reprint of the input file. The input data is then summarized and printed in tabular form. Along with the input data are tables of a few preliminary calculations such as

connectivity arrays. All the above information is printed only once. The rest of the output listing consists of the results of calculation at the end of time steps at intervals specified by the user. Each of the results listings begins with time step information, followed by the percent fission gas released and percent swelling. The calculated burnup in several forms is listed next followed by the power density, temperature, and pressure used. Finally, the bulk of the results listing involves the calculated bubble concentrations, radii, and pressures by node.

In addition to the above output, two other files containing the calculated results is written. One of these consists of only numbers separated by spaces and is suitable as input to other analysis programs. The data included in this file are the node number, the radius of the node, the bubble size, the bubble concentration, the bubble radius, and the bubble pressure on each line.

The second is a machine readable restart file which can only be read by program REDSTONE. This file is written automatically at the end of each time period and overwrites any previous information in the file. It contains the last calculated time period values only. This includes the convergence denominators, the total fission gas atoms produced, as well as the bubble, vacancy, and interstitial data, i.e. concentrations, radii, and pressures. As a check for restart, this file also contains the number of elements, number of bubble sizes, and number of nodes per element used. This information is compared with the input file upon restart to ensure that a compatible problem is being restarted. A non-compatibility results in an immediate program halt and an error.

CHAPTER VIII
RESULTS

8.1 Validation of Program REDSTONE Internals

Program results will never be valid if the program doesn't perform as intended. So that the internal consistency of REDSTONE could be checked, calculated results were compared with analytic solutions where possible. At least two analytic solutions are available for comparison. Both are linear, single bubble size solutions. The first is the steady-state solution and the second is a time dependent solution for simple diffusion with a constant source.

8.1.1 Steady-State Solution

For a single bubble group in 1-D spherical geometry, the steady-state equation is

$$\frac{D}{r^2}\frac{\mathrm{d}}{\mathrm{d}r}\left(r^2\frac{\mathrm{d}C}{\mathrm{d}r}\right)+K=0 \qquad (198)$$

where D = diffusion coefficient, K= production source, and C = concentration. The boundary conditions are

$$\frac{\mathrm{d}C}{\mathrm{d}r}=0 \text{ at } r=0 \qquad (199)$$

and

$$C=0 \text{ at } r=R_g \qquad (200)$$

where R_g is the grain radius. The analytic solution to this equation is

$$C=\frac{K}{6D}\left(R_g^2-r^2\right). \qquad (201)$$

The constants K, D, and R_g can be defined with reasonable values, i.e. K=4.104 $\times 10^{11}$ $s^{-1}cm^{-3}$, D = 1.38×10^{-18} cm^2s^{-1}, and R_g = 10 μm. Using these values

in program REDSTONE and in the analytic solution results in the comparison shown in fig. 3. Exact agreement is obtained by REDSTONE as would be expected for a strictly linear smoothly varying solution.

8.1.2 Time-Dependent Solution

The problem that was solved above was simple. A more difficult problem involves the time dependence since this will introduce near discontinuities in the slope of the solution at the grain boundary. The time dependent equation for a single bubble group in 1-D spherical geometry is

$$\frac{\partial C}{\partial t} = \frac{D}{r^2}\frac{\partial}{\partial r}\left(r^2 \frac{\partial C}{\partial r}\right) + K \ . \qquad (202)$$

The same boundary conditions are used as before with the added initial condition of $C = 0$ at t=0. For these conditions, a series solution can be obtained [6], i.e.

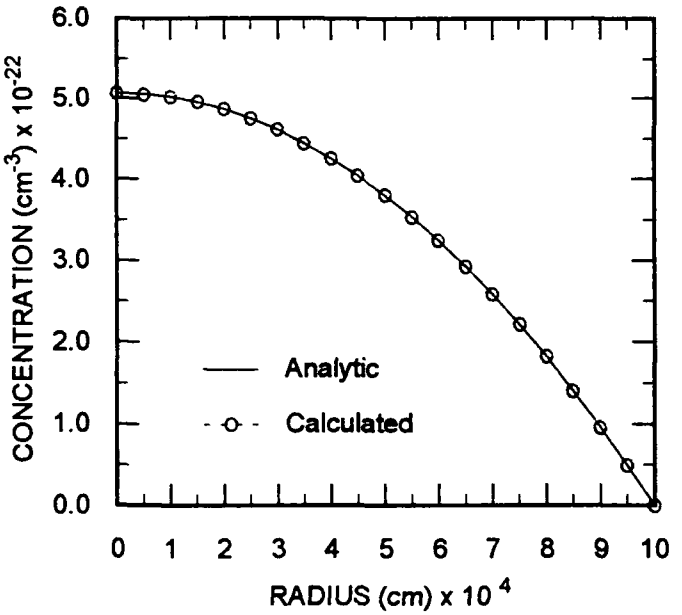


Fig. 3 Linear Steady-State Comparison with Analytic Solution

$$C = \frac{K}{6D}\left(R_g^2 - r^2\right) + \frac{2KR_g^3}{D\pi^3 r}\sum_{n=0}^{\infty}\frac{(-1)^n}{n^3}\sin\left(\frac{n\pi r}{R_g}\right)\exp\left(-\frac{Dn^2\pi^2 t}{R_g^2}\right). \qquad (203)$$

Calculations using REDSTONE for time = 3×10^5 secs are compared with the series solution in fig. 4. Although the REDSTONE calculation represents the series solution well near the center of the grain, it oscillates wildly at the outside of the grain. This oscillation is caused by the inability of the interpolation functions to adequately represent the behavior of the concentration in a region of sharp change and is similar to shock behavior in mechanical modeling. Ideally, one would want interpolation functions which followed the profile of the concentration exactly. This is not generally possible for real world problems. However, one can improve the approximation of the concentration profile in several ways, two which are used here.

The first improvement involves adjusting the interpolation functions in the outermost element to better approximate the sharp decline in concentration at the grain boundary. Burnett [120] notes that the slope of three node interpolation

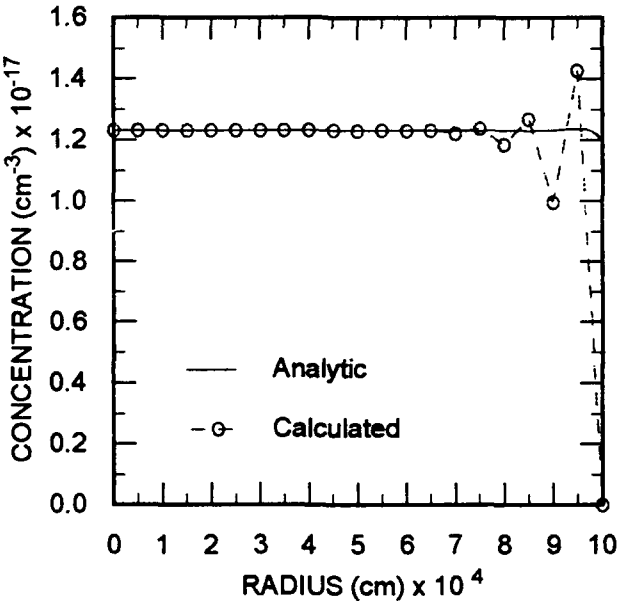


Fig. 4 Linear Transient Comparison with Series Solution

functions at one edge of an element can be made infinite by moving the central element off center to a position 0.75 of the distance between the two outer nodes. This arrangement and the effect on the interpolation functions are shown in fig. 5. The effect on the results from REDSTONE are compared with the original REDSTONE calculation in fig. 6. A noticeable lessening of oscillatory behavior has occurred. However, oscillations remain because the interpolation functions still do not exactly represent the concentration.

A second way to improve the representation is to increase the number of elements and nodes in the vicinity of the change. This was tried for REDSTONE using the element scheme as outlined in fig. 7. Although the same number of elements as used previously are present, their spacings have changed so that much

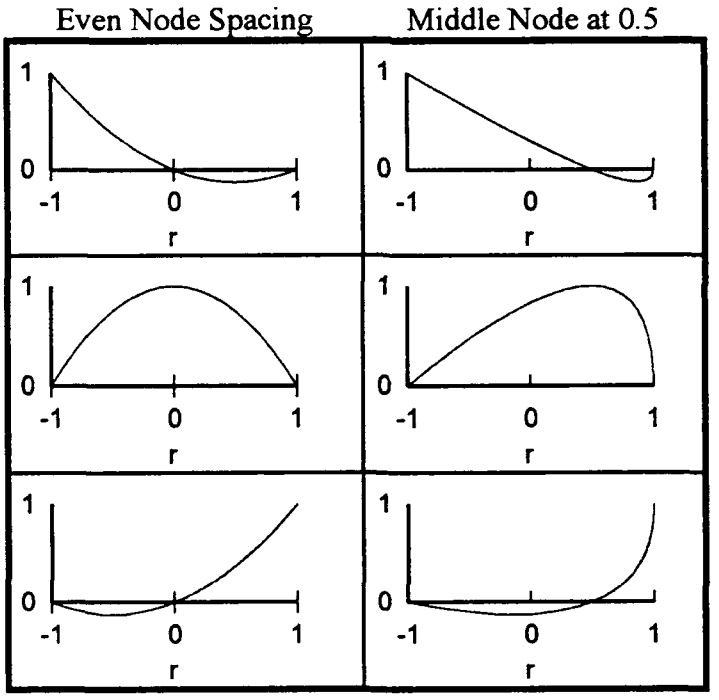


Fig. 5 Change in Interpolation Functions with Middle Node Shift

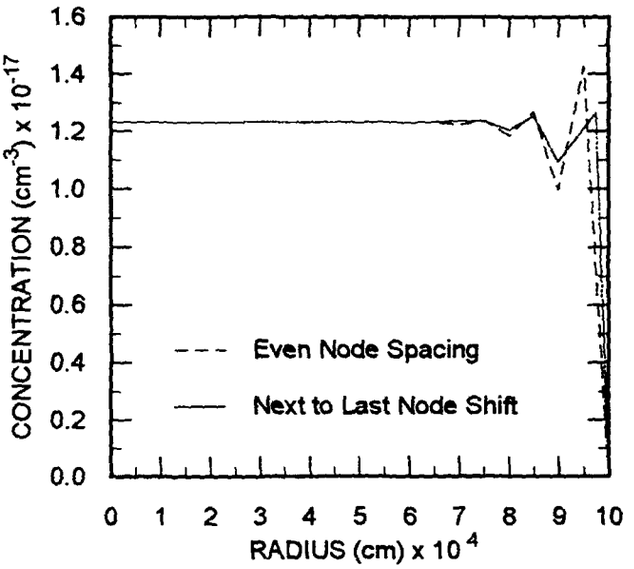


Fig. 6 Comparison of Results with and without Node Shift

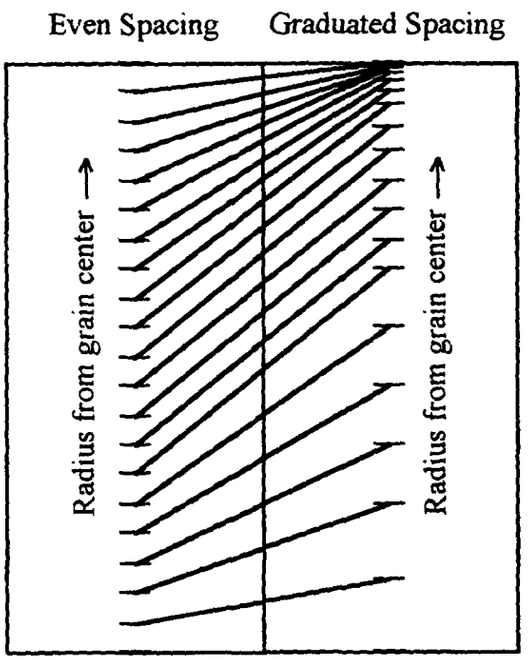


Fig. 7 Graduated Node Density Spacing in Fuel Grain

more detail can be modeled near the grain boundary. The elements sizes gradually diminished from inside the grain to the surface. The REDSTONE calculation using this elemental scheme (and the node shift) is compared to the previous REDSTONE calculations in fig. 8. Oscillatory behavior has now become nearly negligible, being reduced by a factor of at least 1000. Increasing the nodal density in the region of change has caused a significant improvement in REDSTONE's ability to model actual distribution profiles.

It is apparent from the above analysis that a combination of correct finite element programing and an intelligent approach to the problem are required to achieve acceptable results. Through the methods presented above, REDSTONE has been shown to be capable of adequately representing the problem at hand.

8.2 Validation of Program REDSTONE Results

A program may work well yet still provide unacceptable results if the correct equations are not programmed. Comparison with known experimental results provides indications of how well a program models the real world. Those comparisons for REDSTONE are presented here.

8.2.1 Additional Constraints

Initial calculations of multi-bubble problems utilized the full radius dependent formulations of bubble radii change presented in Chapter VII. It was quickly realized that the bubble radii calculations were extremely sensitive to changes in the radii. If bubble radii changed too much in a given time step, the program would begin to diverge. Substitution of the simpler partial-radius dependent formulation improved the stability of the program by slowing the rate of change. However, the bubble radii calculation was still very sensitive to the size of

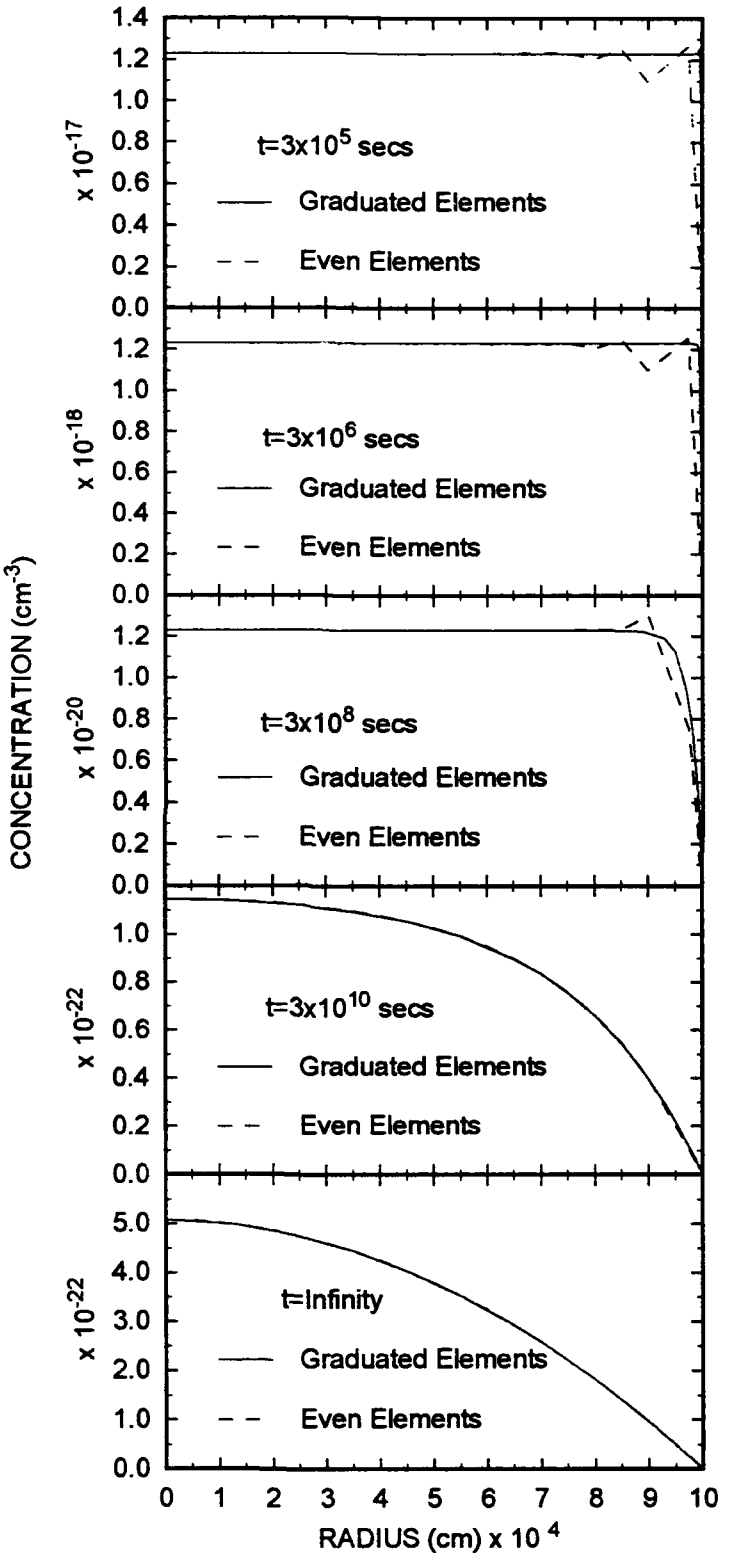


Fig. 8 Comparison of Results With and Without Graduated Node Density

change. A contributing factor to program instability was found in the calculation of bubble pressures. Due to the form of the reduced van der Waals equation of state, it was possible to calculate negative pressures if the radii became too small. For the small highly pressurized bubbles here, this occasionally happened in the course of calculations. These negative pressures affected calculations through the vacancy emission terms in both vacancy concentration and radii calculation. The bubble pressure was found to be very sensitive to bubble radius in this region and the overly-small radii were always only a few percent away from a radius that would produce a positive pressure. This problem was solved by limiting the pressure of a bubble to a maximum pressure. The yield pressure is a natural choice for this maximum pressure. Unfortunately, the yield pressure of UN as measured in the laboratory exhibits strain hardening, making if difficult to ascertain what the appropriate value to use *within the grain is*. Because of this a crude approximation of the yield pressure was made by using the curves of Werner and Blank [84], as mentioned in Chapter III. The hope was that these curves could at least provide a temperature dependence and an indication of the pressure needed for plastic deformation. Fortunately, the bubble pressure is very sensitive to radius in this region. Conversely, the radius is insensitive to pressure. Thus large changes in pressure do not significantly affect the radius in the region near the yield point. A few preliminary calculations using REDSTONE proved this to be the case. Thus the correlation presented in Chapter III was successfully used to eliminate problems with negative calculated pressures.

This, however, did not eliminate the sensitivity of the radii calculations to changes in bubble radii, especially during the severe transient to be modeled here. It was found that the vacancy emission term was still the destabilizing term. The

unique formulation of this term due to the pressure dependence may have caused a loss of too much of the non-linearities. This could have then resulted in the inability of the correction term to drive the solution in the correct direction when changes became too large. It was found that the calculations could be stabilized by limiting the maximum size of the bubbles to the equilibrium size. This was an acceptable constraint since bubbles would be expected to remain at or below their equilibrium size in steady-state or ramp-up conditions. However, in a ramp-down condition, bubbles are likely to become underpressurized resulting in sizes larger than the equilibrium size. Enforcing a maximum bubble size constraint in this situation causes the bubbles to attain their equilibrium size which might affect the calculated results.

8.2.2 Comparison with Experimental Studies

With program calculations stabilized, REDSTONE was ready to compare with experimental studies. Unfortunately, the number of experimental studies using UN are a small group. In addition, those studies measuring the types of data needed for comparison to intragranular code predictions are even fewer. No studies of UN were found where a microscopic determination of swelling components were made. Studies which were found included only those measuring macroscopic swelling and gas release. Most of these studies, especially the more recent ones, have dealt with fuel temperatures much higher than the those of interest here. This further reduces the candidates for comparison since program REDSTONE has the added restriction that the species modeled cannot be too mobile. Large bubbles are produced when the species are mobile enough for significant coalescence to occur, as with high temperatures. This is a restriction because only individual bubble sizes are modeled

rather than bubble groups with averaged properties. As a result, large bubbles can only be modeled using very large program arrays, which take up large amounts of computer storage and exponentially increase the running times. Local computer constraints restricted the size of the problem here to 50 bubble sizes. Even at 50 bubble sizes, the running times increased over six-fold when the number of bubble sizes was doubled. Thus, only studies utilizing lower temperatures, shorter burnups, or both were candidates for comparison.

When comparing REDSTONE results to experimental release and swelling studies, one should keep in mind that these studies are based on release from fuel pellets, not individual fuel grains. In general, since gas released from the grain will be collected on grain boundaries before release from the pellet, the release reported in the studies should be lower than that predicted by an intragranular code. However, some minimum amount of release would be expected from secondary knockouts and recoil atoms on the pellet surface and in any open porosity. Direct recoil release will come from fissions within approximately 10μm of the pellet surface, while knock-ons, with less energy, will only come from about 200 Å deep into the pellet [5]. Olander derives theoretical equations for release rates for both these mechanisms. After noting that the equations only predict a 6% release from 10% burnup fuel in a 0.6 diameter pellet, he concludes that these mechanisms are not very important in reactor fuel-element performance. However, this calculated release is equivalent to 0.6% release per atom% burnup, which although low, is significant when comparisons are to be made. While Olander's result is only a theoretical release rate, it does indicate that very low, but measurable, releases of fission gas from fuel could be explained entirely by recoil and knockout mechanisms.

With regards to swelling, it needs to be noted that swelling at the grain boundaries and from non-gaseous fission products are not included in REDSTONE's calculations. Olander [5] has stated that this additional volume swelling amounts to between 0.15 and 0.45% per atom% burnup for UO_2, depending on the stoichiometry of the fuel. A similar range can be assumed for UN. Additionally, any significant release to the grain boundary will cause larger swelling rates than in the interior of the grain. Thus, swelling predictions by REDSTONE should be less than in experimental studies, with better agreement obtained for low grain boundary release situations.

Two studies were found which satisfied the lower temperature, lower burnup criteria. One was a swelling study performed at the National Aeronautical and Space Administration (NASA) in 1973 [121]. The second was a release and swelling study performed at the Japan Atomic Energy Research Institute (JAERI) in 1977 [122]. In addition, Storms [123] has correlated experimental release data to theoretical density, burnup, and temperature for releases between 0.1 and 15%. He compared his correlation with similar correlations by Ross [124], Baars [125], and Thomas etal [126]. He notes that his correlation, like others, tends to underpredict release for large releases. Although he does not discuss it, his correlation also overpredicts release for low burnup, low temperature situations as is the case here. This is also true for the correlation by Thomas et al, but not true for those by Ross and Baars. Partly this is because there is a large scatter in the data, and the various sets of data used by each were slightly different. However, Storms also discarded from his data all releases below 0.1% regardless of irradiation temperature. This eliminated most of the low temperature, low burnup data, i.e 85% of the data below 1300°K / 3% burnup, and 70% of the data below 1500°K / 3% burnup. Additionally, he includes

data from U-Pu-N, as does Thomas et al., which typically showed much larger releases at the same temperatures and burnups. As a result, these latter two correlations are skewed to predict larger releases than experiment has shown in this low temperature, low burnup region. It should be noted, however, that the amount of error introduced by these correlations is not generally important since the releases are low. In consideration of the above, rather than use a correlation to compare with REDSTONE results, the data tabulated by Storms will be used to indicate the range of release expected experimentally for each of the studies below.

8.2.3 Comparison with NASA Results

The study at NASA measured swelling of six UN fuel pins containing four pellets each which were irradiated for approximately 4000 hours to burnups of up to two percent. The average cladding surface temperature was 1100°K. However, there was some degree of thermal cycling due to reactor reloads. Swelling data was obtained by removing three pellets from two of the pins for measurement. Of these, one had significantly lower burnup which allowed the results to be compared with REDSTONE calculations. Reported fuel characteristics and operational conditions for this pellet are given in table 5.

The NASA swelling results are compared with the REDSTONE result in table 6. Also in table 6 are REDSTONE release data and Storms' data for burnups between 0.74 and 1.36% and temperatures below 1400°K. In performing the REDSTONE calculation, the fuel grain size and exact operating history were not known. The grain size used was 20μm diameter and no thermal cycling was considered.

Table 5

Fuel Pellet Characteristics of NASA Study

Outside Diameter (cm)	0.3807
Length (cm)	0.6363
Enrichment (%)	10
Theoretical Density (%)	94.3
Power Density (W/cm^3)	724
Average Temperature (°K)	1222±28
Burnup (%)	0.948

Table 6

Comparison of NASA Results with REDSTONE Results

	% ΔV/V	% Release
NASA	0.99 -1.8	---
Storms	---	0.01-0.10
REDSTONE	0.367	0.0567

The values obtained by REDSTONE are lower than other values in table 6. Some of this difference might be due to the thermal cycling in the operating history. Conceivably, the thermal cycling could have involved short periods of higher temperature causing greater coalescence of bubbles, greater release, or intra-pellet cracking which could have reduced the overall density of the pellet and conversely increased volume. Additionally, the chemistry of the fuel pellets was not controlled to ensure that stoichiometric fuel was used. It was previously mentioned that other investigators were having trouble with non-stoichiometry around this period, which would have resulted in increased diffusional properties [123].

Some additional swelling was almost certainly contributed by non-gaseous fission products. When this additional swelling is considered, program REDSTONE's swelling result falls close to the range experimentally observed. The residual difference can be attributed to the factors mentioned above.

REDSTONE's gas release result falls in the middle of the data tabulated by Storms. One would expect release from the grain to exceed that from the fuel pellet, but in fact a small intragranular release may not be released outside the fuel pellet at all. It is likely that the smallest release results tabulated by Storms involved release by the recoil and knockout mechanisms, not by release from grains in the interior of the fuel.

8.2.4 Comparison with JAERI Results

The investigators at JAERI studied carbide, carbonitride, and nitride fuels irradiated to burnups of 24000 MWD/t at temperatures between 1264°K and 1634°K. Controllers were used to maintain temperatures within $\pm$50°K during irradiation. Three UN pellets were measured for both gas release and swelling. Two of these had low enough burnups to be compared with REDSTONE results. Reported fuel characteristics and operational conditions for these pellets are given in table 7.

The JAERI swelling results are compared with the REDSTONE results in table 8. Also in table 8 are Storms' data for burnups equivalent to that in the JAERI study, between 1.36 and 2.34% and temperatures below 1400°K. In performing the REDSTONE calculation, the fuel grain size used was 20μm diameter.

As with the NASA study, the REDSTONE results are on the lower end of the swelling range observed experimentally, although slightly better agreement is obtained. When non-gaseous swelling is considered the REDSTONE results fall in the middle of the observed swelling ranges. However, experimental swelling data can be seen to vary widely. This variation, besides making comparisons with REDSTONE results tenuous, indicates that in low fission gas swelling situations,

Table 7

Fuel Pellet Characteristics of JAERI Study

	Pellet 1	Pellet 2
Outside Diameter (cm)	0.325	0.345
Length (cm)	0.7	0.7
Enrichment (%)	20	5
Theoretical Density (%)	88-95	88-95
Power Density (W/cm^3)	1998	2224
Average Temperature (°K)	1169	1344
Burnup (MWD/t)	12000	16000

Table 8

Comparison of JAERI Results with REDSTONE Results

	% ΔV/V	% Release	% ΔV/V	% Release
JAERI (Pellet 1)	0.5-1.8	1.2	---	---
JAERI (Pellet 2)	---	---	0.7-2.4	0.7
Storms	---	0.09-1.91	---	0.09-1.91
REDSTONE	0.4619	0.0223	0.7200	0.0185

other processes are contributing as much or more to the overall swelling of the fuel. Some of the possible processes were discussed previously. Based on the comparisons made here, REDSTONE appears to be satisfactorily accounting for the intragranular fission gas component in the overall swelling. In general, however, it appears that REDSTONE results will be below experimentally observed swelling percentages by as much as 1.5%.

Fission gas release results are also compared in table 8, with REDSTONE results being below that of the JAERI study but within the experimentally observed range of Storms. Because the percentage release is low, the same arguments apply here as for the NASA study. Most experimentally measured release likely did not come from diffusion out of fuel grains.

8.3 Modeling of the Transient Scenario

The transient scenario of interest here involves sustained low temperature operation followed by a rapid surge in power by at least two orders of magnitude which is sustained for several minutes. The parameters of this scenario were discussed earlier. Table 9 lists those parameters which were used in REDSTONE to model the scenario.

Results of the REDSTONE calculation are tabulated in table 10. These results show that the swelling before the transient is small and that the transient has little effect on the overall swelling and release, presumably because the time at high temperature is short. The amount of gas released is insignificant, especially when release from other non-diffusional sources are considered. The post-transient swelling of 1.0649% corresponds to a diametral increase of 0.354%. If one adds to this the possible 1.5% swelling due to non-gaseous processes, the diametral increase is still only 0.848%. A check of the data in table 1 will reveal that the fuel must increase in diameter 4.6% before coming into contact with the liner, and assuming unrestrained swelling against the cladding, must swell 5.9% before causing design failure. Design failure was specified as 1% diametral increase of the fuel pin. Alternately, swelling of the fuel pin because of released gas is not indicated since the gas release from the grain is insignificant. It is clear that the results reported here do not indicate fuel pin failure will occur due to swelling or release of fission gases from the fuel.

Specific characteristics of the solution are illustrated in figs. 9-12. Fig. 9 shows the swelling and gas release during the transient and high temperature periods. As seen in fig. 9, the swelling initially increases with the transient, lagging slightly, then continues to increase. The percentage release shows no effect from the

Table 9

Transient Scenario Modeling Parameters

Fuel Characteristics		
Fuel	UN	
Theoretical density (%)	94.3	
Enrichment (%)	93	
Grain diameter (μm)	20	
Operational Characteristics		
Initial power (W/cm^3)	52.6	(equivalent to 10 kW electric)
Initial temperature (K)	500	
Initial temperature gradient (K/cm)	38	
Time at initial power (sec)	3×10^8	
Transient time (sec)	30	
Final power (W/cm^3)	5260	(equivalent to 1 MW electric)
Final temperature (K)	1500	
Final temperature gradient (K/cm)	2550	
Time at final power (sec)	1000	

Table 10

Transient Scenario REDSTONE Results

	Pre-transient	Post-transient
Volume Swelling (%)	1.0627	1.0649
Gas Release (%)	*	5.274×10^{-3}

* Pre-transient gas release was below the significance maintained by the computer.

transient and continues to decrease slightly. Not shown is the pre-transient time profile, but this would have shown an initial percentage release of gas which reaches a maximum and then begins a steady decrease once the bubble concentration has been built up enough to prevent significant diffusional loss from the grain. The bubbles do not contribute to any significant release, either before the transient or after.

An inspection of the characteristics shown in the other figures indicates that the swelling rate during the high temperature region, although small, is greater than that from simple deposition of gas atoms in the lattice. Fig. 10 shows the concentration profile in the center of the grain both immediately before the transient

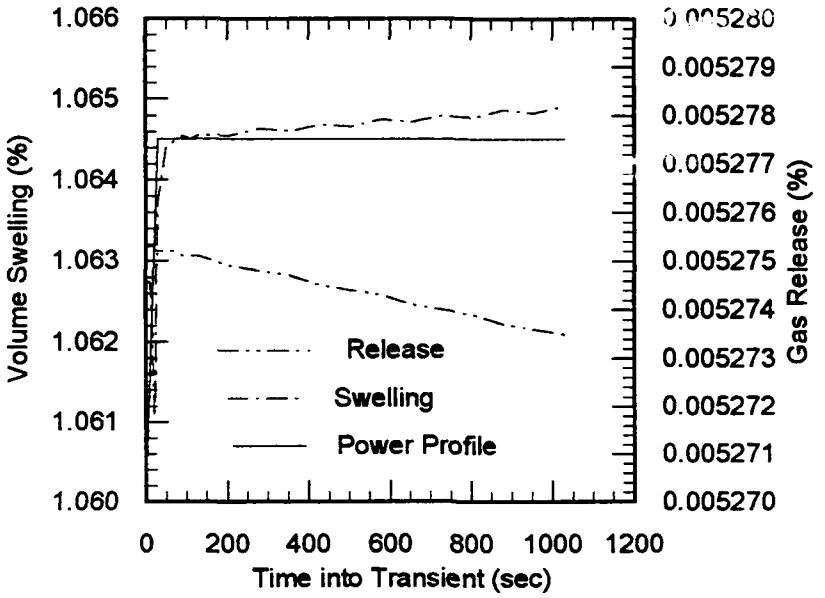


Fig. 9 Swelling and Release Percentages After Onset of Transient

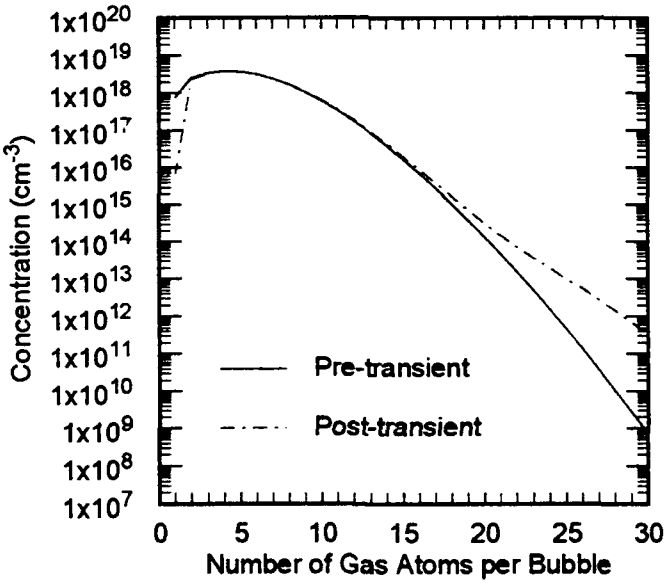


Fig. 10 Pre- and Post-transient Concentration Profiles

and at the end of the high temperature period. The concentration profile significantly changes only for single gas atoms and in the large bubble tail of the profile. The single gas atom concentration is becoming depleted, and the larger bubbles are beginning to grow in number. The highest concentration of bubbles are those containing four gas atoms both before and after the transient. It is clear that there is a shift occurring to larger bubbles. The question remains, however, on whether this translates into an increase in swelling. A check of the bubble radii shown in fig. 11, both before and after the transient, shows little change occurring, with only a slight increase for larger bubbles. This is supported by fig. 12 which shows the change in bubble pressures. The smaller bubbles have actually increased in pressure after the transient, as would be expected if the temperature rose but the radii remained relatively constant. The largest bubbles, on the contrary, started out overpressurized and actually decreased in pressure after the transient. This could only happen if the radii had become larger. The implications of the above analysis

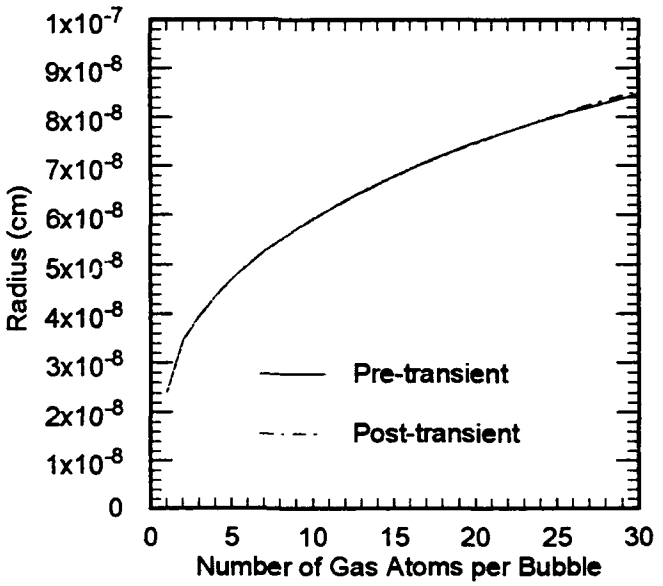


Fig. 11 Pre- and Post-transient Radii Profiles

is that swelling is increasing faster than the deposition of new gas atoms. While the changes in swelling are extremely small and not important for the scenario modeled here, they do indicate processes occurring which may become of importance at higher temperatures. To investigate this possibility, the scenario above was modeled at higher temperatures. This investigation is discussed below.

8.4 Variation of Fuel Temperature

An investigation into varying the fuel temperature was done by increasing the pre- and post-transient temperatures by 100°K, 200°K, and 300°K. Thus the pre-transient temperatures modeled were 500°K, 600°K, 700°K, and 800°K. The corresponding post-transient temperatures were 1500°K, 1600°K, 1700°K, and 1800°K. As shown in fig. 13, the swelling increases steadily with temperature both before and after the transient. However, the amount of increase is small and not important for the scenario modeled.

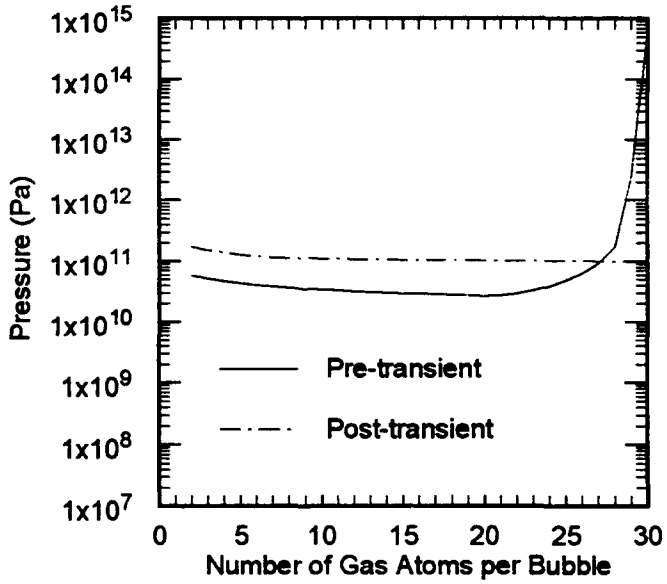


Fig. 12 Pre- and Post-transient Bubble Pressure Profiles

The associated gas release presented in fig. 14 shows only post-transient release as the pre-transient release was extremely small and essentially unaffected by the temperature. This lack of pre-transient temperature effect is supported by the pre-transient concentration profiles in fig. 15 which show virtually no change in profiles with temperature. The post-transient release also changes little with temperature except for the results at 1800°K. The small changes in release at 1500°K, 1600°K, and 1700°K indicates that bubbles still are not significantly contributing to release, and in fact are keeping the single gas atoms from being released. The larger release at 1800°K may indicate that the bubbles are becoming more mobile at this temperature.

A check of the post-transient concentration profile in fig. 16 supports this assertion. In fact, the concentration profiles at all temperatures show marked

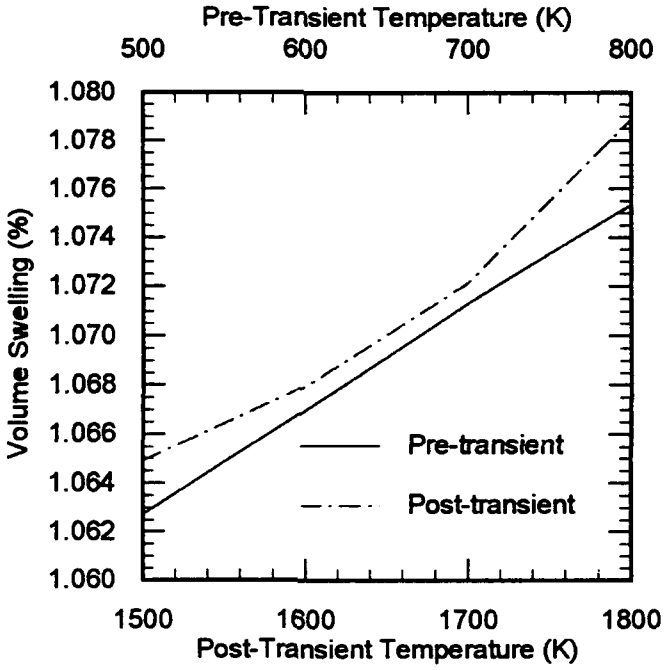


Fig. 13 Volume Swelling vs. Pre- and Post-transient Temperatures

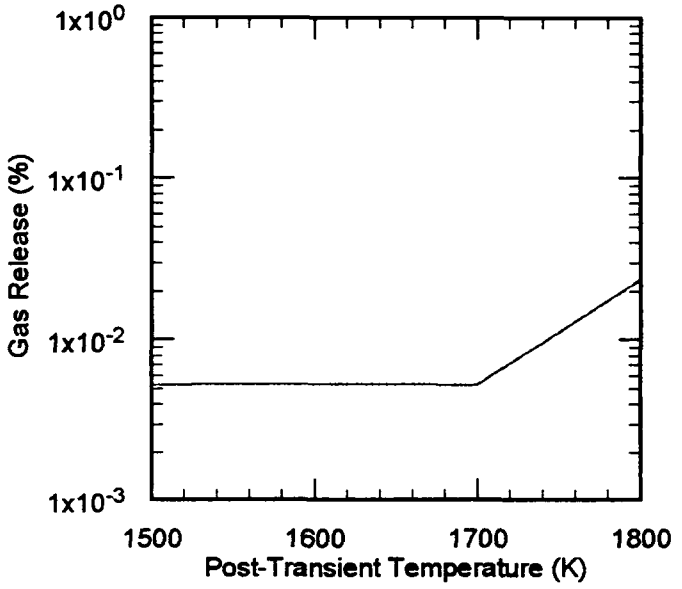


Fig. 14 Post-transient Gas Release vs. Temperature

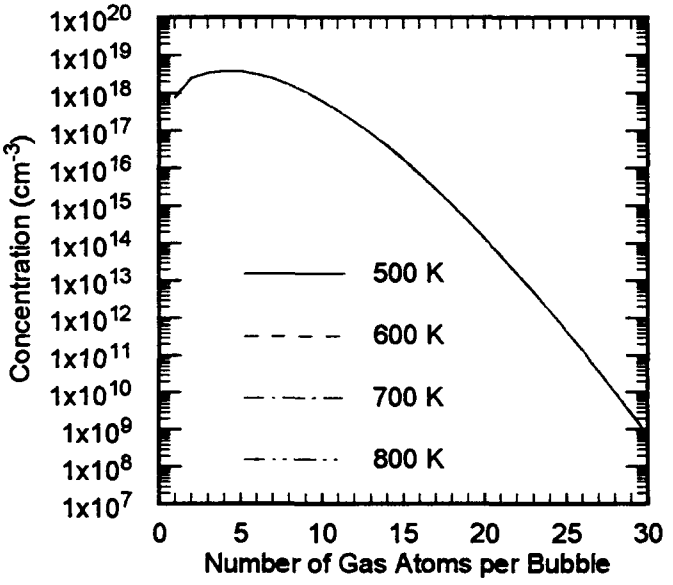


Fig. 15 Pre-transient Concentration Profiles vs. Temperature

differences between each other. The concentration of the largest bubble size increases significantly with temperature until, at 1800°K, it appears to be on the leading edge of an upward trend and is of the same order of magnitude as the highest concentration shown. Additionally, the profiles at 1700°K and 1800°K show oscillations at the smaller bubble sizes. This is possibly due to the increasingly depleted concentration of single gas atoms with higher temperature which induces a calculational shock to the solution. Increasing the number of time steps in the transient should reduce this oscillatory behavior.

Investigating the concentration profiles at the higher temperatures further, it appears that the bubbles are becoming too mobile for REDSTONE to be reliable at 1800°K and possibly at 1700°K. Fig. 16 would indicate that there are much larger bubbles at the same or higher concentrations at 1800°K. These larger bubbles have the potential to significantly increase the actual swelling over what REDSTONE has calculated. Even when the 1800°K scenario was run with 50 bubble sizes, the

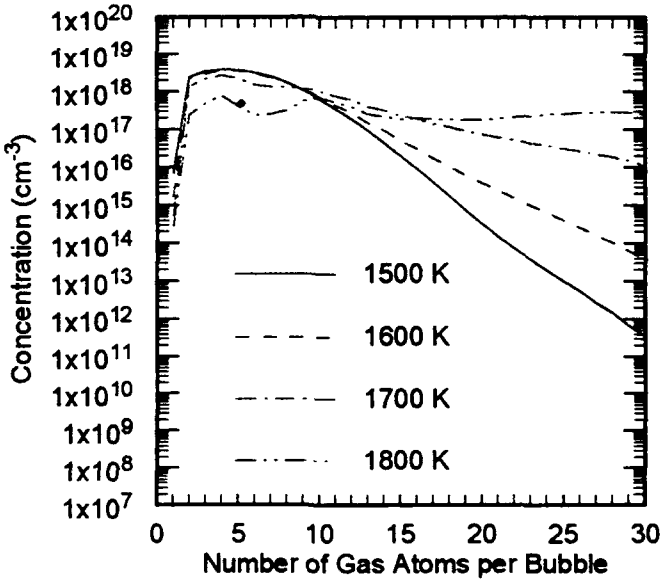


Fig. 16 Post-transient Concentration Profiles vs. Temperature

profile remained relatively flat. Without a reasonable indication that nearly all the bubble sizes are accounted for, it is questionable to rely on REDSTONE's results.

So that some measurable indication could be used, a figure-of-merit was developed to help discriminate when REDSTONE's results might be unreliable. This figure-of-merit was simply a ratio of the concentration at the maximum in the profile to the concentration in the largest bubble size. This is plotted in fig. 17. The effect of increasing temperature on post-transient results can clearly be seen in fig. 17. The question now is where to draw the line below which results are considered to be unreliable.

For the purposes here, a liberal definition of a reliable REDSTONE result was chosen. If the figure-of-merit was above 100, the result was considered reliable. Using this criteria, the results at 1800°K would clearly be unreliable, while those at

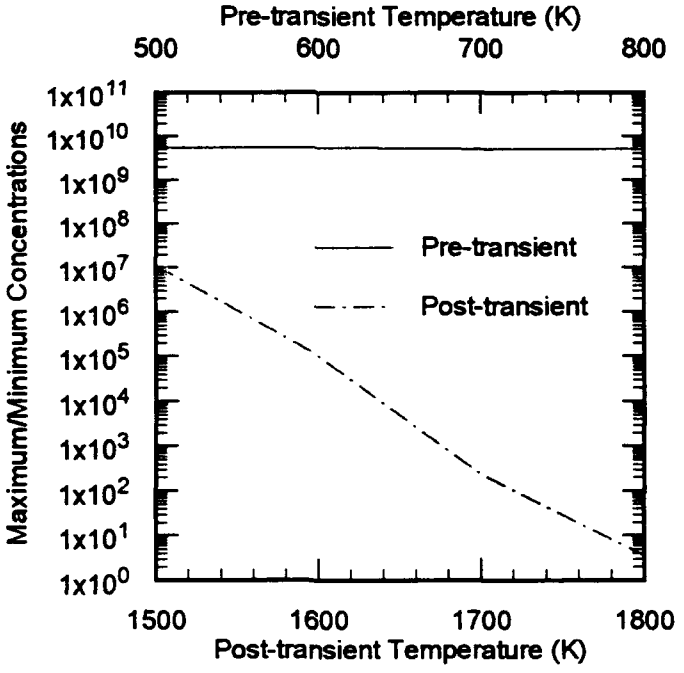


Fig. 17 Pre- and Post-transient Figures-of-Merit

1700°K, while close, would be considered reliable. The previously mentioned modeling of 50 bubble sizes at 1800°K resulted in an equally poor figure-of-merit as with 30 bubble sizes.

The most important observation of the above discussion is that the swelling and release behavior at 1800°K remains unknown. Indications from REDSTONE are that bubble mobility becomes appreciable at this temperature, leading to large bubble sizes and possibly significant swelling. This is important since scoping calculations indicated a fuel centerline temperature of around 1800°K, even though the average temperature of the fuel would remain cool enough that swelling would not be a problem. However, the calculations here assume a perfectly uniform system. In an operating reactor, fuel will vary in temperature by position in the core and possible imperfections in the fuel or cladding. Thus, some pellets might have average temperatures near or above 1800°K, and the temperatures would certainly increase if final power were boosted with minimal changes in design. Because this temperature is close to the temperatures expected to be present during the current transient scenario, knowing the gas behavior at 1800°K is an important design prerequisite. Unfortunately, REDSTONE is not readily capable of producing this without some sort of bubble grouping scheme. Additionally, in consideration of the large uncertainties in some of the input properties used in REDSTONE, one should not discount the possibility that increased bubble mobility actually becomes a problem at lower temperatures than predicted by REDSTONE.

8.5 Variation of Property Data

Sensitivity studies were performed to determine whether the uncertainties in properties could cause a significant effect on the results. The properties investigated

included the xenon, surface, and irradiation enhanced diffusion coefficients, the surface energy, and the resolution parameter. The same transient scenario investigated above was used except that a much coarser time stepping routine was used to speed program operation.

Each of the following discussions utilize a gas release plot showing post-transient gas release, a pre- and post-transient swelling plot, and the figure-of-merit plot discussed previously. Except for surface energy, the properties were varied through scaling by factors of 0.1, 1, 10, and 100. The resolution parameter was additionally scaled by 0.01. Surface energy was compared at 780, 1000, 1500, and 2000 ergs/cm^2.

8.5.1 Xenon Diffusion Coefficient

The gas release results obtained by scaling the xenon diffusion coefficient are presented in fig. 18. As expected, an increase in post-transient release is seen

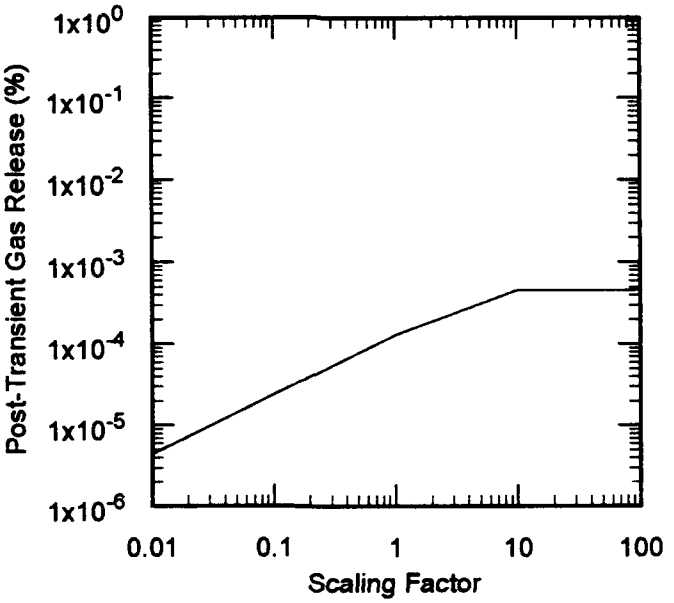


Fig. 18 Scaled Xenon Diffusion Coefficient Gas Release

with increased xenon diffusion coefficients. The reason for the leveling off for the highest diffusivities is unknown. Conceivably, this may have resulted from the quick depletion of gas atoms through release or capture.

The swelling results (fig. 19) show similar trends. As expected, the pre-transient swelling remains completely unaffected by the xenon diffusion coefficient since irradiation enhanced diffusion is by far the dominant process at the low pre-transient temperature. The post-transient results show the expected increase in swelling as increased gas atom capture occurs, but with an unexplained dip in swelling for the highest diffusivity. This dip corresponds with the leveling off in release results. A check of the figure-of-merits in fig. 20 indicates that all results should be reliable.

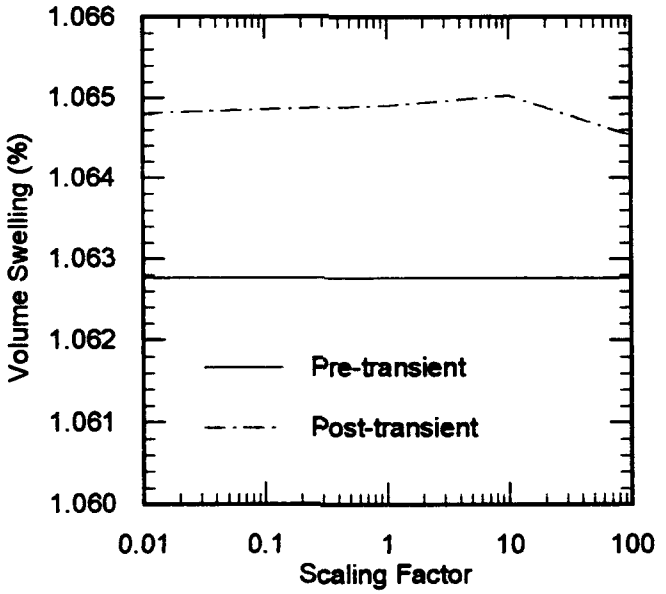


Fig. 19 Scaled Xenon Diffusion Coefficient Swelling

8.5.2 Surface Diffusion Coefficient

The gas release results obtained by scaling the surface diffusion coefficient are presented in fig. 21. Since bubbles do not appear to contribute to gas release, it was expected that the surface diffusion would have minimal effect on the gas release. This is generally true except for the highest diffusivity. At 100 times the surface diffusion, it appears that bubbles began contributing to the gas release. Again, however, the amount released is extremely small.

Because the release is small, one would expect that increased surface diffusion would result in more coalescences leading to higher concentrations of large bubbles and higher swelling. Indeed, this is seen to be the case in fig. 22. Again, because of the already very small mobilities at the low pre-transient temperature, the pre-transient bubble motion appears unaffected by increased surface diffusion. A check of the figure-of-merit (fig. 23) supports this assertion and

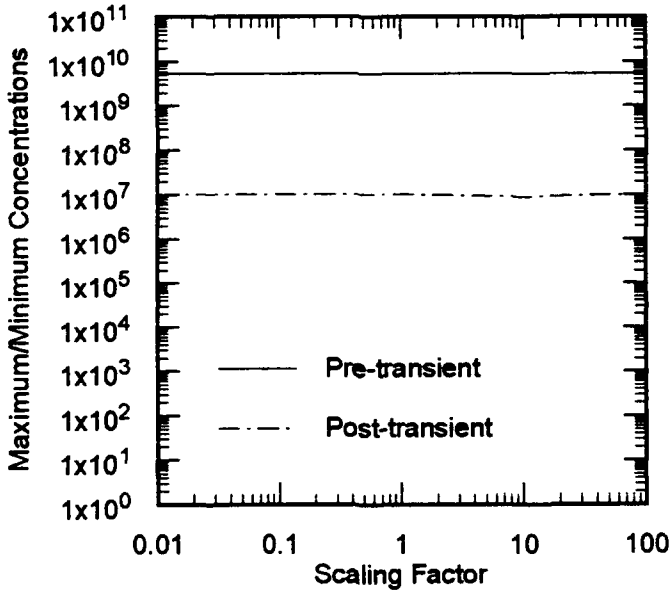


Fig. 20 Scaled Xenon Diffusion Coefficient Figures-of-Merit

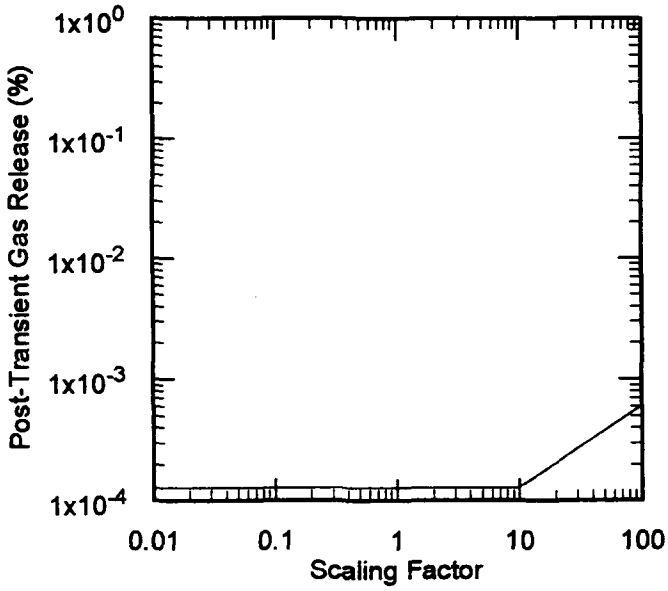


Fig. 21 Scaled Surface Diffusion Coefficient Gas Release

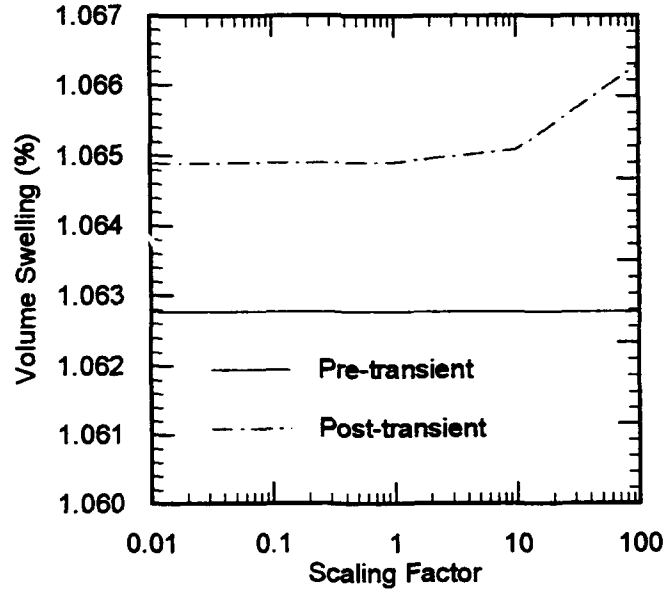


Fig. 22 Scaled Surface Diffusion Coefficient Swelling

the assertion that more coalescences are occurring. However, the figure-of-merit for the post-transient swelling at the highest diffusivity comes close to the reliability cutoff.

8.5.3 Irradiation Enhanced Diffusion Coefficient

The gas release results obtained by scaling the irradiation enhanced diffusion coefficient are presented in fig. 24. Although irradiation enhanced diffusion is important in the pre-transient region, release results in this region remained below the level of significance stored in the computer for all scalings of the diffusion coefficient. When the effect on swelling (fig. 25) is considered, it is seen that swelling increases with increased irradiation enhanced diffusion during the pre-transient period. This trend is the result of increased gas atom capture rather than release. The post-transient release results suggest this same trend since greater gas atom capture in the pre-transient period would result in a reduced gas concentration

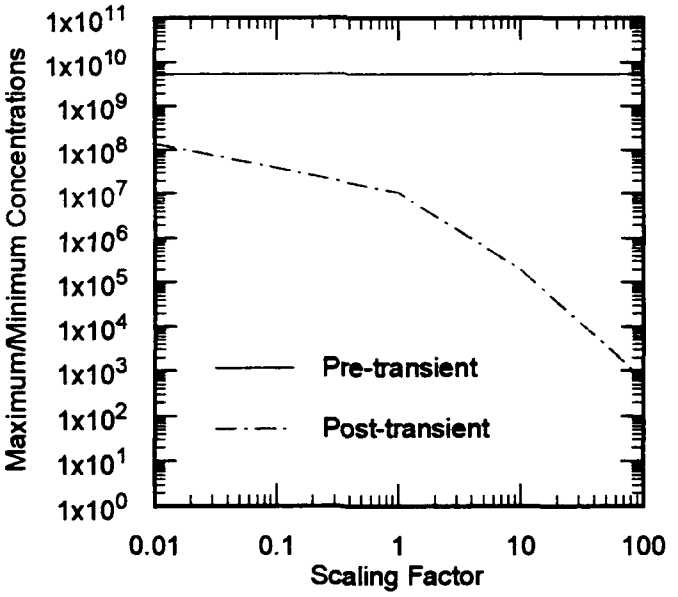


Fig. 23 Scaled Surface Diffusion Coefficient Figures-of-Merit

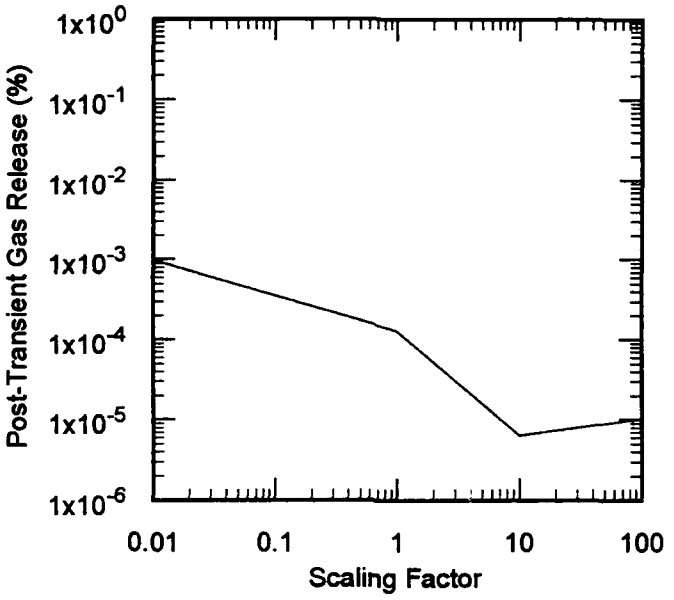


Fig. 24 Scaled Enhanced Diffusion Coefficient Gas Release

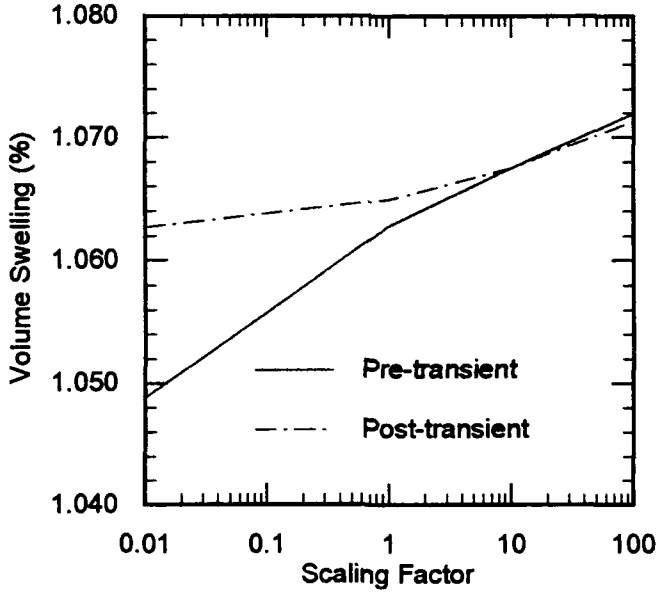


Fig. 25 Scaled Enhanced Diffusion Coefficient Swelling

being available for release after the transient. The bubbles do not appear to contribute significantly to release.

The swelling results shown in fig. 25 indicate both pre- and post-transient swelling increases with increasing irradiation enhanced diffusion. The pre-transient trend was discussed above. The post-transient swelling increases because of the corresponding increase in the pre-transient period. Interestingly, however, the post-transient results exhibit smaller increases compared to the pre-transient increases. If bubbles did not have time to move appreciably during the high temperature phase, one might expect a closer relationship between the two trends. Since they appear different, this could indicate that the post-transient swelling is only partially dependent on the pre-transient condition, and that coalescences after the transient play a significant role in setting up the post-transient profile. However, a check of the figures-of-merit (fig. 26) before and after the transient indicate that the results at he highest diffusivity are near the reliability limit, so there may be

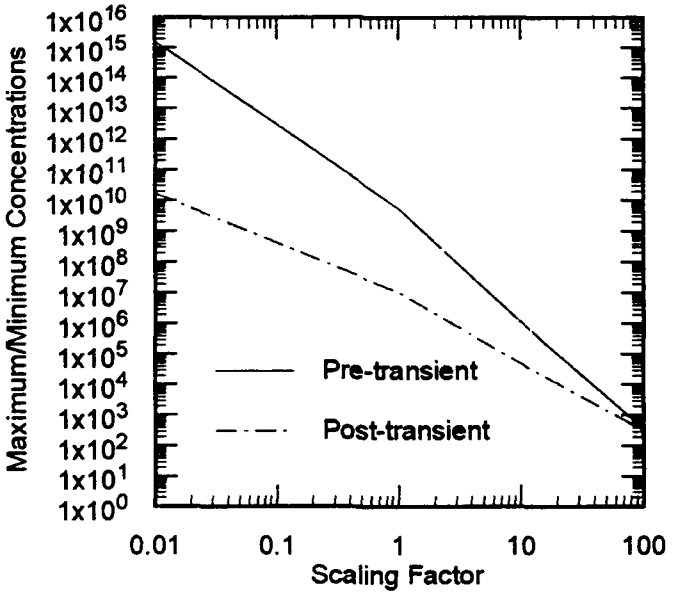


Fig. 26 Scaled Enhanced Diffusion Coefficient Figures-of-Merit

some uncertainty in the exact values calculated at 100 times the diffusion coefficient..

8.5.4 Surface Energy

The gas release results obtained by setting the surface energy at various energies are presented in fig. 27. Since the surface energy's main effect is in the determination of the size of the bubbles, it was not expected that gas release from the grain would be affected by changing the surface energy. This is demonstrated in the release results obtained.

The swelling results (fig. 28) clearly show the effect of changing the bubble sizes. As expected, increasing the surface energy decreases the swelling, and decreasing the surface energy increases swelling. There does not appear to be any significant effect on bubble mobility since the two curves in fig. 28 exhibit similar trends. A check of the figures-of-merit (fig. 29) indicates that all data are reliable.

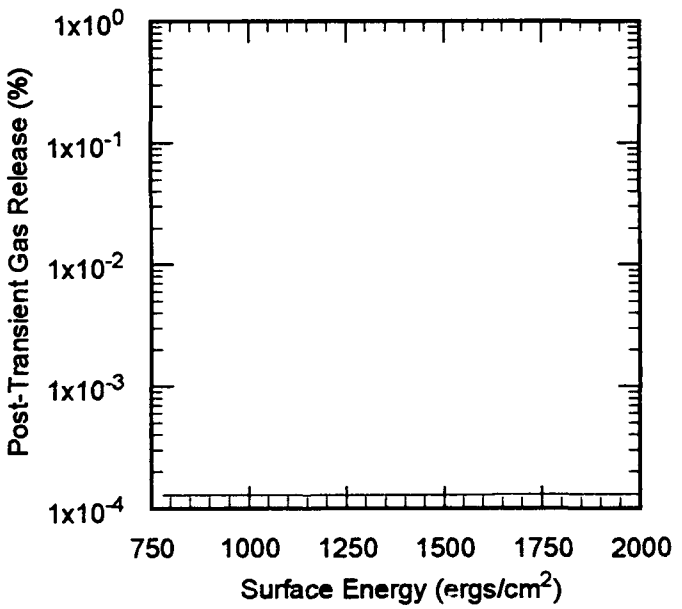


Fig. 27 Surface Energy Gas Release

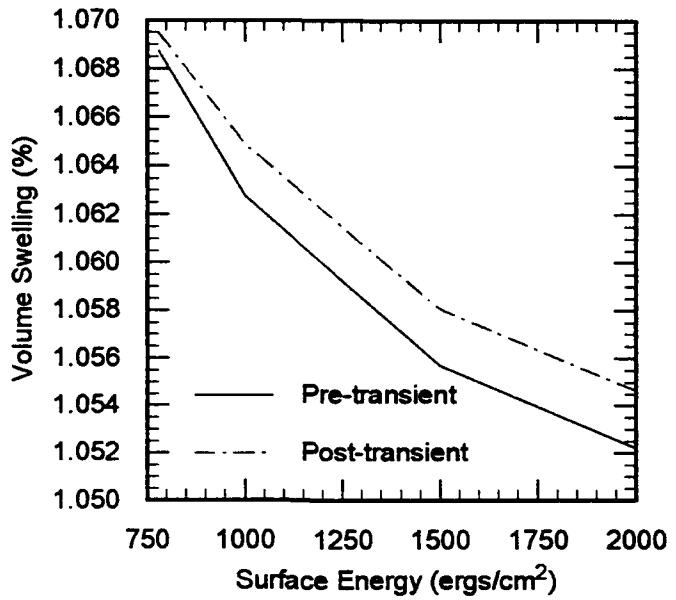


Fig. 28 Surface Energy Swelling

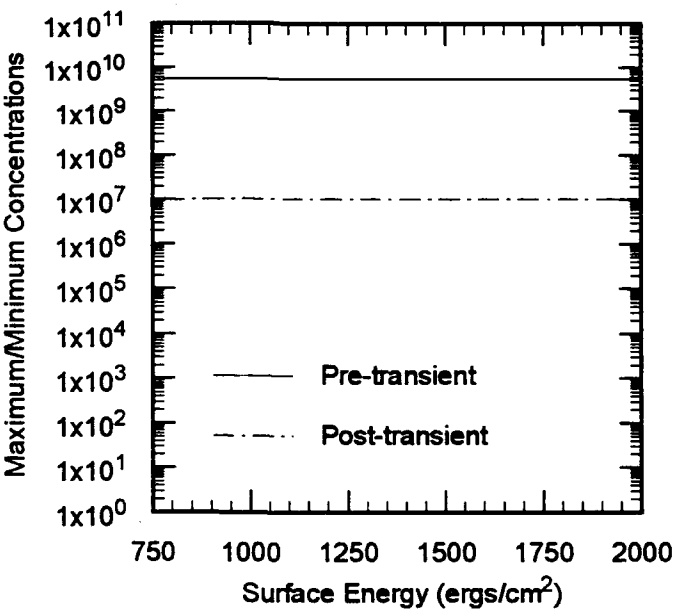


Fig. 29 Surface Energy Figures-of-Merit

8.5.5 Resolution Parameter

The gas release results obtained by scaling the resolution parameter are presented in fig. 30. A significant effect on release can be seen. Since the resolution parameter used here is one to two orders of magnitude lower than experimentally observed for UO_2, these results show that UO_2 equivalent resolution results in a four order of magnitude increase in release. The amount of release is still not large enough to be of importance. The cause of the increase is easily understood. As resolution increases, more gas atoms are kept free in the lattice. These single gas atoms are then more likely to diffuse out of the grain than be captured and kept by a bubble.

A look at the swelling results in fig. 31 show interesting trends. The pre-transient swelling exhibits a maximum at a scaling factor of one with decreased swelling as one either increases or decreases the resolution rate. The decrease in

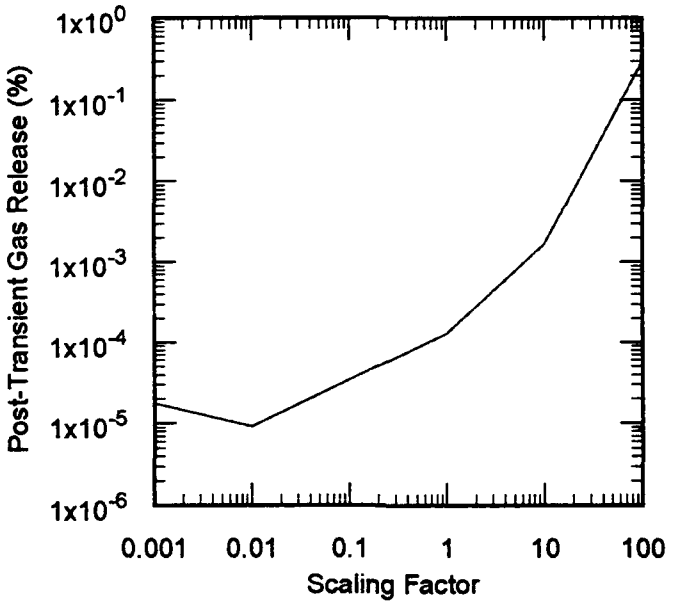


Fig. 30 Scaled Resolution Parameter Gas Release

swelling with increasing resolution is easily understood. The decrease in swelling with decreasing resolution is confusing until one checks the associated figures-of-merit in fig. 32. From these, it is clear that the concentration profile is flattening as resolution decreases. The lowest resolution results are clearly unreliable, and the next-to-lowest results are suspect. The pre-transient result at 0.1 cannot be eliminated based on the criteria that has been set up. The reason for this decrease is not clear. Since the bubbles are essentially frozen at the low pre-transient temperature and thus bubble concentration grows primarily through gas atom capture, one could argue for some decrease in swelling with lower resolution if the single gas concentration became so low that few were left to adequately develop a significant concentration of large bubbles. However, this argument seems tenuous, at best.

Considering, now, the post-transient swelling one sees that it exhibits a much milder dependence on resolution. As expected, it decreases with increasing

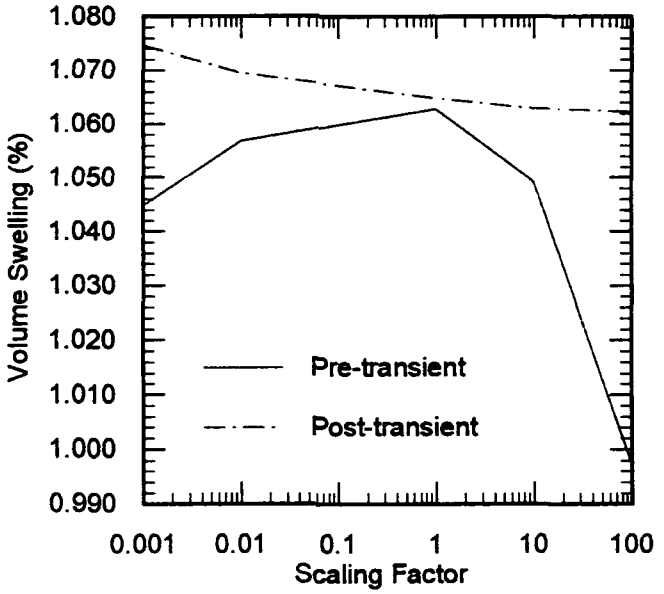


Fig. 31 Scaled Resolution Parameter Swelling

resolution rate. However, it appears that this is further evidence that post-transient mobilities may have had an effect in setting up the post-transient concentration profile.

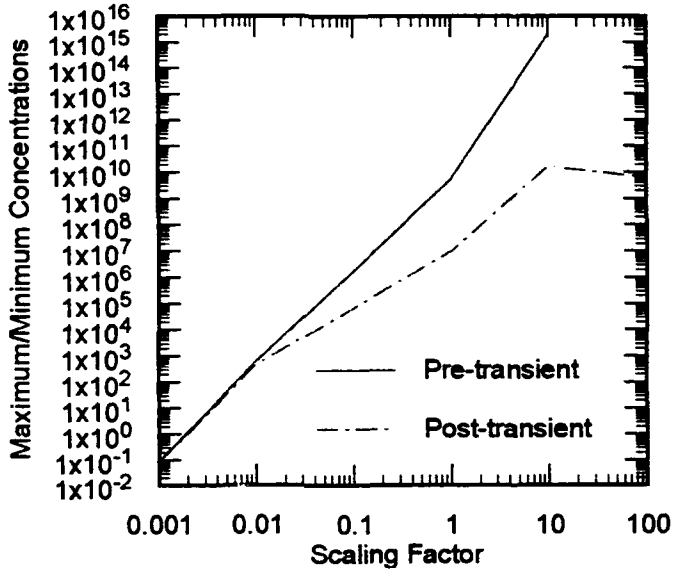


Fig. 32 Scaled Resolution Parameter Figures-of-Merit

CHAPTER IX
SUMMARY AND CONCLUSIONS

9.1 Summary

In order to investigate whether fission gas swelling and release would be significant factors in a space based nuclear reactor operating under the Strategic Defense Initiative (SDI) program, a finite element program was developed to model the transient fission gas behavior in a single uranium nitride fuel grain. Only the intragranular problem was investigated because of the importance of the intragranular fission gas as a precursor to all other fission gas based processes, and because it was necessary to limit the investigation to a reasonably sized problem.

The program developed was named REDSTONE (Routine for Evaluating Dynamic Swelling in Transient Operational Nuclear Environments) and was based on formulation of the governing 1-D, spherical geometry diffusion equations using the Galerkin method. The equations were incrementalized so that non-linear dependences would be retained, and the solution was obtained through iteration. Calculations were divided into three sets of governing equations: bubble concentration changes, bulk vacancy and interstitial concentration changes, and bubble radii changes. The bubble concentration and radii equations were derived from considerations of production and loss mechanisms of individual bubble groups defined by the number of gas atoms in the bubble. This is in contrast to similar investigations for UO_2 which have utilized groups with average properties describing bubbles containing a range of gas atoms. The choice to model individual bubbles placed limitations on the types of scenarios REDSTONE could model. The limitation was due to the increased storage requirements and running times needed

to model a large range of bubble groups. Since higher temperatures and longer burnups cause larger bubbles to form, REDSTONE's calculations were limited to those scenarios where low temperatures, low burnups, or both were present.

The bubble concentration equations incorporated the production and loss mechanisms of diffusion, production, resolution, homogeneous nucleation, random gas atom capture, biased gas atom capture (due to thermal gradients), random coalescence, and biased coalescence. In addition, effects due to biased diffusion out of the grain were included in a separate routine which calculated the biased release by the overlap of two spheres shifted by an amount consistent with the bubble velocity and the time period.

The bulk vacancy and interstitial concentration equations incorporated the production and loss mechanisms of diffusion, production, thermal emission from bubbles and dislocations, capture by bubbles and dislocations, and recombination.

The bubble radii equations were derived from considerations of the number of vacancies and interstitials entering and leaving a bubble. Governing processes from the bubble concentration equations were incorporated and modified by the associated vacancies involved during the interactions. In addition, direct interaction with vacancies and interstitials was incorporated through appropriate terms for thermal emission of vacancies and capture of vacancies and interstitials. Originally, every radius dependent term in these equations was updated during iterations. However, instabilities were encountered which were lessened when the terms coming directly from the concentration equation were held constant during iterations. Holding these terms constant did not eliminate all the causes of instability, and further restrictions had to be imposed on bubble radii calculations. The first of these was the enforcement of a minimum radius a bubble could have.

This was necessary because of the occasional calculation of negative pressures based on the radius. The minimum radius was determined based on the yield pressure of uranium nitride. The yield pressure was determined from a crude least squares fit to available stress-strain curves. A second restriction imposed was the enforcement of a maximum radius the bubble could have. The maximum radius was the equilibrium radius of the bubble. This avoided a stability problem in the thermal emission term, presumably caused by a loss of too much non-linearity in formulating the correction part of this term. These two additional restrictions stabilized the radii calculations, but added additional constraints on what REDSTONE could model. Steady-state or ramp-up scenarios would still be able to have their overpressurized non-equilibrium bubbles modeled. However, ramp-down scenarios would have their underpressurized bubbles modeled with equilibrium bubbles.

The validity of REDSTONE calculations were checked against analytical solutions for internal consistency and against experimental studies for agreement with the swelling and release results. Internal consistency checks validated that REDSTONE's basic structure was correctly programmed. These checks also indicated that the best agreement with transient results would be obtained by using an element scheme with smaller and smaller elements as the grain surface is approached, and by shifting the middle node in the outermost element towards the grain surface. The checks performed against experimental studies showed REDSTONE was satisfactorily accounting for the intragranular fission gas portion of swelling, but that other non-gaseous sources of swelling accounted for up to 1.5% additional swelling.

Once validated, REDSTONE was used to calculate the swelling and gas release associated with a hypothetical SDI scenario. Since an SDI nuclear system has not been specified, the reactor parameters used here were obtained from the SP-100 nuclear reactor. This reactor has a design failure point of 1% diametral increase of the fuel pin. Operating conditions for the scenario were calculated by a SP-100 systems code developed at Texas A&M University. The SDI scenario involved 10 years of low power operation at 10 kW followed by a 30 second rise to 1 MW. The 1 MW power was maintained for approximately 15 minutes. Fuel temperature was 500°K at low power and 1500°K at high power. The fission gas swelling and release results calculated by REDSTONE indicated that swelling would only be slightly above 1% and release would be negligible. Even when swelling due to non-gaseous components is added to this, the diametral increase in the fuel pellet remains below 1%, with almost 6% needed for design failure.

This same scenario was run at higher temperatures to see if uncertainties in the operating parameters could cause significantly different results. Both the low power and high power temperatures were increased by 100°K, 200°K, and 300°K. The increased temperatures had no significant effect during the low power period. After the high power period, the swelling and release results changed slightly, but the differences were unimportant. However, further investigation into the calculated solutions revealed that the results at the highest temperature of 1800°K appeared unreliable because high bubble mobilities were causing a shift to bubbles larger than those modeled by REDSTONE. Thus the swelling calculated by REDSTONE was smaller than if all possible bubble groups had been considered. Whether the actual swelling would have been important for failure considerations is not known since REDSTONE is not capable of modeling the bubble sizes required. REDSTONE's

results are important indications, however, since centerline fuel temperatures were expected to be near 1800°K.

It was apparent from REDSTONE's results that further investigations into the swelling behavior near 1800°K were needed. Additional uncertainties in property values used in REDSTONE and in the operating parameters of the fuel might even lower this transition temperature to 1700°K or lower.

Many of the property values used in REDSTONE had to be gleaned from a small library of studies. In some cases, only one value may have been available for use. In other cases, the values had to be taken from other fuels such as UC or UO_2. As a result, most of the properties have a great deal of uncertainty associated with them. In an effort to understand the effect these uncertainties might have on the solution, several of the properties were varied and the results compared. These included the xenon, surface, and irradiation enhanced diffusion coefficients, the surface energy, and the resolution parameter. Except for surface energy, the properties were varied by two orders of magnitude on either side of the programmed value. Most of the changes caused by varying the properties were small and unimportant with regards to failure. However, each property showed definite trends which might become important at higher temperatures. Thus, knowing these trends is an important design consideration.

The largest effects were observed from changes in irradiation enhanced diffusion and resolution. Although the changes caused by varying these properties were small, both had solutions which exhibited significant shifts to larger bubble sizes. The irradiation enhanced diffusion coefficient produced such results above 100 times the programmed value. The resolution parameter produced such results below 0.01 of the programmed value. However, the resolution parameter used in

REDSTONE is already lower than that experimentally observed for UO_2, so it is unlikely that the value used in REDSTONE is two orders of magnitude too high. On the contrary, the uncertainties in the irradiation enhanced diffusion coefficient would be more likely to range towards larger values in the same direction as the fall in reliability.

In situations where the bubble sizes remain small, the largest effect of uncertainties in property values could come from the resolution parameter. The value used in REDSTONE is one to two orders of magnitude lower than in UO_2 because it was assumed that UN resolution would occur via a single knock-on mechanism versus a thermal spike mechanism in UO_2. If, in fact, resolution is the same in both fuels, then the parameter study here indicates a reduction in the swelling and a corresponding increase in release.

Both the irradiation enhanced diffusion and resolution property studies suggested that the bubble profile after the power rise was determined by a combination of the pre-rise profile and the mobility of the bubbles during the high temperature period. Even though the time at high temperatures was short, it appeared that the conditions after the power rise were less a function of the pre-rise conditions than hypothesized.

9.2 Need for Further Work

Throughout this study, various areas, both external and internal to this study, were found which deserve further investigation. External to the study, it was clear that information on microscopic UN properties is severely lacking and much of it dates to the early 1970's and 1960's. Most of the information available is scattered in the literature and not very well corroborated. This includes data on diffusion of

gaseous species, the number of vacancies associated with gaseous species, the effect of stoichiometery, the effect of fission damage, surface energies, and nuclear processes such as resolution mechanisms, fission damage production rate, and nucleation mechanisms. These are clear candidates for additional work. In addition, microscopic studies of swelling in UN needs to be performed so that an intragranular code can be benchmarked directly against the data it produces.

Internal to this study, additional work was identified both by additional questions which arose and by deficiencies which were inevitably incorporated. The additional questions were 'what if' questions dealing with the scenario modeled. The study here looked at a single hypothetical scenario and a few of the parameters which could affect it. There are of course other possible scenarios, all which could be investigated. These other scenarios might involve different power histories as well as other reactor types. Some of the questions cannot be answered until firmer design specifications for an SDI system are developed. Additionally, sensitivity studies such as the ones performed on property uncertainties remain areas for additional work. Besides additional property data, calculational factors such as time step size and element spacing could be investigated. These latter studies are useful for fully characterizing the capabilities of the program.

The deficiencies needing additional work deal with limitations on the type of scenarios which could be modeled. The main limitation was that REDSTONE needed to model each individual bubble size. This prevents the calculation of results in high swelling situations. As a result, first on the list of additional work needed on REDSTONE is the incorporation of a bubble grouping scheme which can model the larger bubbles. The next most important area of improvement in REDSTONE is the calculation of bubble radii changes. As discussed previously, constraints built into

REDSTONE limit is usefulness in modeling underpressurized bubbles. These constraints need to be removed. In order to accomplish this the formulation of the bubble radii equations needs to be reworked to eliminate instabilities. Part of this would have to involve a restatement of the equation of state for xenon so that negative pressures could not be calculated. Possible approaches to rework the radii equations could be as simple as replacing the full equations with a relationship which does not incorporate the terms of the concentration change equation and does not require iteration, or as detailed as finding a way to incorporate all the types of equations into one calculation, and keeping the non-linearities. Many investigators have taken the former approach, and this is probably the easiest to accomplish.

9.3 Conclusions

Several conclusions can be drawn from this study. These are:

1) Non-linear finite element techniques can be used successfully to model the transient and non-equilibrium equations describing fission gas behavior in nuclear fuel. The most difficult part of the finite element formulations as performed here was in dealing with the non-equilibrium bubble behavior.
2) There is a lack of adequate experimental studies on microscopic swelling of UN to adequately benchmark an intragranular fission gas routine.
3) Within the benchmarking limitations, the program developed for this study appears to satisfactorily account for the fission gas component of swelling in the grain. Non-gaseous component swelling of fuel could add up to 1.5% additional swelling at low burnups.

4) Many of the property values needed to model fission gas behavior in UN either are not known and must be adopted from other materials or have large uncertainties because of the lack of corroborating studies. Uncertainties in the irradiation enhanced diffusion coefficient and the resolution parameter appear to have the greatest potential for causing significant swelling or release changes at higher burnups and/or higher temperatures.
5) The SDI scenario modeled here is not predicted to produce significant fuel swelling or release provided the temperature remains below 1700°K. Between 1700°K and 1800°K, bubble mobilities increase significantly so that accelerated swelling is possible. Uncertainties present in property values and operating parameters could lower this temperature even more.

REFERENCES

[1] A.C. Marshall and B. McKissock, Report SAND-89-1635C (Sandia National Laboratories, Albuquerque NM, 1989).

[2] R.F. Hilbert , V.W. Storhok, W. Chubb and D.L. Keller, in: Fast Reactor Fuel Element Technology, ed. R. Farmakes (American Nuclear Society, La Grange Park IL 1971) 753.

[3] M.D. DeHart, Numeric/Symbolic Computations Laboratory, Texas A&M University, Unpublished work (1989).

[4] R.B. Matthews, K.M. Chidester, C.W. Hoth, R.E. Mason and R.L. Petty, J.Nucl. Mater. 151 (1988) 334.

[5] D.R. Olander, Fundamental Aspects of Nuclear Reactor Fuel Elements, Available NTIS No. TID-26711-P1 (US Energy Research and Development Administration, Springfield VA, 1976).

[6] H.S. Carslaw and J.C. Jaeger, Conduction of Heat in Solids, 2nd ed., (Oxford University Press, Oxford UK, 1959).

[7] M.H. Wood, J. Nucl. Mater. 82 (1979) 264.

[8] H. Blank and Hj. Matzke, Rad. Effects. 17 (1973) 57.

[9] C. Ronchi and P.T. Elton, J. Nucl. Mater. 140 (1986) 228.

[10] R.S. Nelson, J. Nucl. Mater. 31 (1969) 153.

[11] B. Lustman, in: Uranium Dioxide: Properties and Nuclear Applications, ed.J. Belle (US GPO, Washington DC, 1961) 484.

[12] A.H. Booth, Report AECL-496 (Atomic Energy of Canada Ltd., Chalk River, Ontario,1957).

[13] J.R. MacEwan and W.H. Stevens, J. Nucl. Mater. 11 (1964) 77.

[14] E.E. Gruber, J. Appl. Phys. 38 (1967) 243.

[15] J.R. Matthews and M.H. Wood, J. Nucl. Mater. 84 (1979) 125.

[16] M.H. Wood and J.R. Matthews, Report AERE TP.825 (UK Atomic Energy Research Establishment, Harwell,1980).

[17] J. Rest and A.W. Cronenberg, J. Nucl. Mater. 150 (1987) 203.

[18] P. Bailey, S.E. Donnelly, D.G. Armour and Hj. Matzke, J. Nucl. Mater. 158 (1988) 19.

[19] R.L. Ritzman, A.J. Markworth, W. Oldfield and W. Chubb, Nucl. Appl. Tech. 9 (1970) 167.

[20] J.B. Melehan and J.E. Gates, Report BMI-1701 (Battelle Memorial Institute, Columbus OH,1964).

[21] T. Iseki, T. Hase, H. Suzuki, S. Kimura and T. Tamai, B. Tokyo Inst. Tech. 116 (1973) 91.

[22] P. Prajoto, A.R. Wazzan and D. Okrent, Nucl. Engrg. Des. 48 (1978) 461.

[23] C. Dollins and F.A. Nichols, J. Nucl. Mater. 66 (1977) 143.

[24] M.V. Speight, Nucl. Sci. Engrg. 37 (1969) 180.

[25] D.A. Collins and R. Hargreaves, in: Physical Metallurgy of Reactor Fuel Elements, eds. J.E. Harris and E.C. Sykes (The Metals Society, London, 1975) 253.

[26] M.R. Hayns and M.H. Wood, J. Nucl. Mater. 59 (1976) 293.

[27] J. Rest and S.M. Geil, Nucl. Engrg. Des. 56 (1980) 233.

[28] M.O. Tucker, J. Nucl. Mater. 79 (1979) 199.

[29] M.O. Tucker, J. Nucl. Mater. 78 (1978) 17.

[30] M.O. Tucker, J. Nucl. Mater. 79 (1979) 206.

[31] J.R. Matthews and M.H. Wood, Report AERE TP.830 (UK Atomic Energy Research Establishment, Harwell,1980).

[32] S.H. Chien, A.R. Wazzan and D. Okrent, Nucl. Tech. 60 (1983) 69.

[33] G.W. Greenwood and M.V. Speight, J. Nucl. Mater. 10 (1963) 140.

[34] S. Chandrasekhar, Rev. of Mod. Phys. 15 (1943) 1.

[35] M. Coquerelle and C.T. Walker, Nucl. Tech. 48 (1980) 43.

[36] C. Ronchi and C. Sari, J. Nucl. Mater. 58 (1975) 140.

[37] C. Ronchi, I.L.F. Ray, H. Thiele and J. Van de Laar, J. Nucl. Mater. 74 (1978) 193.

[38] S.L. Hayes, D.J. Senor, J.K. Thomas and K.L. Peddicord, Report ANFL-10-R (Advanced Nuclear Fuels Laboratory, Texas A&M University, 1989).

[39] S. Beck, Report BMI-1433 (Battelle Memorial Institute, Columbus OH,1960).

[40] P. Biddle, Report AERE-R4700 (UK Atomic Energy Research Establishment, Harwell,1964).

[41] A.H. Booth and G.T. Rymer, Report CRDC-720 (Atomic Energy of Canada Ltd., Chalk River, Ontario, 1957).

[42] N. Oi, Z. Naturforschg 21a (1966) 863.

[43] R.L. Ritzman, R.B. Price, R.B. Clark and D.L. Morrison, in: Progress on Development of Fuels and Technology for Advanced Reactors During July 1969 Through June 1970, ed. D.L. Keller, Report BMI-1886 (Battelle Memorial Institute, Columbus OH, 1970) B-66.

[44] D.G. Hurst, Report AECL-1550, CRRP-1124 (Atomic Energy of Canada Ltd., Chalk River, Ontario,1962).

[45] M.B. Weinstein, T.A. Kirchgessner and T.N. Tambling, Report NASA TND-7171 (Lewis Research Center, Cleveland OH, 1973).

[46] M.B. Weinstein and H.W. Davison, Report NASA TND-7401 (Lewis Research Center, Cleveland OH, 1973).

[47] H. Matsui, K. Sakanishi and T. Kirichara, J. Nucl. Sci. Tech., 12 (1975) 436.

[48] G.J. Dienes and A.C. Damask, J. Appl. Phys. 29 (1958) 1713.

[49] A. Höh and Hj. Matzke, J. Nucl. Mater. 48 (1973) 157.

[50] Hj. Matzke, Rad. Effects 53 (1980) 219.

[51] R.M. Cornell, J. Nucl. Mater. 38 (1971) 319.

[52] J.A. Brinkman and W.V. Johnston, presented at the AIME Symposium on Radiation Effects (Asheville NC, 1965) Referenced in [22].

[53] M.V. Speight, in:Physical Metallurgy of Reactor Fuel Elements, eds. J.E. Harris and E.C. Sykes (The Metals Society, London,1975) 222.

[54] C. Ronchi and C. Sari, J. Nucl. Mater. 50 (1974) 91.

[55] C.F. Clement, J. Nucl. Mater. 68 (1977) 63.

[56] S.C. Weaver, Report ORNL-TM-2016 (Oak Ridge National Laboratory TN, 1967).

[57] C. Ronchi, J. Nucl. Mater. 148 (1987) 316.

[58] F.A. Nichols, J. Nucl. Mater. 30 (1969) 143.

[59] C. Baker, J. Nucl. Mater. 71 (1977) 117.

[60] L.E. Willertz and P.G. Shewmon, Met. Trans. 1 (1970) 2217.

[61] M.E. Gulden, J. Nucl. Mater. 23 (1967) 30.

[62] W.B. Beere, J. Nucl. Mater. 45 (1972/73) 91

[63] E.E. Gruber, in: Physical Metallurgy of Reactor Fuel Elements, eds. J.E. Harris and E.C. Sykes (The Metals Society, London, 1975) 238.

[64] C. Ronchi, J. Nucl. Mater. 84 (1979) 55.

[65] W.M. Robertson, J. Nucl. Mater. 30 (1969) 36.

[66] M.R. Hayns, J. Nucl. Mater. 59 (1976) 175.

[67] J.L. Straalsund and G.L. Guthrie, Nucl. Tech.16 (1972) 36.

[68] R.G. Esteves, A.R. Wazzan and D. Okrent, Nucl. Engrg. Des. 45 (1978) 343.

[69] A.D. Brailsford and R.B. Bullough, in: Physical Metallurgy of Reactor Fuel Elements, eds. J.E. Harris and E.C. Sykes (The Metals Society, London, 1975) 148.

[70] G.W. Greenwood, A.J. Foreman and D.E. Rimmer, J. Nucl. Mater. 4 (1959) 305.

[71] C.C. Dollins and M. Jurisch, J. Nucl. Mater. 107 (1982) 46.

[72] R.O. Simmons, J.S. Koehler and R.W. Ballufi, in: Radiation Damage in Solids, Vol I (International Atomic Energy Agency, Vienna, 1962) 167.

[73] Y.D. Ting, A.R. Wazzan and D. Okrent, Nucl. Engrg. Des. 54 (1979) 29.

[74] D. Donner and W. Schiile, J. Nucl. Mater. 45 (1973) 293.

[75] D.T. Livey and P. Murray, J. Am. Ceram. Soc. 39 (1956) 363.

[76] G.C. Benson, P.I. Freeman and E. Dempsey, J. Am. Ceramic Soc. 46 (1963) 43.

[77] H.R. Warner and F.A. Nichols, Report WAPD-T-2161 (Bettis Atomic Power Lab, Pittsburgh PA, 1968).

[78] J.G. Eberhart, J. Nucl. Mater. 25 (1968) 103.

[79] P.S. Maiya, J. Nucl. Mater. 40 (1971) 57.

[80] E.M. Hodkin, et al., J. Nucl. Mater. 39 (1971) 59.

[81] R.W. Weeks, R.O. Scattergood and S.R. Pati, J. Nucl. Mater. 36 (1970) 223.

[82] Hj. Matzke and C. Ronchi, in: Advanced LMFBR Fuels, eds. J. Leary and H. Kittle (American Nuclear Society, La Grange Park IL, 1977) 219.

[83] R. Warren, in: V. Solid Compounds of Transition Elements (International Atomic Energy Agency, Vienna,1976) 82.

[84] P. Werner and H. Blank, Nucl. Tech. 52 (1981) 73.

[85] C.C. Dollins, J. Nucl. Mater. 60 (1976) 102.

[86] H. Wiedersich, in: Physical Metallurgy of Reactor Fuel Elements, eds. J.E. Harris and E.C. Sykes (The Metals Society, London, 1975) 142.

[87] H.R. Warner and F.A. Nichols, Nucl. Appl. Tech. 9 (1970) 148.

[88] B.L. Eyre and R. Bullough, J. Nucl. Mater. 26 (1968) 249.

[89] A.D. Whapman, Phil. Mag. 23 (1971) 987.

[90] C.C. Dollins and H. Ocken, Nucl. Appl. Tech. 9 (1970) 141.

[91] J.A. Turnbull, J. Nucl. Mater. 31 (1969) 153.

[92] J.A.L. Robertson, in: Physical Metallurgy of Reactor Fuel Elements, eds. J.E. Harris and E.C. Sykes (The Metals Society, London, 1975) 219.

[93] C.C. Dollins, J. Nucl. Mater. 60 (1976) 107.

[94] H.R. Warner and F.A. Nichols, Nucl. Appl. Tech. 9 (1970) 148.

[95] Hj. Matzke and J.A. Davies, J. Appl. Phys. 30 (1967) 805.

[96] C.R.A. Catlow, Rad. Effects 53 (1980) 127.

[97] Hayns, M.R. and M.H. Wood, J. Nucl. Mater. 67 (1977) 155.

[98] E.M. Baroody, J. Appl. Phys. 38 (1967) 243.

[99] C.E. Johnson, Report ANL-AFP-27 (Argonne National Laboratory IL,1976).

[100] J.A. Turnbull and R.M. Cornell, J. Nucl. Mater. 36 (1970) 161.

[101] J.A. Turnbull and R.M. Cornell, J. Nucl. Mater., 41 (1971) 156.

[102] A.M. Ross, J. Nucl. Mater. 30 (1969) 134.

[103] J.A. Turnbull, J. Nucl. Mater. 38 (1971) 203.

[104] C.C. Dollins and H. Ocken, J. Nucl. Mater. 45 (1972/73) 150.

[105] A.R. Whapman, Nucl. Appl., 2 (1966) 123.

[106] Hj. Matzke and C. Ronchi, in: Physical Metallurgy of Reactor Fuel Elements, eds. J.E. Harris and E.C. Sykes (The Metals Society, London, 1975) 259.

[107] R.S. Nelson, J. Nucl. Mater. 25 (1968) 227.

[108] A.J. Manley, Report TRG-Report-1681 (UK Atomic Energy Research Establishment, Sellafield, 1968).

[109] E.E. Gruber, Report ANL-8143 (Argonne National Laboratory IL, 1974).

[110] M.H. Wood, J. Nucl. Mater. 78 (1978) 58.

[111] R.C. Weast and S.M. Selby, eds., Handbook of Tables for Mathematics, 3rd Ed. (The Chemical Rubber Co., Cleveland OH, 1967).

[112] R.B. Poeppel, in: Conference on Fast Reactor Fuel Technology (American Nuclear Society, La Grange Park IL, 1977) 377.

[113] J.A. Turnbull, C.A. Friskney, J.R. Findlay, F.A. Johnson and A.J. Walter, Report CEGB-RD/B/N-4892 (Central Electricity Generating Board, Berkeley UK, 1980).

[114] J.B. Gibson, A.N. Goland, M. Milgram and G.H. Vineyard, Phys. Rev. 120 (1960) 1229.

[115] A.D. Brailsford and R. Bullough, J. Nucl. Mater. 44 (1972) 121.

[116] R. Bullough and R.C. Perrin, in: Voids Formed by Irradiation of Reactor Materials, eds. S.F. Pugh, M.H. Loretta and D.I.R. Norris (UK Atomic Energy Research Establishment, Harwell, 1971) 79.

[117] J. Rest and S.M. Gehl, Nucl. Engrg Des. 56 (1980) 233.

[118] S.H. Chien, A.R. Wazzan and D. Okrent, Nucl. Tech. 46 (1979) 110.

[119] S.D. Harkness and C.Y. Li, in: Radiation Damage in Reactor Materials, IAEA No. IAEA-SM-120/F-4 (International Atomic Energy Agency, Vienna,1969) 189.

[120] D.S. Burnett, Finite Element Analysis: From Concepts to Applications (Addison-Wesley Publishing Co., Reading MA, 1987).

[121] R.G. Rohal and T.N. Tambling, Report NASA Report TM X-2907 (Lewis Research Center, Cleveland OH, 1973).

[122] H. Watanabe, M. Kurihara, J. Shimokawa, M. Ichikawa, and M. Uchida, in: Advanced LMFBR Fuels, eds. J. Leary and H. Kittle (American Nuclear Society, La Grange Park IL, 1977) 361.

[123] E.K. Storms, J. Nucl. Mater. 158 (1988) 119.

[124] S.B. Ross, Uranium Nitride Fuel Swelling and Thermal Conductivity Correlations, M.S. Thesis (University of New Mexico, Albuquerque NM, 1987) Referenced in [123].

[125] R. Baars, NITRID: A Simple Code for Analysis of Uranium Nitride Fuel Pins, 4th Symposium on Space Nuclear Systems (Albuquerque, NM, 1987) Referenced in [123].

[126] J.K. Thomas, D.E. Brozak, and K.L. Peddicord, Material Property and Irradiation Performance Correlations for Nitride Fuels, 5th Symposium on Space Nuclear Power Systems (Albuquerque, NM, 1988) Referenced in [123].. See also: Final Project Report-5/30/87, Texas A&M University.

APPENDIX 1

PROGRAM REDSTONE LISTING

A1.1 Main Program

```
*=====================================================================+
*                                                                     |
*RRRRRRR  EEEEEEEE DDDDDDD   SSSSSS  TTTTTTTT  OOOOOO  NN    NN EEEEEEEE
*RRRRRRRR EEEEEEEE DDDDDDDD SSSSSSSS TTTTTTTT OOOOOOOO NNN   NN EEEEEEEE
*RR    RR EE       DD    DD SS          TT    OO    OO NNNN  NN EE      |
*RR    RR EEEEE    DD    DD SSSSSSS     TT    OO    OO NN NN NN EEEEE   |
*RRRRRRRR EEEEE    DD    DD  SSSSSSS    TT    OO    OO NN  NNNN EEEEE   |
*RRRRRR   EE       DD    DD       SS    TT    OO    OO NN   NNN EE      |
*RR   RR  EEEEEEEE DDDDDDDD SSSSSSSS    TT    OOOOOOOO NN    NN EEEEEEEE
*RR    RR EEEEEEEE DDDDDDD   SSSSSS     TT     OOOOOO  NN    NN EEEEEEEE
*                                                                     |
*  This program calculates displacements and stresses using           |
*   finite element algorithms.                                        |
*                                                                     |
*              Daniel L. DeForest                                     |
*              Nuclear Engineering Department                         |
*              Texas A&M University                                   |
*=====================================================================+
      PROGRAM REDSTONE

*-------------------------------------+
*  Get type declarations, common block,|
*   and initializations                |
*-------------------------------------+
      INCLUDE (MAIN)

      REAL*16 CNVRR(21),CNIRR(21)
      LOGICAL SUBINC
      CALL XUFLOW(0)

      OPEN(UNIT=10,STATUS='NEW',ACCESS='SEQUENTIAL',FORM='UNFORMATTED')

*-------------------------------------+
*  Define iteration parameters        |
*   NLINC - number of load increments |
*   LINC  - current load increment    |
*   CONRAT - initial convergence ratio|
*            set high initially       |
*   CONLIM - limit for determining    |
*            convergence              |
*-------------------------------------+
 1    CONLIM = 1.0Q-06
      CONRAT = 1.0Q+06
      VACRAT = 1.0Q+06
      RADRAT = 1.0Q+06

*-------------------------------------+
*  Read and output input parameters   |
*-------------------------------------+
      CALL INPUT(MAXITR,NLINC,SUBINC,IUNITS,LOUT,CONLIM)

*-------------------------------------+
*  Generate connectivity arrays       |
*-------------------------------------+
      CALL ICOND

*-------------------------------------+
*  Generate J and L arrays            |
*   Used for determining prescribed dof|
*-------------------------------------+
      CALL INTARS

*-------------------------------------+
*  Find max bandwidth                 |
*-------------------------------------+
      CALL SEMBND

      DENOM1 = 1.0Q0
      DNOM2V = 1.0Q0
      DNOM2I = 1.0Q0
      DENOM3 = 1.0Q0
      ATMREL = 0.Q0

      IF (IREST.EQ.1) CALL RESINP(DENOM1,DNOM2V,DNOM2I,DENOM3)

      DO 101 IPRD=1,NPRD

         DO 102 LINC=1,NLINC(IPRD)

            CALL PROP(IPRD,LINC,NLINC,0)

*-------------------------------------+
*  Start iterations                   |
*-------------------------------------+
            DO 50 ITERG=1,MAXITR
               ITRFLG = 0
```

```
            DO 10 ITERAT = 1,MAXITR

*-----------------------------------------+
*  Calculate elemental stiffness matricis |
*   and force vectors and assemble into   |
*   global matricis.                      |
*   The convergence ratio is calculated   |
*   here based on ||R-F||/||R||           |
*-----------------------------------------+
                CALL ASSEMB(IPRD,LINC,NLINC,ITERAT)

*-----------------------------------------+
*  Call routine to solve equations        |
*-----------------------------------------+
                CALL SOLVE (MAXH,INONPD)

              IF (IPRD.NE.1.OR.LINC.NE.1.OR.ITERAT.NE.1.OR.ITERG.NE.1)
     +           CONRAT = RNORM(RRBC,INONPD)/DENOM1
              IF (IREST.EQ.1) CONRAT = RNORM(RRBC,INONPD)/DENOM1

                CALL EXPAND(1)
*-----------------------------------------+
*  Test for convergence.                  |
*   If convergence then                   |
*    Increment load vector                |
*    Output results                       |
*    If LINC=NLINC then end               |
*-----------------------------------------+
                DENOM1 = RNORM(CONC,INONPD)+RNORM(DELCON,INONPD)
                IF (CONRAT.LT.CONLIM) GOTO 151
                ITRFLG = 1

*-----------------------------------------+
*  Go to next iteration                   |
*-----------------------------------------+
  10            CONTINUE

*-----------------------------------------+
*  If get to this point then too          |
*   many iterations                       |
*-----------------------------------------+
              CALL ERROR('Maximum iterations encountered.')

  151         CONTINUE
```

```
*-----------------------------------------+
*  Start iterations                       |
*-----------------------------------------+
            DO 20 ITERAT = 1,MAXITR

*-----------------------------------------+
*  Calculate elemental stiffness matricis |
*   and force vectors and assemble into   |
*   global matricis.                      |
*   The convergence ratio is calculated   |
*   here based on ||R-F||/||R||           |
*-----------------------------------------+

                CALL ASSMBV(IPRD,LINC,NLINC,ITERAT)

*-----------------------------------------+
*  Call routine to solve equations        |
*-----------------------------------------+
                CALL SOLVE (MAXHV,INONPV)

                CALL EXPAND(2)
*-----------------------------------------+
*  Test for convergence.                  |
*   If convergence then                   |
*    Increment load vector                |
*    Output results                       |
*    If LINC=NLINC then end               |
*-----------------------------------------+

          DO 103 I=1,NN-1
             CNVRR(I) = RRBC((I-1)*2+1)
             CNIRR(I) = RRBC((I-1)*2+2)
 103         CONTINUE
             IF (IPRD.NE.1.OR.LINC.NE.1.OR.ITERAT.NE.1.OR.ITERG.NE.1)
     +          VACRAT = RNORM(CNVRR,NN-1)/DNOM2V
     +                 + RNORM(CNIRR,NN-1)/DNOM2I
             IF (IREST.EQ.1)
     +          VACRAT = RNORM(CNVRR,NN-1)/DNOM2V
     +                 + RNORM(CNIRR,NN-1)/DNOM2I

*-----------------------------------------+
*  Test for convergence.                  |
*   If convergence then                   |
*    Increment load vector                |
*    Output results                       |
```

```
*    If LINC=NLINC then end                |
*------------------------------------------+
                DNOM2V = RNORM(CONCV,NN-1)+RNORM(DELCNV,NN-1)
                DNOM2I = RNORM(CONCI,NN-1)+RNORM(DELCNI,NN-1)
                IF (VACRAT.LT.CONLIM) GOTO 122

*------------------------------------------+
*  Go to next iteration                    |
*------------------------------------------+
  20            CONTINUE

*------------------------------------------+
*  If get to this point then too           |
*   many iterations                        |
*------------------------------------------+
              CALL ERROR('Maximum iterations encountered.')

 122          CONTINUE

*------------------------------------------+
*  Start iterations                        |
*------------------------------------------+
              DO 30 ITERAT = 1,MAXITR

                 CALL ASSMBR(IPRD,LINC,NLINC,ITERAT)

                 CALL SOLVE (MAXHR,INONPR)

                 CALL EXPAND(3)
*------------------------------------------+
*  Test for convergence.                   |
*   If convergence then                    |
*    Increment load vector                 |
*    Output results                        |
*    If LINC=NLINC then end                |
*------------------------------------------+
              IF (IPRD.NE.1.OR.LINC.NE.1.OR.ITERAT.NE.1.OR.ITERG.NE.1)
     +           RADRAT = RNORM(RR8C,INONPR)/DENOM3
              IF (IREST.EQ.1) RADRAT = RNORM(RR8C,INONPR)/DENOM3

                 DENOM3 = RNORM(RAD,INONPR)+RNORM(DELRAD,INONPR)
*------------------------------------------+
*  Test for convergence.                   |
*   If convergence then                    |
*    Increment load vector                 |
*    Output results                        |
*    If LINC=NLINC then end                |
*------------------------------------------+
                IF (RADRAT.LT.CONLIM) GOTO 123
                ITRFLG = 1

*------------------------------------------+
*  Go to next iteration                    |
*------------------------------------------+
  30            CONTINUE

*------------------------------------------+
*  If get to this point then too           |
*   many iterations                        |
*------------------------------------------+
              CALL ERROR('Maximum iterations encountered.')

 123          CONTINUE

              CALL RADII
              IF (ITRFLG.EQ.0) GOTO 200
  50          CONTINUE

 200  CONTINUE

        CALL SETCON(IPRD,LINC,NLINC,1)
        CALL SETCON(IPRD,LINC,NLINC,2)
        CALL SETCON(IPRD,LINC,NLINC,3)

        CALL SWELL(IPRD,LINC,NLINC,PCTREL,PCTSWL,ATMREL)

        IF (((LINC/LOUT(IPRD))*LOUT(IPRD)).EQ.LINC)
     +     CALL OUTPUT(LINC,IPRD,NLINC,.TRUE.,ITERAT,IUNITS,
     +                 PCTREL,PCTSWL)

 102    CONTINUE
        CALL RESSAV(DENOM1,DNOM2V,DNOM2I,DENOM3)
 101    CONTINUE

 9999 END
*==========================================================================+
*                                                                          |
*      AAAAAA    SSSSSS    SSSSSS   EEEEEEEE  MMMMMM   BBBBBBB           |
*     AAAAAAAA  SSSSSSSS  SSSSSSSS  EEEEEEEE  MMMMMMM  BBBBBBBB          |
*     AA    AA  SS        SS        EE        MM MM MM  BB     BB         |
```

```
*      AA    AA  SSSSSSS   SSSSSSS   EEEEE     MM MM MM  BBBBBB
*      AAAAAAAA   SSSSSS   SSSSSS  EEEEE     MM MM MM  BBBBBBB
*      AAAAAAAA        SS        SS EE        MM MM MM  BB    BB
*      AA    AA  SSSSSSSS  SSSSSSSS  EEEEEEEE  MM MM MM  BBBBBBBB
*      AA    AA   SSSSSS    SSSSSS   EEEEEEEE  MM MM MM  BBBBBBB
*
*     This subroutine assembles the global, bandstorage,
*     symmetric, condensed system matrix.
*
*    Inputs are:
*       NE - total number of elements
*       NDOF - total number of degrees of freedom
*       NPD - total number of prescribed degrees of freedom
*       NDOFE(i) - number of degrees of freedom for element i
*       ICONDF(i,j) - connectivity array of local dofs to global dofs
*                     i.e. global node number of local node J
*                     for element i
*       IDOFPD(i) - global degree of freedom number of ith
*                   prescribed displacement
*       UP(i) -  prescribed displacement of global node i
*
*=======================================================================

      SUBROUTINE ASSEMB(IPRD,LINC,NLINC,ITERAT)

*-----------------------------------------------------------------------
*  Get declarations
*-----------------------------------------------------------------------
      INCLUDE (MAIN)

*-----------------------------------------------------------------------
*  Zero K and R matrices
*
*   RKBC(i,j) - global value of ith row and jth column of stiffness
*   RRBC(i) - global value of ith row of force vector
*-----------------------------------------------------------------------

      DO 108 I=1,INONPD
         DO 109 J=1,2*MAXH
C     DO 108 I=1,IDMRB1
C        DO 109 J=1,IDMRB2
            RKBC(I,J) = 0.0Q0
 109        CONTINUE
         RRBC(I) = 0.0Q0
 108     CONTINUE

*-----------------------------------------------------------------------
*  Get element stiffness matrices and force vectors and create the
*   global matrix and vector.
*   The result is stored in banded storage with prescribed
*   displacement rows eliminated and accounted for.
*
*   EL1KF - subroutine that provides elemental values
*   IBAND - row or global matrix
*   JBAND - column of global matrix (banded)
*   ELK(i,j) - element K value for ith element row and jth element colu
*   ELF(i) - element F value for ith row
*   UP(i) - prescribed displacement of ith global node
*-----------------------------------------------------------------------

      DO 110 I=1,NE
         NEDOF = NNELEM*NBBLSZ

 3001    CALL ELKFG(I,IPRD,LINC,NLINC,ITERAT)

 3999    DO 111 M=1,NEDOF
            IGME = ICONDF(I,M)
            LIGME = LARRAY(IGME)
            IBAND = LIGME
            JMAX = LIGME+MAXH-1
            JMIN = LIGME-MAXH+1
            DO 112 N=1,NEDOF
               IGNE = ICONDF(I,N)
               LIGNE = LARRAY(IGNE)

*-----------------------------------------------------------------------
*  Do not add in stiffness matrix if either value matches that
*   of a prescribed dof...these values are subtracted from the
*   force vector later
*-----------------------------------------------------------------------

               IF (LIGNE.EQ.0.OR.LIGME.EQ.0) GOTO 40
               JBAND = LIGNE-LIGME+MAXH

*-----------------------------------------------------------------------
*  Don't worry about the lower half of the matrix..its symmetric
*-----------------------------------------------------------------------

               IF (LIGNE.LT.JMIN.OR.LIGNE.GT.JMAX) GOTO 40
               RKBC(IBAND,JBAND) = RKBC(IBAND,JBAND) + ELK(M,N)
```

```
 40           CONTINUE

*-------------------------------------------------------------+
*  Subtract prescribed values from force vector if row is not a  |
*   prescribed dof but column is                                 |
*-------------------------------------------------------------+

 112          CONTINUE

*-------------------------------------------------------------+
*  Add in element force value if not for a prescribed dof     |
*-------------------------------------------------------------+

          IF (LIGME.EQ.0) GOTO 111
          RRBC(LIGME) = RRBC(LIGME)+ELF(M)
 111      CONTINUE
 110    CONTINUE

      RETURN
      END
*=============================================================+
*                                                             |
*       AAAAAA    SSSSSS    SSSSSS   MMMMMM   BBBBBBB   RRRRRRR      |
*      AAAAAAAA  SSSSSSSS  SSSSSSSS  MMMMMMM  BBBBBBBB  RRRRRRRR     |
*      AA    AA  SS        SS        MM MM MM  BB    BB  RR    RR     |
*      AA    AA  SSSSSSS   SSSSSSS   MM MM MM  BBBBBBB   RR    RR     |
*      AAAAAAAA   SSSSSSS   SSSSSSS  MM MM MM  BBBBBBB   RRRRRRRR     |
*      AAAAAAAA        SS        SS  MM MM MM  BB    BB  RRRRRR       |
*      AA    AA  SSSSSSSS  SSSSSSSS  MM MM MM  BBBBBBBB  RR   RR      |
*      AA    AA   SSSSSS    SSSSSS   MM MM MM  BBBBBBB   RR    RR     |
*                                                             |
*     This subroutine assembles the global, bandstorage,      |
*     symmetric, condensed system matrix.                     |
*                                                             |
*    Inputs are:                                              |
*       NE - total number of elements                         |
*       NDOF - total number of degrees of freedom             |
*       NPD - total number of prescribed degrees of freedom   |
*       NDOFE(i) - number of degrees of freedom for element i |
*       ICONDF(i,j) - connectivity array of local dofs to global dofs |
*                     i.e. global node number of local node J |
*                     for element i                           |
*       IDOFPD(i) - global degree of freedom number of ith    |
*                   prescribed displacement                   |
*       UP(i) -  prescribed displacement of global node i     |
*                                                             |
*=============================================================+

      SUBROUTINE ASSMBR(IPRD,LINC,NLINC,ITERAT)

*-------------------------------------------------------------+
*  Get declarations                                           |
*-------------------------------------------------------------+
      INCLUDE (MAIN)

*-------------------------------------------------------------+
*  Zero K and R matrices                                      |
*                                                             |
*   RKBC(i,j) - global value of ith row and jth column of stiffness |
*   RRBC(i) - global value of ith row of force vector         |
*-------------------------------------------------------------+

       INONPR = (NN-1)*(NBBLSZ-1)
       MAXHR = NNELEM*(NBBLSZ-1)

      DO 108 I=1,INONPR
         DO 109 J=1,2*MAXHR
            RKBC(I,J) = 0.0Q0
 109        CONTINUE
         RRBC(I) = 0.0Q0
 108     CONTINUE

*-------------------------------------------------------------+
*  Get element stiffness matrices and force vectors and create the |
*   global matrix and vector.                                 |
*   The result is stored in banded storage with prescribed    |
*   displacement rows eliminated and accounted for.           |
*                                                             |
*   EL1KF - subroutine that provides elemental values         |
*   IBAND - row or global matrix                              |
*   JBAND - column of global matrix (banded)                  |
*   ELK(i,j) - element K value for ith element row and jth element colu |
*   ELF(i) - element F value for ith row                      |
*   UP(i) - prescribed displacement of ith global node        |
*-------------------------------------------------------------+

      DO 110 I=1,NE
         NEDOF = NNELEM*NBBLSZ
```

```
3001    CALL ELKFR(I,IPRD,LINC,NLINC,ITERAT)

3999    DO 111 M=1,NEDOF
           IGME = ICONDF(I,M)
           LIGME = LARRAY(IGME)
           IF (M-(M/NBBLSZ*NBBLSZ).NE.1.AND.LIGME.NE.0) THEN
              LIGME = (I-1)*(NNELEM-1)*(NBBLSZ-1)-(M-1)/NBBLSZ+M-1
              ELSE
              LIGME = 0
              ENDIF
           IBAND = LIGME
           JMAX = LIGME+MAXHR-1
           JMIN = LIGME-MAXHR+1
           DO 112 N=1,NEDOF
              IGNE = ICONDF(I,N)
              LIGNE = LARRAY(IGNE)
              IF (N-(N/NBBLSZ*NBBLSZ).NE.1.AND.LIGNE.NE.0) THEN
                 LIGNE = (I-1)*(NNELEM-1)*(NBBLSZ-1)-(N-1)/NBBLSZ+N-1
                 ELSE
                 LIGNE = 0
                 ENDIF

*-------------------------------------------------------------+
*  Do not add in stiffness matrix if either value matches that |
*   of a prescribed dof...these values are subtracted from the |
*   force vector later                                         |
*-------------------------------------------------------------+

              IF (LIGNE.EQ.0.OR.LIGME.EQ.0) GOTO 40
              JBAND = LIGNE-LIGME+MAXHR

*-------------------------------------------------------------+
*  Don't worry about the matrix outside band                  |
*-------------------------------------------------------------+

              IF (LIGNE.LT.JMIN.OR.LIGNE.GT.JMAX) GOTO 40

              RKBC(IBAND,JBAND) = RKBC(IBAND,JBAND) + ELK(M,N)

40            CONTINUE

*-------------------------------------------------------------+
*  Subtract prescribed values from force vector if row is not a |
*   prescribed dof but column is                               |
*-------------------------------------------------------------+

112           CONTINUE

*-------------------------------------------------------------+
*  Add in element force value if not for a prescribed dof     |
*-------------------------------------------------------------+

           IF (LIGME.EQ.0) GOTO 111
           RRBC(LIGME) = RRBC(LIGME)+ELF(M)
111        CONTINUE
110     CONTINUE

        RETURN
        END
*=============================================================+
*                                                             |
*      AAAAAA    SSSSSS    SSSSSS   MMMMMM   BBBBBB   VV    VV      |
*     AAAAAAAA  SSSSSSSS  SSSSSSSS  MMMMMMM  BBBBBBB  VV    VV      |
*     AA    AA  SS        SS        MM MM MM  BB    BB  VV    VV      |
*     AA    AA  SSSSSSS   SSSSSSS   MM MM MM  BBBBBBB  VV    VV      |
*     AAAAAAAA   SSSSSSS   SSSSSSS  MM MM MM  BBBBBB   VV    VV      |
*     AAAAAAAA        SS        SS  MM MM MM  BB    BB  VV  VV       |
*     AA    AA  SSSSSSSS  SSSSSSSS  MM MM MM  BBBBBBB    VVVV       |
*     AA    AA   SSSSSS    SSSSSS   MM MM MM  BBBBBB      VV        |
*                                                             |
*    This subroutine assembles the global, bandstorage,       |
*    non-symmetric, condensed system matrix for calculating the |
*    vacancy and interstitial concentration in the grain      |
*                                                             |
*   Inputs are:                                               |
*=============================================================+

        SUBROUTINE ASSMBV(IPRD,LINC,NLINC,ITERAT)

*-------------------------------------------------------------+
*  Get declarations                                           |
*-------------------------------------------------------------+
        INCLUDE (MAIN)

*-------------------------------------------------------------+
*  Zero K and R matrices                                      |
*                                                             |
*   RKBC(i,j) - global value of ith row and jth column of stiffness |
*   RRBC(i) - global value of ith row of force vector         |
*-------------------------------------------------------------+
```

```
      MAXHV = NNELEM*2
      INONPV = (NN-1)*2
      DO 108 I=1,INONPV
         DO 109 J=1,2*MAXHV
            RKBC(I,J) = 0.0Q0
 109        CONTINUE
         RRBC(I) = 0.0Q0
 108     CONTINUE

*-----------------------------------------------------------------------+
*  Get element stiffness matrices and force vectors and create the      |
*   global matrix and vector.                                           |
*   The result is stored in banded storage with prescribed              |
*   displacement rows eliminated and accounted for.                     |
*                                                                       |
*   EL1KF - subroutine that provides elemental values                   |
*   IBAND - row or global matrix                                        |
*   JBAND - column of global matrix (banded)                            |
*   ELK(i,j) - element K value for ith element row and jth element colu |
*   ELF(i) - element F value for ith row                                |
*   UP(i) - prescribed displacement of ith global node                  |
*-----------------------------------------------------------------------+

      DO 110 I=1,NE
         NEDOF = NNELEM*2

         CALL ELKFV(I,IPRD,LINC,NLINC,ITERAT)

 3999    DO 111 M=1,NEDOF
            LIGME = (I-1)*(NEDOF-2)+M
            IF (I.EQ.NE.AND.M.EQ.NEDOF) LIGME = 0
            IF (I.EQ.NE.AND.M.EQ.(NEDOF-1)) LIGME = 0
            IBAND = LIGME
            JMAX = LIGME+MAXHV-1
            JMIN = LIGME-MAXHV+1
            DO 112 N=1,NEDOF
               LIGNE = (I-1)*(NEDOF-2)+N
               IF (I.EQ.NE.AND.N.EQ.NEDOF) LIGNE = 0
               IF (I.EQ.NE.AND.N.EQ.(NEDOF-1)) LIGNE = 0

*-----------------------------------------------------------------------+
*  Do not add in stiffness matrix if either value matches that          |
*   of a prescribed dof...these values are subtracted from the          |
*   force vector later                                                  |
*-----------------------------------------------------------------------+

               IF (LIGNE.EQ.0.OR.LIGME.EQ.0) GOTO 40
               JBAND = LIGNE-LIGME+MAXHV

*-----------------------------------------------------------------------+
*  Don't worry about the lower half of the matrix..its symmetric        |
*-----------------------------------------------------------------------+

               IF (LIGNE.LT.JMIN.OR.LIGNE.GT.JMAX) GOTO 40
               RKBC(IBAND,JBAND) = RKBC(IBAND,JBAND) + ELK(M,N)

 40            CONTINUE

*-----------------------------------------------------------------------+
*  Subtract prescribed values from force vector if row is not a         |
*   prescribed dof but column is                                        |
*-----------------------------------------------------------------------+

 112           CONTINUE

*-----------------------------------------------------------------------+
*  Add in element force value if not for a prescribed dof               |
*-----------------------------------------------------------------------+

            IF (LIGME.EQ.0) GOTO 111
            RRBC(LIGME) = RRBC(LIGME)+ELF(M)
 111        CONTINUE
 110     CONTINUE

      RETURN
      END
*=======================================================================+
*                                                                       |
*      DDDDDDD   IIIIIIII  FFFFFFFF  FFFFFFFF  UU    UU   SSSSSS       |
*      DDDDDDDD  IIIIIIII  FFFFFFFF  FFFFFFFF  UU    UU  SSSSSSSS      |
*      DD    DD     II     FF        FF        UU    UU  SS            |
*      DD    DD     II     FFFFF     FFFFF     UU    UU  SSSSSSS       |
*      DD    DD     II     FFFFF     FFFFF     UU    UU   SSSSSSS      |
*      DD    DD     II     FF        FF        UU    UU        SS      |
*      DDDDDDDD  IIIIIIII  FF        FF        UUUUUUUU  SSSSSSSS      |
*      DDDDDDD   IIIIIIII  FF        FF         UUUUUU   SSSSSS        |
*                                                                       |
*    This subroutine calculates diffusion coefficients                  |
*                                                                       |
```

```
      SUBROUTINE DIFFUS

      INCLUDE (MAIN)
      INCLUDE (DIFFCOEF)
      INCLUDE (PI)
      INCLUDE (LATTICE)

     DIFFX = DIFFXA*QEXP(-DIFFXB/TEMP(1)) + DIFFIE*FISSRC(1)
     DIFFS = DIFFSA*QEXP(-DIFFSB/TEMP(1))
     DIFFU = DIFFUA*PRSNGN(1)**DIFFUB*QEXP(-DIFFUC/TEMP(1))
     DIFFN = DIFFNA*PRSNGN(1)**DIFFNB*QEXP(-DIFFNC/TEMP(1))
    +        *GRNDIA**(DIFND1+DIFND2*TEMP(1))
     DIFFV = DIFFVA*QEXP(-DIFFVB/TEMP(1))*1.Q0
     DIFFI = DIFFIA*QEXP(-DIFFIB/TEMP(1))

     PREFIX = 3.Q0*LAMBDA*OMEGA/(2.Q0*PI)
     PREFIX = PREFIX*4.Q0/(DIFFLM**2)

     DO 101 I=1,NN

        DIFFCF(I,1) = DIFFX

        IF (NBBLSZ.GE.2) THEN
           DO 102 J=2,NBBLSZ

              DIFFCF(I,J) = DIFFS/(RADIUS(I,J)**2) * PREFIX
              IF (RADIUS(I,J).LT.(DIFFLM/PI)) DIFFCF(I,J) =
    +            DIFFCF(I,J)*(QSIN(DIFFLM/(2.Q0*RADIUS(I,J))))**2

  102         CONTINUE
           ENDIF

  101    CONTINUE
      RETURN
      END
```

```
*                                                                      |
*          EEEEEEEE  LL        KK    KK  FFFFFFFF   GGGGGG           |
*          EEEEEEEE  LL        KK   KK   FFFFFFFF  GGGGGGGG          |
*          EE        LL        KK  KK    FF        GG                |
*          EEEEE     LL        KKKKK     FFFFF     GG  GGGG          |
*          EEEEE     LL        KKKKK     FFFFF     GG  GGGG          |
*          EE        LL        KK  KK    FF        GG    GG          |
*          EEEEEEEE  LLLLLLLL  KK   KK   FF        GGGGGGGG          |
*          EEEEEEEE  LLLLLLLL  KK    KK  FF         GGGGGG           |
*                                                                     |
*                                                                     |
*  This routine calculates the elemental contributions to fission     |
*   gas bubbles using spherical geometry.                             |
*=====================================================================+
      SUBROUTINE ELKFG(IELEM,IPRD,LINC,NLINC,ITERAT)

      INCLUDE (MAIN)
      INCLUDE (ELKF)
      INCLUDE (GAUSS)
      INCLUDE (PI)
      INCLUDE (BOLTZMAN)
      INCLUDE (SURFENGY)

*-------------------------------------------+
*  Zero element force and stiffness matricis |
*-------------------------------------------+

      DO 450 I=1,IDMEK1
         ELF(I) = 0.0Q0
         DO 451 J=1,IDMEK2
            ELK(I,J) = 0.0Q0
 451        CONTINUE
 450     CONTINUE

*--------------------------+
*  Get global node numbers  |
*--------------------------+

      NODE1 = ICON(IELEM,1)
      NODE2 = ICON(IELEM,2)
      NODE3 = ICON(IELEM,3)

*-------------------------------+
*  Loop over integration points  |
*-------------------------------+
      DO 701 ITR=1,IGP

       CALL SHAPE(NNELEM,ITR,IGP,SHPFCT,DERSHP)

*-------------------------------------------------+
*  Value of R in real coordinates of integration point  |
*-------------------------------------------------+
       RGP = SHPFCT(1)*R(NODE1)+
```

```
     +       SHPFCT(2)*R(NODE2)+
     +       SHPFCT(3)*R(NODE3)

*-------------------------------------------+
*  Calculate Jacobian matrix and determinant  |
*-------------------------------------------+

      AJMAT = DERSHP(1) * R(NODE1)
     +      + DERSHP(2) * R(NODE2)
     +      + DERSHP(3) * R(NODE3)

      AJDET = AJMAT

*-----------------------------------------------------------+
*  Calculate numerical integration multipliers times R**2  |
*-----------------------------------------------------------+

      WMULT = WEIGHT(ITR,IGP)*RGP**2*AJDET

*--------------------+
*  Calculate source  |
*--------------------+

      SOURCE = ATMSRC(IMAT(IELEM))

*-------------------------------------+
*  Loop over matrix rows for one node  |
*-------------------------------------+

        TOTSRC = 0.Q0

        DO 106 N=1,NBBLSZ

*-------------------------------------------------------------------+
*  Calculate integration point bubble radii, diffusion coefficient,  |
*   concentrations, delta concentrations, derivative concentrations, |
*   velocities, resolution parameters                                |
*-------------------------------------------------------------------+

           RADIP(N) = SHPFCT(1)*RADIUS(NODE1,N) +
     +                SHPFCT(2)*RADIUS(NODE2,N) +
     +                SHPFCT(3)*RADIUS(NODE3,N)

           DFCFIP(N) = SHPFCT(1)*DIFFCF(NODE1,N) +
     +                 SHPFCT(2)*DIFFCF(NODE2,N) +
     +                 SHPFCT(3)*DIFFCF(NODE3,N)

           IDOF1 = ICONDF(IELEM,N)
           IDOF2 = ICONDF(IELEM,(NBBLSZ+N))
           IDOF3 = ICONDF(IELEM,(2*NBBLSZ+N))
           IDOF4 = ICONDF(IELEM,(3*NBBLSZ+N))
           CONCIP(N) = SHPFCT(1)*(CONC(IDOF1) + DELCON(IDOF1)) +
     +                 SHPFCT(2)*(CONC(IDOF2) + DELCON(IDOF2)) +
     +                 SHPFCT(3)*(CONC(IDOF3) + DELCON(IDOF3))
           DELCIP(N) = SHPFCT(1)*DELCON(IDOF1) +
     +                 SHPFCT(2)*DELCON(IDOF2) +
     +                 SHPFCT(3)*DELCON(IDOF3)
           DERCIP(N) = DERSHP(1)*(CONC(IDOF1) + DELCON(IDOF1)) +
     +                 DERSHP(2)*(CONC(IDOF2) + DELCON(IDOF2)) +
     +                 DERSHP(3)*(CONC(IDOF3) + DELCON(IDOF3))

           VELIP(N) = 3.Q0 / (BLTZMN*(TEMP(1)**2)*(LTPARM**3))
     +              *DIFFS*TMPGR(1)*PI*(RADIP(N)**3)*SRFACT
           IF (RADIP(N).GT.15.Q-8) THEN
              RESOL(N) = RESOLP*QFLOAT(N)*
     +                    (1.Q0-(1.Q0-15.Q-8/RADIP(N)-RESADJ)**3)
              ELSE
              RESOL(N) = RESOLP*QFLOAT(N)
              ENDIF

              SRCFRC(N) = 8.40128Q0*PI*OMEGA*RADIP(N)**2/LTPARM**2
              TOTSRC = TOTSRC + SRCFRC(N)*CONCIP(N)

  106        CONTINUE

        IF (TOTSRC.GT.1.Q0) THEN
                 WRITE(8,*)  '1 ******************** WARNING!!!!!! ',
     +                       '********************'
                 WRITE(8,*)  '  Bubble concentrations have exceeded',
     +                       ' the capacity of the lattice.'
                 ENDIF

           TOTSRC = QMIN1(TOTSRC,1.Q0)

*-------------------------------------------------------------------+
*  Start assembly of equations                                      |
*-------------------------------------------------------------------+

           DO 101 N=1,NBBLSZ
*------------------+
```

```
*  Zero constants  |
*------------------+

           ALPHA = 0.Q0
           BETA  = 0.Q0
           GAMMA = 0.Q0

*-------------------------------------------------+
*  Set up re-solution loss and atomic source terms  |
*-------------------------------------------------+

           IF (N.NE.1) THEN
              BETA = 0.Q0
              BETA = -RESOL(N)
              ELSE
              ALPHA = SOURCE
              ENDIF

*-----------------------------------------------------------------+
*  Self-coalescence loss (twice regular coalescence loss)         |
*-----------------------------------------------------------------+

           IF ((2*N).LE.NBBLSZ) GAMMA = -32.Q0*PI*RADIP(N)*DFCFIP(N)

*-----------------------------------------------------------------+
*  Add in terms to matrix along with time terms                   |
*-----------------------------------------------------------------+
           TELK(N,N) = -1.Q0/DELTIM(IPRD) + BETA + 2.Q0*GAMMA*CONCIP(N)
           TELF(N) = DELCIP(N)/DELTIM(IPRD) - ALPHA - BETA*CONCIP(N)
     +               - GAMMA*CONCIP(N)*CONCIP(N)

*--------------------------+
*  Max bubble size for ROE  |
*--------------------------+

           NCHECK = NBBLSZ - N

*-------------------------------------------------+
*  Loop over contributions from other bubble groups  |
*-------------------------------------------------+

           DO 102 K=1,NBBLSZ

*-------------------------------------------------+
*  Already taken care of process involving only N   |
*-------------------------------------------------+

           IF (K.EQ.N) GOTO 102

*------------------+
*  Zero constants  |
*------------------+

           ROE = 0.Q0
           ETA = 0.Q0
           EPSILN = 0.Q0

*-----------------------------------------+
*  Coalescence loss, both random and biased  |
*-----------------------------------------+

           IF (K.LE.NCHECK) ROE = -4.Q0*PI*(RADIP(N)+RADIP(K))
     +                                  *(DFCFIP(N)+DFCFIP(K))
     +                              -PI*(RADIP(N)+RADIP(K))**2
     +                                 *QABS(VELIP(N)-VELIP(K))

*-----------------------------------------------------+
*  Resolution gain and source gain (loss for N=1)      |
*-----------------------------------------------------+

           IF (N.EQ.1) THEN
              IF (K.EQ.2) THEN
                 ETA = 2.Q0*RESOL(K)
                 ETA = ETA - SOURCE*SRCFRC(K)
                 ELSE
                 ETA = RESOL(K)
                 ETA = ETA - SOURCE*SRCFRC(K)
                 ENDIF
              ELSE
              IF (K.EQ.(N+1)) THEN
              ETA = RESOL(K)
              ENDIF
              IF (K.EQ.(N-1))
     +                  ETA = SOURCE*SRCFRC(K)
              ENDIF

*---------------------------------------+
*  Add in contributions from ETA and ROE  |
*---------------------------------------+
```

```
            TELK(N,N) = TELK(N,N) + ROE*CONCIP(K)
            TELK(N,K) = ETA + ROE*CONCIP(N)
            TELF(N) = TELF(N) - ETA*CONCIP(K)
     +                - ROE*CONCIP(N)*CONCIP(K)

*-------------------------------------------------------------------+
*  Gain from coalescence of other than N bubbles, random and biased |
*-------------------------------------------------------------------+

            J = N-K
            IF (J.GT.0.AND.K.GE.J) THEN
               EPSILN = 4.Q0*PI*(RADIP(J) + RADIP(K))
     +                        *(DFCFIP(J) + DFCFIP(K))
     +                  +PI*(RADIP(J)+RADIP(K))**2
     +                     *QABS(VELIP(J)-VELIP(K))

*-------------------------------------------------------------------+
*  Add in contributions                                             |
*-------------------------------------------------------------------+

               TELK(N,K) = TELK(N,K) + EPSILN*CONCIP(J)
               TELK(N,J) = TELK(N,J) + EPSILN*CONCIP(K)
               TELF(N) = TELF(N) - EPSILN*CONCIP(J)*CONCIP(K)
               ENDIF
  102       CONTINUE

*-------------------------------------------------------------------+
*  Fill out rest of matrix (other nodes) and include diffusive terms |
*-------------------------------------------------------------------+

         DO 103 ND1=NNELEM,1,-1
            IND1 = (ND1-1)*NBBLSZ + N
            DO 104 L=1,NBBLSZ
               DO 105 ND2=NNELEM,1,-1
                  IND2 = (ND2-1)*NBBLSZ + L
                     IF (N.NE.L) THEN
                        TELK(IND1,IND2) = TELK(N,L)*WMULT*SHPFCT(ND1)
     +                                                 *SHPFCT(ND2)
                        ELSE
                        TELK(IND1,IND2) = (-DFCFIP(N)/AJDET**2
     +                                    *DERSHP(ND1)*DERSHP(ND2)
     +                                    +TELK(N,L)*SHPFCT(ND1)
     +                                    *SHPFCT(ND2))*WMULT
                        ENDIF
  105             CONTINUE
  104          CONTINUE
                  TELF(IND1) = (DFCFIP(N)*DERSHP(ND1)/AJDET**2
     +                        *DERCIP(N)+SHPFCT(ND1)*TELF(N))*WMULT
  103       CONTINUE
  101    CONTINUE

*-------------------------------------------------------------------+
*  Add results for this integration point into overall element matrix |
*-------------------------------------------------------------------+

      CALL MTXADD(TELK,ELK,NEDOF,NEDOF,IELEM,
     +            IDMCM1,IDMCM2,IDMCM3,IDMEK1,IDMEK2,IDMEK3,
     +            .FALSE.,.FALSE.)
*-----------------------------------------------+
*  Add load vector into overall force vector    |
*-----------------------------------------------+

      CALL MTXADD(TELF,ELF,NEDOF,1,IELEM,
     +           IDMCV1,IDMCV2,IDMCV3,IDMEF1,IDMEF2,IDMEF3,
     +           .FALSE.,.FALSE.)

*-----------------------------------------------+
*  Next integration point                       |
*-----------------------------------------------+

  701 CONTINUE
      RETURN
      END
*=================================================================+
*                                                                 |
*          EEEEEEEE  LL        KK    KK  FFFFFFFF  RRRRRRR        |
*          EEEEEEEE  LL        KK   KK   FFFFFFFF  RRRRRRRR       |
*          EE        LL        KK  KK    FF        RR     RR      |
*          EEEEE     LL        KKKKK     FFFFF     RR     RR      |
*          EEEEE     LL        KKKKK     FFFFF     RRRRRRRR       |
*          EE        LL        KK  KK    FF        RRRRRR         |
*          EEEEEEEE  LLLLLLLL  KK   KK   FF        RR    RR       |
*          EEEEEEEE  LLLLLLLL  KK    KK  FF        RR     RR      |
*                                                                 |
*   This routine calculates the elemental contributions to fission |
*    gas bubbles using spherical geometry.                        |
*=================================================================+
      SUBROUTINE ELKFR(IELEM,IPRD,LINC,NLINC,ITERAT)

      INCLUDE (MAIN)
```

```
      INCLUDE (ELKF)
      INCLUDE (GAUSS)
      INCLUDE (PI)
      INCLUDE (BOLTZMAN)
      INCLUDE (SURFENGY)
      INCLUDE (VANDWAAL)
      INCLUDE (LATTICE)
      INCLUDE (DIFFCOEF)


*-------------------------------------------+
*  Zero element force and stiffness matricis  |
*-------------------------------------------+

 1    DO 450 I=1,IDMEK1
         ELF(I) = 0.0Q0
         DO 451 J=1,IDMEK2
            ELK(I,J) = 0.0Q0
 451        CONTINUE
 450     CONTINUE

      DO 453 I=1,IDMCM1
         TELF(I) = 0.0Q0
         DO 454 J=1,IDMCM2
            TELK(I,J) = 0.0Q0
 454        CONTINUE
 453     CONTINUE

*--------------------------+
*  Get global node numbers  |
*--------------------------+

      NODE1 = ICON(IELEM,1)
      NODE2 = ICON(IELEM,2)
      NODE3 = ICON(IELEM,3)

*-------------------------------+
*  Loop over integration points  |
*-------------------------------+
      DO 701 ITR=1,IGP

       CALL SHAPE(NNELEM,ITR,IGP,SHPFCT,DERSHP)

*-----------------------------------------------------+
*  Value of R in real coordinates of integration point  |
*-----------------------------------------------------+

       RGP = SHPFCT(1)*R(NODE1)+
     +       SHPFCT(2)*R(NODE2)+
     +       SHPFCT(3)*R(NODE3)

*-------------------------------------------+
*  Calculate Jacobian matrix and determinant  |
*-------------------------------------------+

      AJMAT = DERSHP(1) * R(ICON(IELEM,1))
     +      + DERSHP(2) * R(ICON(IELEM,2))
     +      + DERSHP(3) * R(ICON(IELEM,3))

      AJDET = AJMAT

*-----------------------------------------------------------+
*  Calculate numerical integration multipliers times R**2  |
*-----------------------------------------------------------+

      WMULT = WEIGHT(ITR,IGP)*RGP**2*AJDET

*--------------------+
*  Calculate source  |
*--------------------+

      SOURCE = ATMSRC(IMAT(IELEM))

*-------------------------+
*  Loop over matrix rows  |
*-------------------------+

         CONVIP = SHPFCT(1)*(CONCV(NODE1) + DELCNV(NODE1)) +
     +            SHPFCT(2)*(CONCV(NODE2) + DELCNV(NODE2)) +
     +            SHPFCT(3)*(CONCV(NODE3) + DELCNV(NODE3))
         CONVIP = QMAX1(CONVIP,0.Q0)
         CONIIP = SHPFCT(1)*(CONCI(NODE1) + DELCNI(NODE1)) +
     +            SHPFCT(2)*(CONCI(NODE2) + DELCNI(NODE2)) +
     +            SHPFCT(3)*(CONCI(NODE3) + DELCNI(NODE3))
         CONIIP = QMAX1(CONIIP,0.Q0)
          TOTSRC = 0.Q0

         DO 106 N=1,NBBLSZ

            IDOF1 = ICONDF(IELEM,N)
            IDOF2 = ICONDF(IELEM,(NBBLSZ+N))
```

```
      IDOF3 = ICONDF(IELEM,(2*NBBLSZ+N))
      CONCIP(N) = SHPFCT(1)*(CONC(IDOF1) + DELCON(IDOF1)) +
     +            SHPFCT(2)*(CONC(IDOF2) + DELCON(IDOF2)) +
     +            SHPFCT(3)*(CONC(IDOF3) + DELCON(IDOF3))
      DELCIP(N) = SHPFCT(1)*DELCON(IDOF1) +
     +            SHPFCT(2)*DELCON(IDOF2) +
     +            SHPFCT(3)*DELCON(IDOF3)
      DERCIP(N) = DERSHP(1)*(CONC(IDOF1) + DELCON(IDOF1)) +
     +            DERSHP(2)*(CONC(IDOF2) + DELCON(IDOF2)) +
     +            DERSHP(3)*(CONC(IDOF3) + DELCON(IDOF3))

      RADIP(N) = SHPFCT(1)*RADIUS(NODE1,N) +
     +           SHPFCT(2)*RADIUS(NODE2,N) +
     +           SHPFCT(3)*RADIUS(NODE3,N)
      CNRDIP(N) = SHPFCT(1)*(RAD(IDOF1) + DELRAD(IDOF1)) +
     +            SHPFCT(2)*(RAD(IDOF2) + DELRAD(IDOF2)) +
     +            SHPFCT(3)*(RAD(IDOF3) + DELRAD(IDOF3))
      DLRDIP(N) = SHPFCT(1)*DELRAD(IDOF1) +
     +            SHPFCT(2)*DELRAD(IDOF2) +
     +            SHPFCT(3)*DELRAD(IDOF3)
      DRRDIP(N) = DERSHP(1)*(RAD(IDOF1) + DELRAD(IDOF1)) +
     +            DERSHP(2)*(RAD(IDOF2) + DELRAD(IDOF2)) +
     +            DERSHP(3)*(RAD(IDOF3) + DELRAD(IDOF3))
      VELIP(N) = 3.Q0 / (BLTZMN*(TEMP(1)**2)*(LTPARM**3))
     +          *DIFFS*TMPGR(1)*PI*(RADIP(N)**3)*SRFACT
      BPRSA(N) = QFLOAT(N)*BLTZMN*TEMP(1)*3.Q0/(4.Q0*PI)
      BPRSB(N) = QFLOAT(N)*VANDWL(2)*3.Q0/(4.Q0*PI)
      BPTMIP(N) = BPRSA(N)/(CNRDIP(N)**3-BPRSB(N))
      IF (BPTMIP(N).LT.0.Q0.OR.BPTMIP(N).GT.YPRS) BPTMIP(N)=YPRS
      BPRSIP(N) = SHPFCT(1)*BBLPRS(NODE1,N) +
     +            SHPFCT(2)*BBLPRS(NODE2,N) +
     +            SHPFCT(3)*BBLPRS(NODE3,N)
      IF (BPRSIP(N).LT.0.OR.BPRSIP(N).GT.YPRS) BPRSIP(N)=YPRS

      DFCFIP(N) = SHPFCT(1)*DIFFCF(NODE1,N) +
     +            SHPFCT(2)*DIFFCF(NODE2,N) +
     +            SHPFCT(3)*DIFFCF(NODE3,N)
      IF (RADIP(N).GT.15.Q-8) THEN
         RESOL(N) = RESOLP*QFLOAT(N)*
     +              (1.Q0-(1.Q0-15.Q-8/RADIP(N)-RESADJ)**3)
         ELSE
         RESOL(N) = RESOLP*QFLOAT(N)
         ENDIF
         SRCFRC(N) = 8.40128Q0*PI*OMEGA*RADIP(N)**2/LTPARM**2
         TOTSRC = TOTSRC + SRCFRC(N)*CONCIP(N)
            TOTSRC = QMIN1(TOTSRC,1.Q0)

  106       CONTINUE

          DO 101 N=2,NBBLSZ
*-----------------+
*  Zero constants  |
*-----------------+

            ALPHA = 0.Q0
            BETA  = 0.Q0
            GAMMA = 0.Q0

*-------------------------+
*  Set up resolution loss  |
*-------------------------+
            ALPHA = ALPHA+RESOL(N)*CONCIP(N)*CNRDIP(N)/3.Q0
            BETA = BETA-RESOL(N)*CONCIP(N)/3.Q0

*--------------------------------+
*  Set up self-coalescence loss   |
*--------------------------------+
            IF ((2*N).LE.NBBLSZ) THEN
               ALPHA = ALPHA +32.Q0*PI*CONCIP(N)**2*DFCFIP(N)/3.Q0
     +                        *RADIP(N)*CNRDIP(N)
               BETA  = BETA - 32.Q0*PI*CONCIP(N)**2*DFCFIP(N)/3.Q0
     +                        *RADIP(N)
               ENDIF

*----------------------+
*  Set up source loss  |
*----------------------+

            ALPHA = ALPHA+SRCFRC(N)*CONCIP(N)*CNRDIP(N)/3.Q0
            BETA = BETA-SRCFRC(N)*CONCIP(N)/3.Q0

*----------------+
* Vacancy gain   |
*----------------+

            ALPHA = ALPHA-OMEGA*DIFFV*CONCIP(N)*CONVIP
     +              *RADIP(N)/CNRDIP(N)**2
            BETA  = BETA - 2.Q0*OMEGA*DIFFV*CONCIP(N)*CONVIP
     +              *RADIP(N)/CNRDIP(N)**3
```

```
*----------------+
* Vacancy loss  |
*----------------+

          ALPHA = ALPHA+OMEGA*DIFFV*CONCIP(N)*CONCVE(1,2)
     +            *RADIP(N)/CNRDIP(N)**2
     +            *QEXP(OMEGA/(BLTZMN*TEMP(1))*
     +             (2.Q0*SRFENG/RADIP(N)-BPRSIP(N)+PRESS(1)))

          BETA = BETA+OMEGA*DIFFV*CONCIP(N)*CONCVE(1,2)
     +            *RADIP(N)/CNRDIP(N)**2
     +            *QEXP(OMEGA/(BLTZMN*TEMP(1))
     +            *(2.Q0*SRFENG/CNRDIP(N)-BPTMIP(N)+PRESS(1)))
     +            *(2.Q0/CNRDIP(N)+2.Q0*OMEGA*SRFENG/(BLTZMN*TEMP(1)
     +            *CNRDIP(N)**2)-3.Q0*OMEGA*BPRSA(N)*CNRDIP(N)**2
     +            /(BLTZMN*TEMP(1)*(CNRDIP(N)**3-BPRSB(N))**2))

*--------------------+
* Interstitial loss  |
*--------------------+

          ALPHA = ALPHA+OMEGA*CONCIP(N)*DIFFI*CONIIP
     +            *RADIP(N)/CNRDIP(N)**2
          BETA = BETA+OMEGA*CONCIP(N)*DIFFI*CONIIP
     +            *RADIP(N)/CNRDIP(N)**3

*--------------------+
* Time differential  |
*--------------------+
          ALPHA = ALPHA+(CONCIP(N)*DLRDIP(N)
     +            +CNRDIP(N)*DELCIP(N)/3.Q0)
     +            /DELTIM(IPRD)
          BETA = BETA-(CONCIP(N)+DELCIP(N)/3.Q0)/DELTIM(IPRD)

          TELK(N,N) = BETA
          TELF(N) = ALPHA

*----------------------+
*  Index limit for ROE  |
*----------------------+

          NCHECK = NBBLSZ - N

*-------------------------------------------------+
*  Loop over contributions from other bubble groups  |
*-------------------------------------------------+

          DO 102 K=1,NBBLSZ

*------------------+
*  Zero constants  |
*------------------+

          ALPHA = 0.Q0
          BETAN = 0.Q0
          BETAK = 0.Q0
          BETAJ = 0.Q0
          ETA = 0.Q0
          EPSILN = 0.Q0

*-------------------+
*  Resolution gain  |
*-------------------+

          IF (K.EQ.(N+1)) THEN
             ALPHA = ALPHA - RESOL(K)*CONCIP(K)/3.Q0
     +               *CNRDIP(K)**3/CNRDIP(N)**2
             BETAN = BETAN - RESOL(K)*CONCIP(K)/3.Q0
     +               *CNRDIP(K)**3/CNRDIP(N)**2
     +               *2.Q0/CNRDIP(N)

             BETAK = BETAK + RESOL(K)*CONCIP(K)/3.Q0
     +               *CNRDIP(K)**3/CNRDIP(N)**2
     +               *3.Q0/CNRDIP(K)
          ENDIF

*---------------+
*  Source gain  |
*---------------+

          IF (K.EQ.(N-1)) THEN
             ALPHA = ALPHA - SRCFRC(K)*CONCIP(K)/3.Q0
     +               *CNRDIP(K)**3/CNRDIP(N)**2
             BETAN = BETAN - SRCFRC(K)*CONCIP(K)/3.Q0
     +               *CNRDIP(K)**3/CNRDIP(N)**2
     +               *2.Q0/CNRDIP(N)
             BETAK = BETAK + SRCFRC(K)*CONCIP(K)/3.Q0
     +               *CNRDIP(K)**3/CNRDIP(N)**2
     +               *3.Q0/CNRDIP(K)
          ENDIF
```

```
            IF (K.EQ.N) GOTO 102

*-------------------+
*  Coalescence loss  |
*-------------------+

            IF (K.LE.NCHECK) THEN
               ALPHA = ALPHA+4.Q0*PI*CONCIP(N)*CONCIP(K)/3.Q0
     +                     *(RADIP(N)+RADIP(K))
     +                     *(DFCFIP(N)+DFCFIP(K))
     +                     *CNRDIP(N)
     +                     +PI*CONCIP(N)*CONCIP(K)/3.Q0
     +                     *(RADIP(N)+RADIP(K))**2
     +                     *QABS(VELIP(N)-VELIP(K))
     +                     *CNRDIP(N)
               BETAN = BETAN-4.Q0*PI*CONCIP(N)*CONCIP(K)/3.Q0
     +                     *(RADIP(N)+RADIP(K))
     +                     *(DFCFIP(N)+DFCFIP(K))
     +                     +PI*CONCIP(N)*CONCIP(K)/3.Q0
     +                     *(RADIP(N)+RADIP(K))**2
     +                     *QABS(VELIP(N)-VELIP(K))
               ENDIF

*-------------------+
*  Coalescence gain  |
*-------------------+

            J = N-K
            IF (J.GT.0.AND.K.GE.J) THEN
               ALPHA = ALPHA-4.Q0*PI*CONCIP(K)*CONCIP(J)
     +                     /(3.Q0*CNRDIP(N)**2)
     +                     *(RADIP(J)+RADIP(K))
     +                     *(DFCFIP(J)+DFCFIP(K))
     +                     *(CNRDIP(K)**3+CNRDIP(J)**3)
     +                -PI*CONCIP(J)*CONCIP(K)/(3.Q0*CNRDIP(N)**2)
     +                 *(RADIP(K)+RADIP(J))**2
     +                 *QABS(VELIP(K)-VELIP(J))
     +                 *(CNRDIP(K)**3+CNRDIP(J)**3)
               BETAN = BETAN-8.Q0*PI*CONCIP(K)*CONCIP(J)
     +                     /(3.Q0*CNRDIP(N)**2)
     +                     *(RADIP(J)+RADIP(K))
     +                     *(DFCFIP(J)+DFCFIP(K))
     +                     *(CNRDIP(K)**3+CNRDIP(J)**3)/CNRDIP(N)
     +                -PI*CONCIP(J)*CONCIP(K)/(3.Q0*CNRDIP(N)**2)
     +                 *(RADIP(K)+RADIP(J))**2
     +                 *QABS(VELIP(K)-VELIP(J))
     +                 *(CNRDIP(K)**3+CNRDIP(J)**3)
     +                 *2.Q0/CNRDIP(N)
               BETAK = BETAK+4.Q0*PI*CONCIP(K)*CONCIP(J)
     +                     /(3.Q0*CNRDIP(N)**2)
     +                     *(RADIP(J)+RADIP(K))
     +                     *(DFCFIP(J)+DFCFIP(K))
     +                     *3.Q0*CNRDIP(K)**3/CNRDIP(K)
     +                +PI*CONCIP(J)*CONCIP(K)/(3.Q0*CNRDIP(N)**2)
     +                 *(RADIP(K)+RADIP(J))**2
     +                 *QABS(VELIP(K)-VELIP(J))
     +                 *CNRDIP(K)**3
     +                 *3.Q0/CNRDIP(K)
               BETAJ = BETAJ+4.Q0*PI*CONCIP(K)*CONCIP(J)
     +                     /(3.Q0*CNRDIP(N)**2)
     +                     *(RADIP(J)+RADIP(K))
     +                     *(DFCFIP(J)+DFCFIP(K))
     +                     *3.Q0*CNRDIP(J)**3/CNRDIP(J)
     +                +PI*CONCIP(J)*CONCIP(K)/(3.Q0*CNRDIP(N)**2)
     +                 *(RADIP(K)+RADIP(J))**2
     +                 *QABS(VELIP(K)-VELIP(J))
     +                 *CNRDIP(J)**3
     +                 *3.Q0/CNRDIP(J)
               TELK(N,J) = TELK(N,J) + BETAJ
               ENDIF

            TELK(N,N) = TELK(N,N) + BETAN
            TELK(N,K) = TELK(N,K) + BETAK
            TELF(N) = TELF(N) + ALPHA
  102       CONTINUE

*-----------------------------------------------------+
*  Fill out rest of matrix and include diffusive terms  |
*-----------------------------------------------------+

            DO 103 ND1=NNELEM,1,-1
               IND1 = (ND1-1)*NBBLSZ + N
               DO 104 L=1,NBBLSZ
                  DO 105 ND2=NNELEM,1,-1
                    IND2 = (ND2-1)*NBBLSZ + L
                       IF (N.NE.L) THEN
                          TE' '(IND1,IND2) = TELK(N,L)*WMULT*SHPFCT(ND1)
     +                                                    *SHPFCT(ND2)
                       ELSE
```

```
                    TELK(IND1,IND2) = TELK(N,N)*WMULT*SHPFCT(ND1)
     +                                                *SHPFCT(ND2)
     +                                +WMULT*
     +                                (-DFCFIP(N)/(3.Q0*AJDET**2)
     +                                 *DERCIP(N)
     +                                 *DERSHP(ND1)*SHPFCT(ND2))
     +                                +WMULT*
     +                                (-DFCFIP(N)/(3.Q0*AJDET**2)
     +                                 *DERCIP(N)
     +                                 *DERSHP(ND2)*SHPFCT(ND1))
  103         CONTINUE
  101       CONTINUE

        CALL MTXADD(TELK,ELK,NEDOF,NEDOF,IELEM,
     +             IDMCM1,IDMCM2,IDMCM3,IDMEK1,IDMEK2,IDMEK3,
     +             .FALSE.,.FALSE.)
*-----------------------------------------------+
*  Add load vector into overall force vector    |
*-----------------------------------------------+
      CALL MTXADD(TELF,ELF,NEDOF,1,IELEM,
     +            IDMCV1,IDMCV2,IDMCV3,IDMEF1,IDMEF2,IDMEF3,
     +            .FALSE.,.FALSE.)

  701 CONTINUE
      RETURN
      END
*=================================================+
*                                                 |
*          EEEEEEEE  LL        KK    KK  FFFFFFFF  VV     VV   |
*          EEEEEEEE  LL        KK   KK   FFFFFFFF  VV     VV   |
*          EE        LL        KK  KK    FF        VV     VV   |
*          EEEEE     LL        KKKKK     FFFFF     VV     VV   |
*          EEEEE     LL        KKKKK     FFFFF     VV     VV   |
*          EE        LL        KK  KK    FF         VV   VV    |
*          EEEEEEEE  LLLLLLLL  KK   KK   FF          VVVV     |
*          EEEEEEEE  LLLLLLLL  KK    KK  FF           VV      |
*                                                 |
*   This routine calculates the elemental contributions to fission |
*    gas bubbles using spherical geometry.        |
*=================================================+
      SUBROUTINE ELKFV(IELEM,IPRD,LINC,NLINC,ITERAT)

      INCLUDE (MAIN)
      INCLUDE (ELKF)
      INCLUDE (GAUSS)
      INCLUDE (PI)
      INCLUDE (SURFENGY)
      INCLUDE (LATTICE)
      INCLUDE (BOLTZMAN)
      INCLUDE (VANDWAAL)

*-----------------------------------------------+
*  Zero element force and stiffness matricis    |
*-----------------------------------------------+
      DO 450 I=1,IDMEK1
         ELF(I) = 0.0Q0
         DO 451 J=1,IDMEK2
            ELK(I,J) = 0.0Q0
  451       CONTINUE
  450    CONTINUE

*-----------------------------+
*  Get global node numbers    |
*-----------------------------+
      NODE1 = ICON(IELEM,1)
      NODE2 = ICON(IELEM,2)
      NODE3 = ICON(IELEM,3)

*-------------------------------+
*  Loop over integration points |
*-------------------------------+
      DO 701 ITR=1,IGP

       CALL SHAPE(NNELEM,ITR,IGP,SHPFCT,DERSHP)

*-------------------------------------------------------+
*  Value of R in real coordinates of integration point  |
*-------------------------------------------------------+
       RGP = SHPFCT(1)*R(NODE1)+
     +       SHPFCT(2)*R(NODE2)+
     +       SHPFCT(3)*R(NODE3)

*-----------------------------------------+
*  Calculate Jacobian matrix and determinant |
*-----------------------------------------+
      AJMAT = DERSHP(1) * R(NODE1)
```

```
     +      + DERSHP(2) * R(NODE2)
     +      + DERSHP(3) * R(NODE3)

      AJDET = AJMAT

*-------------------------------------------------------------+
*  Calculate numerical integration multipliers times R**2  |
*-------------------------------------------------------------+

      WMULT = WEIGHT(ITR,IGP)*RGP**2*AJDET

*--------------------+
*  Calculate source  |
*--------------------+

      SOURCE = VACSRC(1)

         CONVIP = SHPFCT(1)*(CONCV(NODE1) + DELCNV(NODE1)) +
     +            SHPFCT(2)*(CONCV(NODE2) + DELCNV(NODE2)) +
     +            SHPFCT(3)*(CONCV(NODE3) + DELCNV(NODE3))

         CONIIP = SHPFCT(1)*(CONCI(NODE1) + DELCNI(NODE1)) +
     +            SHPFCT(2)*(CONCI(NODE2) + DELCNI(NODE2)) +
     +            SHPFCT(3)*(CONCI(NODE3) + DELCNI(NODE3))

         DELVIP = SHPFCT(1)*DELCNV(NODE1) +
     +            SHPFCT(2)*DELCNV(NODE2) +
     +            SHPFCT(3)*DELCNV(NODE3)

         DELIIP = SHPFCT(1)*DELCNI(NODE1) +
     +            SHPFCT(2)*DELCNI(NODE2) +
     +            SHPFCT(3)*DELCNI(NODE3)

         DERVIP = DERSHP(1)*(CONCV(NODE1) + DELCNV(NODE1)) +
     +            DERSHP(2)*(CONCV(NODE2) + DELCNV(NODE2)) +
     +            DERSHP(3)*(CONCV(NODE3) + DELCNV(NODE3))

         DERIIP = DERSHP(1)*(CONCI(NODE1) + DELCNI(NODE1)) +
     +            DERSHP(2)*(CONCI(NODE2) + DELCNI(NODE2)) +
     +            DERSHP(3)*(CONCI(NODE3) + DELCNI(NODE3))

*------------------+
*  Zero constants  |
*------------------+

         ALPHAV = 0.Q0
         ALPHAI = 0.Q0
         BETAV  = 0.Q0
         BETAI  = 0.Q0

*-----------------------------------------------------+
*  Set up source and equilibrium gain from dislocations |
*-----------------------------------------------------+

         ALPHAV = SOURCE + ZVAC*DIFFV*CONCVE(1,Z)
     +            *QEXP(-OMEGA*PRESS(1)/(BLTZMN*TEMP(1)))
         ALPHAI = SOURCE
         BETAV  = - ZVAC*DIFFV
         BETAI  = - ZINT*DIFFI

*-------------------------+
*  Loop over matrix rows  |
*-------------------------+

        DO 106 N=2,NBBLSZ

           BPRSIP(N) = SHPFCT(1)*BBLPRS(NODE1,N) +
     +                 SHPFCT(2)*BBLPRS(NODE2,N) +
     +                 SHPFCT(3)*BBLPRS(NODE3,N)
           IF (BPRSIP(N).LT.0.OR.BPRSIP(N).GT.YPRS) BPRSIP(N)=YPRS

           RADIP(N) = SHPFCT(1)*RADIUS(NODE1,N) +
     +                SHPFCT(2)*RADIUS(NODE2,N) +
     +                SHPFCT(3)*RADIUS(NODE3,N)

           IDOF1 = ICONDF(IELEM,N)
           IDOF2 = ICONDF(IELEM,(NBBLSZ+N))
           IDOF3 = ICONDF(IELEM,(2*NBBLSZ+N))

           CONCIP(N) = SHPFCT(1)*(CONC(IDOF1) + DELCON(IDOF1)) +
     +                 SHPFCT(2)*(CONC(IDOF2) + DELCON(IDOF2)) +
     +                 SHPFCT(3)*(CONC(IDOF3) + DELCON(IDOF3))

*-----------------------------------------------------+
*  Add in equilibrium gain from bubbles               |
*-----------------------------------------------------+

            ALPHAV = ALPHAV + 4.Q0*PI*RADIP(N)*CONCIP(N)*DIFFV
     +               *CONCVE(1,2)
     +               *QEXP(OMEGA/(BLTZMN*TEMP(1))
```

```
     +              *(2.Q0*SRFENG/RADIP(N)-BPRSIP(N)+PRESS(1)))

*-----------------------------------------------------------+
*  Include loss to bubbles and dislocations                 |
*-----------------------------------------------------------+

            BETAV = BETAV -
     +              4.Q0*PI*RADIP(N)*CONCIP(N)*DIFFV
            BETAI = BETAI -
     +              4.Q0*PI*RADIP(N)*CONCIP(N)*DIFFI

  106       CONTINUE

*-------------------------------------------------------------------+
*  Start assembly of elemental stiffness matrix and force vector    |
*-------------------------------------------------------------------+

           TELK(1,1) = -1.Q0/DELTIM(IPRD) + BETAV
           TELK(2,2) = -1.Q0/DELTIM(IPRD) + BETAI
           TELF(1) = DELVIP/DELTIM(IPRD) - ALPHAV - BETAV*CONVIP
           TELF(2) = DELIIP/DELTIM(IPRD) - ALPHAI - BETAI*CONIIP

*-----------------------------------------------------------+
*  Include loss to recombination                            |
*-----------------------------------------------------------+

             RECOMB = 5040.Q0*OMEGA
             ROE = -RECOMB*(DIFFV+DIFFI)/LTPARM**2

*---------------------------------+
*  Add in contributions from ROE  |
*---------------------------------+

             TELK(1,1) = TELK(1,1) + ROE*CONIIP
             TELK(2,2) = TELK(2,2) + ROE*CONVIP
             TELK(1,2) = ROE*CONVIP
             TELK(2,1) = ROE*CONIIP
             TELF(1) = TELF(1) - ROE*CONVIP*CONIIP
             TELF(2) = TELF(2) - ROE*CONVIP*CONIIP

*-----------------------------------------------------------+
*  Fill out rest of matrix and include diffusive terms      |
*-----------------------------------------------------------+

           DO 103 ND1=NNELEM,1,-1
             IND1A = (ND1-1)*2+1
             IND1B = (ND1-1)*2+2
             DO 105 ND2=NNELEM,1,-1
               IND2A = (ND2-1)*2+1
               IND2B = (ND2-1)*2+2
                  TELK(IND1A,IND2A) = (-DIFFV/AJDET**2
     +                                 *DERSHP(ND1)*DERSHP(ND2)
     +                                 +TELK(1,1)*SHPFCT(ND1)
     +                                          *SHPFCT(ND2))*WMULT
                  TELK(IND1B,IND2B) = (-DIFFI/AJDET**2
     +                                 *DERSHP(ND1)*DERSHP(ND2)
     +                                 +TELK(2,2)*SHPFCT(ND1)
     +                                          *SHPFCT(ND2))*WMULT

               TELK(IND1A,IND2B) = TELK(1,2)*WMULT*SHPFCT(ND1)
     +                                          *SHPFCT(ND2)
               TELK(IND1B,IND2A) = TELK(2,1)*WMULT*SHPFCT(ND1)
     +                                          *SHPFCT(ND2)
  105          CONTINUE
             TELF(IND1A) = (DIFFV*DERSHP(ND1)/AJDET**2
     +                      *DERVIP+SHPFCT(ND1)*TELF(1))*WMULT
             TELF(IND1B) = (DIFFI*DERSHP(ND1)/AJDET**2
     +                      *DERIIP+SHPFCT(ND1)*TELF(2))*WMULT
  103        CONTINUE
  101      CONTINUE

        CALL MTXADD(TELK,ELK,NEDOF,NEDOF,IELEM,
     +              IDMCM1,IDMCM2,IDMCM3,IDMEK1,IDMEK2,IDMEK3,
     +              .FALSE.,.FALSE.)
*-----------------------------------------------+
*  Add load vector into overall force vector    |
*-----------------------------------------------+
       CALL MTXADD(TELF,ELF,NEDOF,1,IELEM,
     +             IDMCV1,IDMCV2,IDMCV3,IDMEF1,IDMEF2,IDMEF3,
     +             .FALSE.,.FALSE.)

  701  CONTINUE
       RETURN
       END
*=================================================================+
*                                                                 |
*      EEEEEEEE   QQQQQQ   UU    UU  RRRRRRR    AAAAAA   DDDDDDD  |
*      EEEEEEEE  QQQQQQQQ  UU    UU  RRRRRRRR  AAAAAAAA  DDDDDDDD |
*      EE        QQ    QQ  UU    UU  RR    RR  AA    AA  DD    DD |
*      EEEEE     QQ    QQ  UU    UU  RR    RR  AA    AA  DD    DD |
```

```
*      EEEEE     QQ    QQ  UU    UU  RRRRRRRR  AAAAAAAA  DD    DD      |
*      EE        QQ  QQQQ  UU    UU  RRRRRR    AAAAAAAA  DD    DD      |
*      EEEEEEEE  QQQQQQQQ  UUUUUUUU  RR    RR  AA    AA  DDDDDDDD      |
*      EEEEEEEE   QQQQQQ    UUUUUU   RR    RR  AA    AA  DDDDDD        |
*                                                                        |
*                                                                        |
*     Calculates equilibrium bubble sizes                                |
*     Input is N     - the number of atoms in bubble                     |
*              TEMP  - temperature of the bubble                         |
*              PRESS - external pressure                                 |
*              ROLD  - previous radius                                   |
*========================================================================+

      REAL FUNCTION EQURAD(N,TEMP,PRESS,ROLD)

      IMPLICIT REAL*16 (A-H,O-Z)
      IMPLICIT INTEGER (I-M)

      INCLUDE (PI)
      INCLUDE (BOLTZMAN)
      INCLUDE (SURFENGY)
      INCLUDE (VANDWAAL)


      ITMAX = 1000
      CONLIM = 1.Q-10

      ALPHA = 4.Q0*PI*PRESS/3.Q0
      BETA = 8.Q0*SRFENG*PI/3.Q0
      GAMMA = -QFLOAT(N)*(PRESS*VANDWL(2) + BLTZMN*TEMP)
      ROE = -2.Q0*SRFENG*VANDWL(2)*QFLOAT(N)


      DO 101 I=1,ITMAX
         RNEW = ROLD - (ALPHA*ROLD**4 + BETA*ROLD**3 + GAMMA*ROLD + ROE)
     +               / (4.Q0*ALPHA*ROLD**3 + 3.Q0*BETA*ROLD**2 + GAMMA)

         RATIO = QABS((RNEW-ROLD)/RNEW)

         IF (RATIO.LT.CONLIM) GOTO 102

         ROLD = RNEW

  101    CONTINUE

         CALL ERROR('Maximum Iterations in EQURAD')

  102 EQURAD = RNEW
      RETURN
      END
```

```
*========================================================================+
*                                                                        |
*          EEEEEEEE  RRRRRRR   RRRRRRR    000000   RRRRRRR              |
*          EEEEEEEE  RRRRRRRR  RRRRRRRR  00000000  RRRRRRRR             |
*          EE        RR    RR  RR    RR  00    00  RR    RR             |
*          EEEEE     RR    RR  RR    RR  00    00  RR    RR             |
*          EEEEE     RRRRRRRR  RRRRRRRR  00    00  RRRRRRRR             |
*          EE        RRRRRR    RRRRRR    00    00  RRRRRR               |
*          EEEEEEEE  RR    RR  RR    RR  00000000  RR    RR             |
*          EEEEEEEE  RR    RR  RR    RR   000000   RR    RR             |
*                                                                        |
*                                                                        |
*     Outputs the specified error                                        |
*                                                                        |
*========================================================================+

      SUBROUTINE ERROR(A)

      CHARACTER*40 A

      WRITE(8,999)
  999 FORMAT(/' ***************** ERROR ENCOUNTERED ****************')

      WRITE(8,998) A
  998 FORMAT(' ',A50)

      WRITE(8,997)
  997 FORMAT(' ************************************************')

      STOP
      END
```

```
*========================================================================+
*                                                                        |
*      EEEEEEEE  XX    XX  PPPPPPP    AAAAAA   NN    NN  DDDDDDD       |
*      EEEEEEEE  XX    XX  PPPPPPPP  AAAAAAAA  NNN   NN  DDDDDDDD      |
*      EE         XX  XX   PP    PP  AA    AA  NNNN  NN  DD    DD      |
*      EEEEE       XXXX    PP    PP  AA    AA  NN NN NN  DD    DD      |
*      EEEEE       XXXX    PPPPPPPP  AAAAAAAA  NN  NNNN  DD    DD      |
*      EE         XX  XX   PPPPPPP   AAAAAAAA  NN   NNN  DD    DD      |
*      EEEEEEEE  XX    XX  PP        AA    AA  NN    NN  DDDDDDDD      |
```

```
*      EEEEEEEE  XX    XX  PP        AA    AA  NN    NN  DDDDDDD
*
*
*   Takes the solved nonprescribed displacements and adds them into
*     the prescribed displacement vector UP.  Keeps track of
*     delta Q's, and total Q's
*
*-------------------------------------------------------------------
      SUBROUTINE EXPAND(ITYPE)

*-----------------------------------------------+
*  Get declarations                             |
*-----------------------------------------------+
      INCLUDE (MAIN)

       IF (ITYPE.EQ.1) THEN
          DO 102 I=1,INONPD
             IP = JARRAY(I)
             IF ((CONC(IP)+DELCON(IP)+RRBC(I)).LT.0.Q0)
     +          RRBC(I) =  -CONC(IP) - DELCON(IP)
             DELCON(IP) = DELCON(IP) + RRBC(I)
 102         CONTINUE
          ENDIF

       IF (ITYPE.EQ.2) THEN
          DO 103 I=1,NN-1
             INC = (I-1)*2
             IF ((CONCV(I)+DELCNV(I)+RRBC(INC+1)).LT.0.Q0)
     +          RRBC(INC+1) =  -CONCV(I) - DELCNV(I)
             IF ((CONCI(I)+DELCNI(I)+RRBC(INC+2)).LT.0.Q0)
     +          RRBC(INC+2) =  -CONCI(I) - DELCNI(I)
             DELCNV(I) = DELCNV(I) + RRBC(INC+1)
             DELCNI(I) = DELCNI(I) + RRBC(INC+2)
 103         CONTINUE
          ENDIF

       IF (ITYPE.EQ.3) THEN
          INC = 0
          DO 104 I=1,NN-1
             DO 105 J=2,NBBLSZ
                INC = INC + 1
                IJ = (I-1)*NBBLSZ + J
                IF ((RAD(IJ)+DELRAD(IJ)+RRBC(INC)).LT.RADMIN(J))
     +             RRBC(INC) =  RADMIN(J) - RAD(IJ) - DELRAD(IJ)
                IF ((RAD(IJ)+DELRAD(IJ)+RRBC(INC)).GT.RADMAX(J))
     +             RRBC(INC) =  RADMAX(J) - RAD(IJ) - DELRAD(IJ)
                DELRAD(IJ) = DELRAD(IJ) + RRBC(INC)
 105         CONTINUE
 104      CONTINUE
          ENDIF

       RETURN
       END
*-------------------------------------------------------------------
*
*     FFFFFFFF  RRRRRRR   TTTTTTTT  PPPPPPP    AAAAAA    GGGGGG
*     FFFFFFFF  RRRRRRRR  TTTTTTTT  PPPPPPPP  AAAAAAAA  GGGGGGGG
*     FF        RR    RR     TT     PP    PP  AA    AA  GG
*     FFFFF     RR    RR     TT     PP    PP  AA    AA  GG  GGGG
*     FFFFF     RRRRRRRR     TT     PPPPPPPP  AAAAAAAA  GG  GGGG
*     FF        RRRRRR       TT     PPPPPPP   AAAAAAAA  GG    GG
*     FF        RR   RR      TT     PP        AA    AA  GGGGGGGG
*     FF        RR    RR     TT     PP        AA    AA   GGGGGG
*
*
*   Prints the nodal displacements and elemental stresses, strains
*
*-------------------------------------------------------------------
      SUBROUTINE FRTPAG

      WRITE(8,1)
    1 FORMAT('1'////////
     +     ' RRRRRRR    EEEEEEEE  DDDDDDD    SSSSSS  '
     +    ,' TTTTTTTT   OOOOOO   NN    NN  EEEEEEEE '
     +    /' RRRRRRRR   EEEEEEEE  DDDDDDDD  SSSSSSSS '
     +    ,' TTTTTTTT  OOOOOOOO  NNN   NN  EEEEEEEE '
     +    /' RR    RR   EE        DD    DD  SS       '
     +    ,'    TT     OO    OO  NNNN  NN  EE       '
     +    /' RR    RR   EEEEEE    DD    DD  SSSSSSS  '
     +    ,'    TT     OO    OO  NN NN NN  EEEEEE   '
     +    /' RRRRRRRR  EEEEEE    DD    DD   SSSSSSS '
     +    ,'    TT     OO    OO  NN  NNNN  EEEEEE   '
     +    /' RRRRRR    EE        DD    DD        SS '
     +    ,'    TT     OO    OO  NN   NNN  EE       '
     +    /' RR   RR   EEEEEEEE  DDDDDDDD  SSSSSSSS '
     +    ,'    TT     OOOOOOOO  NN    NN  EEEEEEEE '
     +    /' RR    RR  EEEEEEEE  DDDDDDD    SSSSSS  '
     +    ,'    TT     OOOOOO   NN    NN  EEEEEEEE '
     +    //////)
```

```
      RETURN
      END
*=====================================================================+
*                                                                     |
*          IIIIIIII   CCCCCC    000000   NN     NN  DDDDDDD          |
*          IIIIIIII  CCCCCCCC  00000000  NNN    NN  DDDDDDDD         |
*             II     CC    CC 00    00  NNNN   NN  DD     DD         |
*             II     CC       00    00  NN NN NN  DD     DD         |
*             II     CC       00    00  NN  NNNN  DD     DD         |
*             II     CC    CC 00    00  NN   NNN  DD     DD         |
*          IIIIIIII  CCCCCCCC  00000000  NN    NN  DDDDDDDD         |
*          IIIIIIII   CCCCCC    000000   NN    NN  DDDDDDD          |
*                                                                     |
*                                                                     |
*   This subroutine calculates the degree of freedom                  |
*    connectivity arrays                                              |
*                                                                     |
*=====================================================================+
      SUBROUTINE ICOND

*-------------------------------------------+
*  Get declarations                         |
*-------------------------------------------+

      INCLUDE (MAIN)

*-------------------------------------------+
*  ICONDF - connectivity dof array          |
*  IDOFPD - prescribed dof array            |
*  UP - displacement vector (including      |
*       prescribed dofs                     |
*-------------------------------------------+

      DO 101 I=1,NE
         DO 104 J=1,NBBLSZ*4
            ICONDF(I,J) = 0
 104        CONTINUE
         DO 102 J=1,NNELEM
            DO 103 L=1,NBBLSZ
               INDEX = NBBLSZ*(J-1)+L
               ICONDF(I,INDEX) = NBBLSZ*(ICON(I,J)-1) + L
 103           CONTINUE
 102        CONTINUE
 101     CONTINUE

      DO 105 I=1,NSPC
         IDOFPD(I) = NBBLSZ*(NPD(I)-1) + IDIR(I)
         CONC(IDOFPD(I)) = SPDISP(I)
 105     CONTINUE

      RETURN
      END
*=====================================================================+
*                                                                     |
*          IIIIIIII  NN    NN  PPPPPPP   UU    UU  TTTTTTTT          |
*          IIIIIIII  NNN   NN  PPPPPPPP  UU    UU  TTTTTTTT          |
*             II     NNNN  NN  PP    PP  UU    UU     TT             |
*             II     NN NN NN  PP    PP  UU    UU     TT             |
*             II     NN  NNNN  PPPPPPPP  UU    UU     TT             |
*             II     NN   NNN  PPPPPPP   UU    UU     TT             |
*          IIIIIIII  NN    NN  PP        UUUUUUUU     TT             |
*          IIIIIIII  NN    NN  PP         UUUUUU      TT             |
*                                                                     |
*  SUBROUTINE INPUT reads in the input data and puts it into          |
*   arrays, generates nodes and elements, etc., and outputs           |
*   input values                                                      |
*=====================================================================+

      SUBROUTINE INPUT(MAXITR,NLINC,SUBINC,IUNITS,LOUT,CONLIM)

*-------------------------------------------------+
*  Get declarations                               |
*-------------------------------------------------+
      INCLUDE (MAIN)
      INCLUDE (INOUT)

      LOGICAL SUBINC

*-------------------------------------------------+
*  Get date and time of run for output            |
*-------------------------------------------------+
      IDATE(1) = -1
C     CALL DATIMX(IDATE)
      IF (IDATE(1).EQ.-1) THEN
         DATE = '*** **, ****, *********'
         TIME = '**:** **'
         ELSE
         INDEX = ((IDATE(12)-1)*9)+1
         DATE(15:23) = DAYS(INDEX:INDEX+8)
         DATE(7:7) = ','
```

```
      DATE(13:13) = ','
      INDEX = ((IDATE(7)-1)*3)+1
      DATE(1:3) = MONTHS(INDEX:INDEX+2)
      WRITE(UNIT=DATE(5:6),FMT='(I2)') IDATE(6)
      WRITE(UNIT=DATE(9:12),FMT='(I4)') IDATE(8)
      WRITE(UNIT=TIME(1:2),FMT='(I2)') IDATE(10)
      TIME(3:3) = ':'
      IF (IDATE(10).GE.10) THEN
         WRITE(UNIT=TIME(4:5),FMT='(I2)') IDATE(4)
         ELSE
         WRITE(UNIT=TIME(5:5),FMT='(I1)') IDATE(4)
         TIME(4:4) = '0'
         ENDIF
      INDEX = ((IDATE(11)-1)*2)+1
      TIME(7:8) = AMPM(INDEX:INDEX+1)
      ENDIF

*-----------------------------------------------------+
* Initialize error checking indicis                   |
*-----------------------------------------------------+
      IHDR = 0
      INN  = 0
      INCON = 0
      INPRP = 0
      INMAT = 0
      INHMAT = 0
      INPRD = 0
      IPRD = 1
      INDSP = 0
      INNTR = 0
      INNFOR = 0
      INNBOD = 0
      INNTEM = 0

*-----------------------------------------------------+
*  Read next line of input deck                       |
*   RDCARD also outputs lines read                    |
*   Only non-comment cards are returned               |
*-----------------------------------------------------+
      CALL RDCARD(BUFFER,ICRD,NOFLDS,IFLD,IEND,0)

*-----------------------------------------------------+
*  If end of deck goto 1101                           |
*-----------------------------------------------------+
      IF (IEND.EQ.1) GOTO 1101

      IF (ICRD.NE.0) THEN
         CALL ERROR('List of input deck flag not found.         ')
         ELSE
         READ(UNIT=BUFFER(1:IFLD(1)),FMT=*) IOUT
         ENDIF

 2225 CALL RDCARD(BUFFER,ICRD,NOFLDS,IFLD,IEND,0)

*-----------------------------------------------------+
*  If end of deck goto 1101                           |
*-----------------------------------------------------+
      IF (IEND.EQ.1) GOTO 1101

*-----------------------------------------------------+
*  Goto appropriate input for specific                |
*   card numbers                                      |
*-----------------------------------------------------+
      IF (ICRD/100.NE.1) GOTO 3333

*-----------------------------------------------------+
*  Read header info                                   |
*-----------------------------------------------------+
      IHDR = IHDR+1
      HEADER(IHDR) = BUFFER
      GOTO 2225

 3333 REWIND (UNIT=5)
      CALL FRTPAG
*-----------------------------------------------------+
*  Output header information                          |
*-----------------------------------------------------+
C     WRITE(8,2000)  TIME,DATE
C2000 FORMAT(' ===== PROGRAM REDSTONE ======= (',A8,
C    +             ') ======= (',A23,')')
      WRITE(8,2000)
 2000 FORMAT(' ======= PROGRAM REDSTONE ==========',
     +            '=============================='/)
      DO 1102 I=1,IHDR
         WRITE(8,2001) HEADER(I)
 2001    FORMAT(' ',A80)
 1102    CONTINUE
      WRITE(8,1999)
 1999 FORMAT(/' ',80('='))
```

```
*-------------------------------------------------+
*  Start output of input deck in form read        |
*-------------------------------------------------+
      IF (IOUT.EQ.1) WRITE(8,4444)
 4444 FORMAT('1'//30X,'LISTING OF INPUT DECK'//)

*-------------------------------------------------+
*  Read next line of input deck                   |
*   RDCARD also outputs lines read                |
*   Only non-comment cards are returned           |
*-------------------------------------------------+
 1111 CALL RDCARD(BUFFER,ICRD,NOFLDS,IFLD,IEND,IOUT)

*-------------------------------------------------+
*  If end of deck goto 1101                       |
*-------------------------------------------------+
      IF (IEND.EQ.1) GOTO 1101

*-------------------------------------------------+
*  Goto appropriate input for specific            |
*   card numbers                                  |
*-------------------------------------------------+
      IF (ICRD.EQ.0) GOTO 1111
      GOTO (5001,5002,5003,5004,5005,5006,5007,5008,5009,5010,
     +      5011) ICRD/100

*-------------------------------------------------+
*  If no math then unknown card                   |
*-------------------------------------------------+
      CALL ERROR('Unidentified card number.                    ')

*-------------------------------------------------+
*  Read header info                               |
*-------------------------------------------------+
 5001 IHDR = IHDR+1
      GOTO 1111

*-------------------------------------------------+
*  Read in                                        |
*   IUNITS - 1=British                            |
*            2=Metric                             |
*   IGP - Number of Guass points                  |
*-------------------------------------------------+
 5002 GOTO (5102,5202,5302,5402) ICRD-200
      CALL ERROR('Unidentified card number in 200s             ')

 5102 READ(UNIT=BUFFER(1:IFLD(1)),FMT=*) IUNITS
      GOTO 1111
 5202 READ(UNIT=BUFFER(1:IFLD(1)),FMT=*) IGP
      GOTO 1111
 5302 READ(UNIT=BUFFER(1:IFLD(1)),FMT=*) IDECK
      GOTO 1111
 5402 READ(UNIT=BUFFER(1:IFLD(1)),FMT=*) IREST
      GOTO 1111

*-------------------------------------------------+
*  Read in:                                       |
*   NE    - total number of elements              |
*   NN    - total number of nodes                 |
*   NBBLSZ - number of bubble groups to track     |
*   NNELEM - number of nodes per element          |
*   NMAT - number of material types               |
*   NSPC - number of specified concentrations     |
*   NPRD - number of time periods                 |
*   NLINC - number of load increments             |
*-------------------------------------------------+

 5003 IF (NOFLDS.NE.6)
     +CALL ERROR('Incorrect number of fields on card 300          ')
      READ(UNIT=BUFFER(1:IFLD(1)),FMT=*) NE
      READ(UNIT=BUFFER(IFLD(1)+1:IFLD(2)),FMT=*) NBBLSZ
      READ(UNIT=BUFFER(IFLD(2)+1:IFLD(3)),FMT=*) NNELEM
      READ(UNIT=BUFFER(IFLD(3)+1:IFLD(4)),FMT=*) NSPC
      READ(UNIT=BUFFER(IFLD(4)+1:IFLD(5)),FMT=*) NPRD
      READ(UNIT=BUFFER(IFLD(5)+1:IFLD(6)),FMT=*) MAXITR
      NN = NE * (NNELEM - 1) + 1
      NDOF = NN * NBBLSZ
      NEDOF = NNELEM * NBBLSZ
      DO 301 I=1,NE
         IMAT(I) = 1
         DO 302 J=1,NNELEM
            ICON(I,J) = (I-1) * (NNELEM-1) + J
 302     CONTINUE
 301  CONTINUE
      INPRP = NE
      NMAT = 1
      GOTO 1111

*-------------------------------------------------+
*  Read in connectivity for local nodes           |
*   Four numbers required                         |
```

```
*   Values in order of element number                |
*   ICON(i,j) - connectivity for local               |
*               node j of element i                  |
*-------------------------------------------------+
 5004 INCON = INCON+1
      GOTO 1111

*-------------------------------------------------+
*  Read in coordinates                             |
*   R(I) - R coordinate for global node i          |
*-------------------------------------------------+
 5005 INN = INN+1
      IF (NOFLDS.NE.1)
     + CALL ERROR('Incorrect number of coordinate fields on cards 500')
        READ(UNIT=BUFFER(1:IFLD(1)),FMT=*) R(INN)
        GOTO 1111

*-------------------------------------------------+
*  Read in prescribed displacement data            |
*   NPD(i) - node of prescribed concentration      |
*   IDIR(i) - bubble group of prescribed conc.     |
*   SPDISP(i) - value of prescribed concentration  |
*-------------------------------------------------+
 5006 INDSP = INDSP+1
      IF (NOFLDS.NE.3)
     +CALL ERROR('Incorrect number of fields on cards 600          ')
      READ(UNIT=BUFFER(1:IFLD(1)),FMT=*) NPD(INDSP)
 9596  FORMAT(' INDSP = ',I3,' NPD = ',I3)
 9595  FORMAT(' NPD = ',I3)
      READ(UNIT=BUFFER(IFLD(1)+1:IFLD(2)),FMT=*) IDIR(INDSP)
      READ(UNIT=BUFFER(IFLD(2)+1:IFLD(3)),FMT=*) SPDISP(INDSP)
      GOTO 1111

*-------------------------------------------------+
*  Read in                                         |
*   GRNDIA - grain diameter                        |
*-------------------------------------------------+
 5007 IF (NOFLDS.NE.1)
     +CALL ERROR('Incorrect number of fields on cards 700          ')
      READ(UNIT=BUFFER(1:IFLD(1)),FMT=*) GRNDIA
      GOTO 1111

*-------------------------------------------------+
*  Read in                                         |
*   IMAT(i) - type of material for element i       |
*-------------------------------------------------+
 5008 IF (NOFLDS.NE.NE)
     +CALL ERROR('Incorrect number of fields on cards 800          ')
       IMAT(1) = 1
      IF (NE.EQ.1) GOTO 77
      DO 77 I=2,NE
       IMAT(I) = 1
  77    CONTINUE
      INPRP = NE
      GOTO 1111

*-------------------------------------------------------------+
*  Read in initial porosity                                    |
*   POROS(i) - initial porosity for material i                 |
*-------------------------------------------------------------+
 5009    INMAT = INMAT+1
      IF (NOFLDS.NE.3)
     +CALL ERROR('Incorrect number of fields on cards 900          ')
      READ(UNIT=BUFFER(1:IFLD(1)),FMT=*) POROS(INMAT)
      READ(UNIT=BUFFER(IFLD(1)+1:IFLD(2)),FMT=*) ENRICH
      READ(UNIT=BUFFER(IFLD(2)+1:IFLD(3)),FMT=*) CONLIM
      GOTO 1111

*-------------------------------------------------------------+
*  Read in period characteristics                              |
*   DELTIM(i) - time step for period i                         |
*-------------------------------------------------------------+
 5010 INPRD = INPRD + 1
      IF (NOFLDS.NE.3)
     +CALL ERROR('Incorrect number of fields on cards 1000         ')
      READ(UNIT=BUFFER(1:IFLD(1)),FMT=*) TIMLEN
      READ(UNIT=BUFFER(IFLD(1)+1:IFLD(2)),FMT=*) NLINC(INPRD)
      READ(UNIT=BUFFER(IFLD(2)+1:IFLD(3)),FMT=*) LOUT(INPRD)
      IF (INPRD.EQ.1) THEN
         ENDTIM(INPRD) = TIMLEN
         ELSE
         ENDTIM(INPRD) = ENDTIM(INPRD-1) + TIMLEN
         ENDIF
      GOTO 1111

*-------------------------------------------------------------+
*  Read in for power history for materials                     |
*   POWER1(i,j) - initial power for material j and period i    |
*   TEMP1(i,j) - initial temp for material j and period i      |
*   PRESS1(i,j) - initial press for material j and period i    |
```

```
*   POWER2(i,j) - final power for material j and period i     |
*   TEMP2(i,j) - final temp for material j and period i       |
*   PRESS2(i,j) - final press for material j and period i     |
*-------------------------------------------------------------+
 5011 IF (INHMAT.EQ.NMAT) THEN
         INHMAT = 1
         IPRD = IPRD + 1
         ELSE
         INHMAT = INHMAT+1
         ENDIF
      IF (NOFLDS.NE.4.AND.NOFLDS.NE.8)
     +CALL ERROR('Incorrect number of fields on cards 1100          ')
      READ(UNIT=BUFFER(1:IFLD(1)),FMT=*) POWER1(IPRD,INHMAT)
      READ(UNIT=BUFFER(IFLD(1)+1:IFLD(2)),FMT=*) TEMP1(IPRD,INHMAT)
      READ(UNIT=BUFFER(IFLD(2)+1:IFLD(3)),FMT=*) TMPGR1(IPRD,INHMAT)
      READ(UNIT=BUFFER(IFLD(3)+1:IFLD(4)),FMT=*) PRESS1(IPRD,INHMAT)
      IF (NOFLDS.GT.4) THEN
      READ(UNIT=BUFFER(IFLD(4)+1:IFLD(5)),FMT=*) POWER2(IPRD,INHMAT)
      READ(UNIT=BUFFER(IFLD(5)+1:IFLD(6)),FMT=*) TEMP2(IPRD,INHMAT)
      READ(UNIT=BUFFER(IFLD(6)+1:IFLD(7)),FMT=*) TMPGR2(IPRD,INHMAT)
      READ(UNIT=BUFFER(IFLD(7)+1:IFLD(8)),FMT=*) PRESS2(IPRD,INHMAT)
      ENDIF
      GOTO 1111

*-------------------------------------------------------------+
*  Output                                                     |
*   NE    - total number of elements                          |
*   NN    - total number of nodes                             |
*   NLINC - number of load increments                         |
*   UNITS(IUNITS) - 1=British                                 |
*                   2=Metric                                  |
*   PROBLM(IPRBTP) - 1=Linear plane stress                    |
*                    2=Beam                                   |
*                    3=Axisymmetric                           |
*                    4=Nonlinear plane stress                 |
*-------------------------------------------------------------+
 1101 WRITE(8,2002) NE,NN,NBBLSZ,NMAT,NSPC,NPRD,MAXITR
     +             ,UNITS(0,IUNITS)
 2002 FORMAT('1'/'  Total number of elements                = ',I5/
     +         '  Total number of nodes                     = ',I5/
     +         '  Maximum bubble groups per node            = ',I5/
     +         '  Total number of material types            = ',I5/
     +         '  Total number of specified displacements   = ',I5/
     +         '  Total number of time periods              = ',I5/
     +         '  Maximum number of iterations              = ',I5/
     +         '  Units are in ',A7//)

*-------------------------------------------------------------+
* Output                                                      |
*   IGP - Number of Guass points                              |
*-------------------------------------------------------------+
*     IF (IPRBTP.EQ.3)
*    +   WRITE(8,2222) IGP
         WRITE(8,2222) IGP
 2222 FORMAT('  Number of Guass integration points is ',I2)

*-------------------------------------------------------------+
*  Output element property data with reference                |
*   to material type                                          |
*-------------------------------------------------------------+
      WRITE(8,2032) GRNDIA
 2032 FORMAT(//'  Grain Diameter  = ',1PG10.3E1,' microns'//)

*-------------------------------------------------------------+
*  Output nodal coordinates.                                  |
*-------------------------------------------------------------+
      IPRBTP = 1

      WRITE(8,2011) CRDSYS(1,IPRBTP),
     +              CRDUNT(1,IUNITS,IPRBTP)
 2011 FORMAT('1'//26X,'NODAL POSITIONS FROM CENTER'//
     +           26X,'                    ',A8/
     +           26X,'Global Node>I       (',A6,')'/
     +         '+',25X,'_____________     ________')

*-------------------------------------------------------------+
*  Check for correct amount of data                           |
*-------------------------------------------------------------+
      IF (INN.NE.NN)
     +CALL ERROR('Incorrect number of coordinate sets entered.       ')

      DO 115 I=1,NN
         WRITE(8,2010) I,R(I)
 2010    FORMAT(27X,I5,6X,4X,1PG12.6E1)
 115     CONTINUE

*-------------------------------------------------------------+
*  Output connectivity data                                   |
*   Set                                                       |
*   NNETYP(i) - number of nodes for type i element            |
```

```
*   NDOF - total number of dofs                          |
*-----------------------------------------------------+

      WRITE(8,2021)
 2021 FORMAT('1'//30X,'ELEMENT CONNECTIVITY'//
     +    17X,'            |---- Global Node Number ----|'/
     +    17X,'            |---- for the Local Node ----|'/
     +    17X,'            |---- Numbers Below ---------|'/
     +    17X,'  Element     1       2       3       4  '/
     +'+',16X,' _________  ______  ______  ______  ______')

      NDOF = NBBLSZ*NN

*-----------------------------------------------------+
*  NDOFE(i) - num of dofs for element i                |
*-----------------------------------------------------+
      DO 125 I=1,NE
         WRITE(8,2020) I,(ICON(I,J),J=1,4)
 2020    FORMAT(17X,I5,3X,4(3X,I5))
 125     CONTINUE

*-----------------------------------------------------+
*  Output element property data with reference         |
*   to material type                                   |
*-----------------------------------------------------+
      WRITE(8,4421)
 4421 FORMAT('1',27X,'ELEMENT MATERIAL PROPERTIES'//
     +    25X,'           Material  Initial '/
     +    25X,'Element>I  Type ID   Porosity'/
     +'+',24X,'_________  ______   ________')

*-----------------------------------------------------+
*  Check for correct amount of data                    |
*-----------------------------------------------------+
      IF (INPRP.NE.NE)
     +CALL ERROR('Incorrect number of element property sets entered.')

      DO 425 I=1,NE
         WRITE(8,5420) I,IMAT(I),POROS(IMAT(I))
 5420    FORMAT(25X,I5,3X,I5,8X,1PG10.4E1)
 425     CONTINUE

*-----------------------------------------------------+
*  Output material properties                          |
*-----------------------------------------------------+

      WRITE(8,2031)
 2031 FORMAT('1',29X,'TIME HISTORY CONDITIONS'//
     +' ','Time    Initial    Initial    Initial ',
     +          '   Final      Final      Final '/
     +' ','Incr     Power      Temp       Press  ',
     +          '   Power      Temp       Press '/
     +'+','____ _________ _________ _________ ',
     +          '_________ _________ _________')

*-----------------------------------------------------+
*  Check for correct amount of data                    |
*-----------------------------------------------------+
      IF (INMAT.NE.NMAT)
     +CALL ERROR('Incorrect number of material property sets entered. ')

      DO 136 I=1,NPRD
         DO 135 J=1,NMAT
            WRITE(8,2030) I,POWER1(I,J),TEMP1(I,J),PRESS1(I,J),
     +                      POWER2(I,J),TEMP2(I,J),PRESS2(I,J)
 2030       FORMAT(' ',I3,2X,6(1X,1PG10.3E1))
 135        CONTINUE
 136     CONTINUE

*-----------------------------------------------------+
*  Output material properties                          |
*-----------------------------------------------------+

      WRITE(8,2071)
 2071 FORMAT('1',24X,'TIME STEP INFORMATION'//
     +' ',16X,'Time    Length    Number of   Time Step   Printout'/
     +' ',16X,'Incr   (seconds)  Time Steps  (seconds)   Interval'/
     +'+',15X,'____ _________ _________ _________ _______')

*-----------------------------------------------------+
*  Check for correct amount of data                    |
*-----------------------------------------------------+
      IF (INMAT.NE.NMAT)
     +CALL ERROR('Incorrect number of material property sets entered. ')

      ENDTIM(0) = 0.0Q0
      DO 166 I=1,NPRD
         DELTIM(I) = (ENDTIM(I)-ENDTIM(I-1))/REAL(NLINC(I))
         WRITE(8,2230) I,ENDTIM(I),NLINC(I),DELTIM(I),LOUT(I)
 2230    FORMAT(' ',15X,I3,5X,1PG10.3,2X,I6,6X,1PG10.3,4X,I3)
```

```
 166     CONTINUE

*-------------------------------------------------------+
*  Output specified displacements.                      |
*-------------------------------------------------------+

      IF (NSPC.NE.0) THEN
      WRITE(8,2051)
 2051 FORMAT('1'//28X,'SPECIFIED CONCENTRATIONS'//
     +    18X,'Global Node #    Bubble Gp       Value'/
     +'+',17X,'_____________    _________    _________')

*-------------------------------------------------------+
*  Check for correct amount of data                     |
*-------------------------------------------------------+
      IF (INDSP.NE.NSPC)
     +CALL ERROR('Incorrect number of displacements entered.        ')

      DO 155 I=1,NSPC
         WRITE(8,2050) NPD(I),IDIR(I),SPDISP(I)
 2050    FORMAT(' ',17X,I7,10X,I5,8X,1PG10.3)
 155     CONTINUE

       ENDIF

      RETURN
      END
*=======================================================+
*                                                       |
*      IIIIIIII  NN    NN  TTTTTTTT   AAAAAA   RRRRRRR    SSSSSS      |
*      IIIIIIII  NNN   NN  TTTTTTTT  AAAAAAAA  RRRRRRRR  SSSSSSSS     |
*         II     NNNN  NN     TT     AA    AA  RR    RR  SS           |
*         II     NN NN NN     TT     AA    AA  RR    RR  SSSSSSS      |
*         II     NN  NNNN     TT     AAAAAAAA  RRRRRRRR   SSSSSSS     |
*         II     NN   NNN     TT     AAAAAAAA  RRRRRR          SS     |
*      IIIIIIII  NN    NN     TT     AA    AA  RR   RR   SSSSSSSS     |
*      IIIIIIII  NN    NN     TT     AA    AA  RR    RR   SSSSSS      |
*                                                       |
*     This subroutine creates the J array and L array   |
*                                                       |
*    Inputs are:                                        |
*       NE - total number of elements                   |
*       NDOF - total number of degrees of freedom       |
*       NPD - total number of prescribed degrees of freedom
*       NDOFE(i) - number of degrees of freedom for element i
*       ICONDF(i,j) - connectivity array of local dofs to global dofs
*                     i.e. global node number of local node J
*                     for element i
*       IDOFPD(i) - global degree of freedom number of ith
*                   prescribed displacement
*       UP(i) -  prescribed displacement of global node i
*                                                       |
*=======================================================+

      SUBROUTINE INTARS

*-------------------------------------------------------------+
*  Get declarations                                           |
*-------------------------------------------------------------+
      INCLUDE (MAIN)

*-------------------------------------------------------------+
*  Here we create the J array which contains indices of global |
*   degrees of freedom which are not prescribed.              |
*   Size = Total dof - Prescribed dof                         |
*                                                             |
*   NDOF - total global degrees of freedom                    |
*   NPD - number of prescribed degrees of freedom             |
*   IDOFPD(i) - location in global dof of ith prescribed dof  |
*   JARRAY(i) - values of J array                             |
*   INONPD - total number of non-prescribed dofs              |
*-------------------------------------------------------------+

      ICOUNT = 0
      DO 101 I=1,NDOF
         DO 102 J=1,NSPC
            IF (I.EQ.IDOFPD(J)) GOTO 101
 102        CONTINUE
         ICOUNT = ICOUNT+1
         JARRAY(ICOUNT) = I
 101     CONTINUE
      INONPD = ICOUNT

*-------------------------------------------------------------+
*  Create the L array.  This array is basically the inverse of the
*   J array with its values indicating what position a dof occupies
*   among all the NON-PRESCRIBED dofs.  Size = total dofs.  Zeros
*   as values indicate that dof is prescribed.                |
*                                                             |
*   LARRAY(i) - values of the L array                         |
```

```
*-------------------------------------------------------------------+

      DO 103 I=1,NDOF
         LARRAY(I) = 0
  103    CONTINUE
      DO 104 I=1,INONPD
         IST = JARRAY(I)
         LARRAY(IST) = I
  104    CONTINUE
       RETURN
       END
*===================================================================+
*                                                                   |
*      MMMMMM   TTTTTTTT  XX    XX   AAAAAA   DDDDDDD   DDDDDDD    |
*      MMMMMMM  TTTTTTTT  XX    XX  AAAAAAAA  DDDDDDDD  DDDDDDDD   |
*      MM MM MM    TT      XX  XX   AA    AA  DD    DD  DD    DD   |
*      MM MM MM    TT       XXXX    AA    AA  DD    DD  DD    DD   |
*      MM MM MM    TT       XXXX    AAAAAAAA  DD    DD  DD    DD   |
*      MM MM MM    TT      XX  XX   AAAAAAAA  DD    DD  DD    DD   |
*      MM MM MM    TT     XX    XX  AA    AA  DDDDDDDD  DDDDDDDD   |
*      MM MM MM    TT     XX    XX  AA    AA  DDDDDDD   DDDDDDD    |
*                                                                   |
*   This subroutine takes a matrix A and B and adds them            |
*     to get C.  A and B must be the same size. Input parameters are:|
*                                                                   |
*     A - the array containing the A matrix                         |
*     B - the array containing the B matrix                         |
*     IROW - the number of rows                                     |
*     ICOL - the number of columns                                  |
*     IARDIM - the actual row dimension of A in storage             |
*     IACDIM - the actual column dimension of A in storage          |
*     IBRDIM - the actual row dimension of B in storage             |
*     IBCDIM - the actual column dimension of B in storage          |
*     AFLAG - .TRUE. if dimensioned for num of elements             |
*     BFLAG - .TRUE. if dimensioned for num of elements             |
*                                                                   |
*===================================================================+
      SUBROUTINE MTXADD(A,B,IROW,ICOL,IELEM
     +                 ,IARDIM,IACDIM,IADDIM,IBRDIM,IBCDIM,IBDDIM
     +                 ,AFLAG,BFLAG)

      IMPLICIT REAL*16 (A-H,O-Z)
      IMPLICIT INTEGER (I-N)

      LOGICAL AFLAG,BFLAG

      DIMENSION A(IARDIM,IACDIM,IADDIM)
      DIMENSION B(IBRDIM,IBCDIM,IBDDIM)

      IAELEM = 1
      IBELEM = 1

      IF (AFLAG) IAELEM = IELEM
      IF (BFLAG) IBELEM = IELEM

      DO 101 I=1,IROW
         DO 102 J=1,ICOL
            B(I,J,IBELEM) = A(I,J,IAELEM)+B(I,J,IBELEM)
  102       CONTINUE
  101    CONTINUE

       RETURN
       END
*===================================================================+
*                                                                   |
*       000000   UU    UU  TTTTTTTT  PPPPPPP   UU    UU  TTTTTTTT  |
*      00000000  UU    UU  TTTTTTTT  PPPPPPPP  UU    UU  TTTTTTTT  |
*      00    00  UU    UU     TT     PP    PP  UU    UU     TT     |
*      00    00  UU    UU     TT     PP    PP  UU    UU     TT     |
*      00    00  UU    UU     TT     PPPPPPPP  UU    UU     TT     |
*      00    00  UU    UU     TT     PPPPPPP   UU    UU     TT     |
*      00000000  UUUUUUUU     TT     PP        UUUUUUUU     TT     |
*       000000    UUUUUU      TT     PP         UUUUUU      TT     |
*                                                                   |
*   Prints the nodal displacements and elemental stresses, strains  |
*                                                                   |
*===================================================================+
      SUBROUTINE OUTPUT(LINC,IPRD,NLINC,ITEND,ITERAT,IUNITS,
     +                  PCTREL,PCTSWL)

*-------------------------------------------------------------------+
*  Get declarations                                                 |
*-------------------------------------------------------------------+
      INCLUDE (MAIN)
      INCLUDE (INOUT)

*-------------------------------------------------------------------+
* Calculate percent of load                                         |
*-------------------------------------------------------------------+
       TIMINC = QFLOAT(LINC)*DELTIM(IPRD)
       TIMPRD = ENDTIM(IPRD-1) + TIMINC
```

```
      IPRCNT = 100*LINC/NLINC(IPRD)

*-----------------------------------------------------+
* Output how many iterations and percent load         |
*-----------------------------------------------------+
      IF (ITEND) WRITE(8,8888) IPRD,LINC,DELTIM(IPRD),TIMINC,TIMPRD,
     +                         IPRCNT,PCTREL,PCTSWL
     +                        ,BURATE,BUAPPR,BUAPTL,BUMWTL
      IF (ITEND) WRITE(9,6888) IPRD,LINC,DELTIM(IPRD),TIMINC,TIMPRD,
     +                         IPRCNT,PCTREL,PCTSWL
     +                        ,BURATE,BUAPPR,BUAPTL,BUMWTL
 6888 FORMAT(I3,1X,I3,1X,1PG10.4,1X,1PG10.4,1X,1PG10.4,1X,I3,1X/
     +       1PG9.3,1X,1PG9.3,1X,1PG9.3,1X,1PG9.3,1X,1PG9.3,1X,0PF9.2)
 8888 FORMAT('1'//
     +  16X,'TIME PERIOD ',I3,/
     +  16X,'  Time Step ',I3//
     +  16X,'  Time Step Size = ',1PG10.4E1,' seconds'/
     +  16X,'  Time in Period = ',1PG10.4E1,' seconds'/
     +  16X,'  New Time       = ',1PG10.4E1,' seconds',0P//
     +  16X,'Time step represents ',I3,'% of full time period.'//
     +  16X,'Percent fission gas released = ',1PG10.4,'%'/
     +  16X,'Percent swelling             = ',1PG10.4,'%'///
     +  16X,'Burnup rate       = ',1PG9.3,' atom percent/day'/
     +  16X,'Burnup for period = ',1PG9.3,' atom percent'/
     +  16X,'Burnup (total)    = ',1PG9.3,' atom percent',0P/
     +  16X,'                  = ',F9.2,' MWD/MT'//)

      IF (ITEND) WRITE(8,8887) POWER(1),PRESS(1),TEMP(1)
 8887 FORMAT(' '//16X,'Power density = ',G10.4,' W/cm3'/
     +            16X,'Pressure      = ',G10.4,' Pa'/
     +            16X,'Temperature   = ',G10.4,' K')
      IF (ITEND) WRITE(9,6887) POWER(1),PRESS(1),TEMP(1)
 6887 FORMAT(3(G10.4,1X))

*-----------------------------------------------------+
*  Write out displacements                            |
*-----------------------------------------------------+

      WRITE(8,100)
 100  FORMAT(' '//29X,'NODAL CONCENTRATIONS'//)

      WRITE(8,101)
 101  FORMAT( 13X,
     + 'Bubble   Concentration   Bubble Radius   Bubble Pressure'/
     +       13X,
     + 'Group       (cm-3)          (cm)           (Pa)'/
     +       '+',12X,
     + '_____   ____________   ____________   _____________'/)


      DO 1 I=1,NN
         WRITE(8,7) I,R(I)
 7       FORMAT(/13X,'--------- Node = ',I2,' ----- ',
     +             'Radius = ',1PG10.3,'----------'/)
         DO 2 J=1,NBBLSZ
            IND = (I-1)*NBBLSZ + J
            WRITE(8,6) J,CONC(IND),RADIUS(I,J),BBLPRS(I,J)
 6          FORMAT(14X,I4,3(3X,1PG13.6))
            WRITE(9,60) I,R(I),J,CONC(IND),RADIUS(I,J),BBLPRS(I,J)
 60         FORMAT(I4,1X,E13.6,1X,I4,1X,3(E13.6,1X))
 2          CONTINUE
 1       CONTINUE

       RETURN
       END
*=====================================================+
*                                                     |
*              PPPPPPP   RRRRRRR    000000   PPPPPPP  |
*              PPPPPPPP  RRRRRRRR  00000000  PPPPPPPP |
*              PP    PP  RR    RR  00    00  PP    PP |
*              PP    PP  RR    RR  00    00  PP    PP |
*              PPPPPPPP  RRRRRRRR  00    00  PPPPPPPP |
*              PPPPPPP   RRRRRR    00    00  PPPPPPP  |
*              PP        RR   RR   00000000  PP       |
*              PP        RR    RR   000000   PP       |
*                                                     |
*   This subroutine sets the properties for all the elements in this |
*    subperiod.                                       |
*                                                     |
*=====================================================+
      SUBROUTINE PROP(IPRD,LINC,NLINC,ITER)

      INCLUDE (MAIN)
      INCLUDE (PI)
      INCLUDE (DIFFCOEF)
      INCLUDE (LATTICE)
      INCLUDE (BOLTZMAN)
      INCLUDE (SURFENGY)
      INCLUDE (VANDWAAL)
```

```
*-----------------------------------------------------------------+
*  Calculate sink strengths of dislocations for vacancies         |
*    and interstitials                                            |
*-----------------------------------------------------------------+

         ZVAC = (2.Q0*PI*DISDEN)/(QLOG(1.Q0/(QSQRT(PI*DISDEN)*DISRAD)))
         ZINT = 1.02Q0*ZVAC
         DO 101 IMTL=1,NMAT

            IF (ITER.EQ.0) THEN

*-----------------------------------------------------------------+
*  Calculate scaled power, temp, temp gradient, pressure          |
*-----------------------------------------------------------------+

            POWER(IMTL) = POWER1(IPRD,IMTL)
     +                   +(POWER2(IPRD,IMTL)-POWER1(IPRD,IMTL))
     +                   *QFLOAT(LINC)/QFLOAT(NLINC(IPRD))
            TEMP(IMTL) = TEMP1(IPRD,IMTL)
     +                   +(TEMP2(IPRD,IMTL)-TEMP1(IPRD,IMTL))
     +                   *QFLOAT(LINC)/QFLOAT(NLINC(IPRD))
            TMPGR(IMTL) = TMPGR1(IPRD,IMTL)
     +                   +(TMPGR2(IPRD,IMTL)-TMPGR1(IPRD,IMTL))
     +                   *QFLOAT(LINC)/QFLOAT(NLINC(IPRD))
            PRESS(IMTL) = PRESS1(IPRD,IMTL)
     +                   +(PRESS2(IPRD,IMTL)-PRESS1(IPRD,IMTL))
     +                   *QFLOAT(LINC)/QFLOAT(NLINC(IPRD))

            LTPARM = LTPRMA + LTPRMB*TEMP(1) + LTPRMC*(TEMP(1)**2)
            OMEGA = (LTPARM**3)*.25Q0
            LAMBDA = LTPARM*0.6299605250

*-----------------------------------------------------------------+
*  Calculate equilibrium vacancy and interstitial concs           |
*-----------------------------------------------------------------+

            PRSNGN(IMTL) = 1.Q0
            CONCVE(IMTL,1) = 0.Q0
            CONCVE(IMTL,2) = QEXP(-EFORMV/TEMP(1))/OMEGA

*-----------------------------------------------------------------+
*  Calculate fission rate and                                     |
*   associated fission gas and vacancy production rates.          |
*-----------------------------------------------------------------+

            FISSRC(IMTL) = POWER(IMTL) * 3.1211Q10
            ATMSRC(IMTL) = FISSRC(IMTL) * 0.25Q0
            VACSRC(IMTL) = FISSRC(IMTL) * 5.Q3
            YPRS = 1.Q5*(YPSLPE*TEMP(IMTL)+YPINT)

*-----------------------------------------------------------------+
*  Calculate resolution parameter and resolution adjustment       |
*   for large bubbles                                             |
*-----------------------------------------------------------------+

            RESOLP = ((TEMP(IMTL)-500.Q0)/1000.Q0
     +               *(RESLML-RESLMH)+RESLML)
            RESOLP = QMAX1(RESOLP,RESLML)
            RESOLP = QMIN1(RESOLP,RESLMH)
            RESOLP = RESOLP*FISSRC(IMTL)
            RESADJ = 15.Q-8/VANDWL(2)*(BLTZMN*TEMP(IMTL)/(2.Q0*SRFENG))
            ENDIF

            CONCV(NN) = CONCVE(IMTL,2)
            CONCI(NN) = CONCVE(IMTL,1)

            IF (IPRD.EQ.1.AND.LINC.EQ.1.AND.ITER.EQ.0.
     +                                  AND.IREST.NE.1) THEN
               DO 102 IBBL=1,NBBLSZ

*-----------------------------------------------------------------+
*  Set general initial bubble radii and vacancy concentrations    |
*-----------------------------------------------------------------+

                  IF (IBBL.EQ.1) THEN
                     TRAD = (1.5Q0*OMEGA/PI)**0.33333333
                     VACS = 2
                     ELSE
                     VACS = 2*QFLOAT(IBBL)
                     TRAD = (3.Q0*OMEGA*VACS/(4.Q0*PI))**0.33333333
                     ENDIF

*-----------------------------------------------------------------+
*  Do checking to make sure the above values are what is wanted   |
*-----------------------------------------------------------------+

                  DO 103 INODE=1,NN
                     INDEX = (INODE-1)*NBBLSZ+IBBL
                     CONC(INDEX) = 0.0Q0
                     DELCON(INDEX) = 0.0Q0
                     DELRAD(INDEX) = 0.0Q0
```

```
*-------------------------------------------------------------+
*  Set values for node on grain boundary (equilibrium values)  |
*-------------------------------------------------------------+
                  IF (INODE.EQ.NN.AND.IBBL.NE.1) THEN
                     RADMAX(IBBL) =
     +                  EQURAD(IBBL,TEMP(IMTL),PRESS(IMTL),TRAD)
                     RADMIN(IBBL) = (0.75Q0*QFLOAT(IBBL)/PI
     +                  *(BLTZMN*TEMP(1)/YPRS
     +                  +VANDWL(2)))**0.33333333333333330Q0
                     RADIUS(INODE,IBBL) = RADMAX(IBBL)
                     BBLPRS(INODE,IBBL) = 3.Q0*QFLOAT(IBBL)
     +                            *BLTZMN*TEMP(1)/
     +                            (4.Q0*PI*RADIUS(INODE,IBBL)**3 -
     +                             3.Q0*QFLOAT(IBBL)*VANDWL(2))
                     IF (BBLPRS(INODE,IBBL).GT.YPRS.OR.
     +                   BBLPRS(INODE,IBBL).LE.0.Q0) THEN
                        BBLPRS(INODE,IBBL) = YPRS
                        RADIUS(INODE,IBBL) = (0.75Q0*QFLOAT(IBBL)/PI
     +                     *(BLTZMN*TEMP(1)/BBLPRS(INODE,IBBL)
     +                     +VANDWL(2)))**0.33333333333333330Q0
                        ENDIF
                     ELSE

*-------------------------------------------------------------+
*  Set values for all interior nodes                          |
*-------------------------------------------------------------+
                     IF (IBBL.NE.1) THEN
                        BBLPRS(INODE,IBBL) = 3.Q0*QFLOAT(IBBL)
     +                              *BLTZMN*TEMP(1)/
     +                              (4.Q0*PI*TRAD**3 -
     +                               3.Q0*QFLOAT(IBBL)*VANDWL(2))
                        IF (BBLPRS(INODE,IBBL).GT.YPRS.OR.
     +                      BBLPRS(INODE,IBBL).LE.0.Q0) THEN
                           BBLPRS(INODE,IBBL) = YPRS
                           RADIUS(INODE,IBBL) = (0.75Q0*QFLOAT(IBBL)
     +                        /PI*(BLTZMN*TEMP(1)/BBLPRS(INODE,IBBL)
     +                        +VANDWL(2)))**0.33333333333333330Q0
                           ELSE
                           RADIUS(INODE,IBBL) = TRAD
                           ENDIF
                        ELSE
                        RADIUS(INODE,IBBL) = TRAD
                        RADMAX(IBBL) = TRAD
                        ENDIF
                     ENDIF
                     RAD(INDEX) = RADIUS(INODE,IBBL)

 103                 CONTINUE
 102              CONTINUE
                ENDIF

*-------------------------------------------------------------+
*  Calculate diffusion coefficients                           |
*-------------------------------------------------------------+

            CALL DIFFUS
 101      CONTINUE
        RETURN
        END
*=============================================================+
*                                                             |
*          RRRRRRR    AAAAAA   DDDDDDD   IIIIIIII  IIIIIIII  |
*          RRRRRRRR  AAAAAAAA  DDDDDDDD  IIIIIIII  IIIIIIII  |
*          RR    RR  AA    AA  DD    DD     II        II     |
*          RR    RR  AA    AA  DD    DD     II        II     |
*          RRRRRRRR  AAAAAAAA  DD    DD     II        II     |
*          RRRRRR    AAAAAAAA  DD    DD     II        II     |
*          RR   RR   AA    AA  DDDDDDDD  IIIIIIII  IIIIIIII  |
*          RR    RR  AA    AA  DDDDDDD   IIIIIIII  IIIIIIII  |
*                                                             |
*=============================================================+
      SUBROUTINE RADII

      INCLUDE (MAIN)
      INCLUDE (PI)
      INCLUDE (SURFENGY)
      INCLUDE (BOLTZMAN)
      INCLUDE (VANDWAAL)

      DO 10 J=2,NBBLSZ

         DO 20 I= 1,NN
            INDEX = (I-1)*NBBLSZ+J
            RADIUS(I,J) = RAD(INDEX)+DELRAD(INDEX)
            BBLPRS(I,J) = 3.Q0*QFLOAT(J)*BLTZMN*TEMP(1)/
     +                     (4.Q0*PI*RADIUS(I,J)**3 -
     +                      3.Q0*QFLOAT(J)*VANDWL(2))
            IF (BBLPRS(I,J).LT.0.Q0.OR.BBLPRS(I,J).GT.YPRS) THEN
               BBLPRS(I,J) = YPRS
```

```
            RADIUS(I,J) = (0.75Q0*QFLOAT(J)
     +                           /PI*(BLTZMN*TEMP(1)/BBLPRS(I,J)
     +                           +VANDWL(2)))**0.33333333333333333Q0
            ENDIF
 20       CONTINUE
 10     CONTINUE

      CALL DIFFUS

      RETURN
      END
*=============================================================+
*                                                             |
*      RRRRRRR   DDDDDDD    CCCCCC    AAAAAA   RRRRRRR   DDDDDDD      |
*      RRRRRRRR  DDDDDDDD  CCCCCCCC  AAAAAAAA  RRRRRRRR  DDDDDDDD     |
*      RR    RR  DD    DD  CC    CC  AA    AA  RR    RR  DD    DD     |
*      RR    RR  DD    DD  CC        AA    AA  RR    RR  DD    DD     |
*      RRRRRRRR  DD    DD  CC        AAAAAAAA  RRRRRRRR  DD    DD     |
*      RRRRRR    DD    DD  CC    CC  AAAAAAAA  RRRRRR    DD    DD     |
*      RR   RR   DDDDDDDD  CCCCCCCC  AA    AA  RR   RR   DDDDDDDD     |
*      RR    RR  DDDDDDD    CCCCCC   AA    AA  RR    RR  DDDDDDD      |
*                                                             |
*                                                             |
*     Reads in input deck and parses fields                   |
*     Checks for comment cards beginning with '*' or 'C'      |
*     Also outputs cards as read                              |
*=============================================================+
      SUBROUTINE RDCARD(A,ICRD,NOFLDS,IFLD,IEND,IOUT)

      IMPLICIT REAL*16 (A-H,O-Z)
      IMPLICIT INTEGER (I-N)
      CHARACTER*80 A,B
      DIMENSION IFLD(40)

      IEND = 0

 10   READ(5,'(A80)',END=900) B

      IF (IOUT.EQ.1) WRITE(8,3000) B
 3000 FORMAT(' ',A80)

      IF (B(1:1).EQ.'*'.OR.B(1:1).EQ.'C') GOTO 10

      LSTSPC = 0
      NOFLDS = 0

      DO 101 I=1,80

         IF (B(I:I).EQ.' '.OR.B(I:I).EQ.',') THEN
            B(I:I) = ' '
            IF (NOFLDS.EQ.0) THEN
               IF (I.EQ.80) GOTO 10
               IF (LSTSPC.EQ.0) GOTO 101
               READ(UNIT=B(1:(I-1)),FMT=*,ERR=999) ICRD
               NOFLDS = 1
               A = B(I:80-I+1)
               ISTOP = I-1
               LSTSPC = I
               GOTO 101
               ENDIF

            IF (LSTSPC.EQ.(I-1)) THEN
               LSTSPC = I
               GOTO 101
               ENDIF

            LSTSPC = I
            IFLD(NOFLDS) = I-1-ISTOP
            NOFLDS = NOFLDS+1
            ELSE
            IF (I.EQ.80.AND.LSTSPC.NE.79) THEN
               IFLD(NOFLDS) = 80-ISTOP
               NOFLDS = NOFLDS + 1
               ENDIF
            IF (I.EQ.1) LSTSPC = -1
            ENDIF
 101     CONTINUE
      NOFLDS = NOFLDS - 1

      RETURN
 900  IEND = 1
      RETURN
 999  CALL ERROR('An error occured durin read of the input file')
      STOP
      END
*=============================================================+
*                                                             |
*      RRRRRRR   EEEEEEEE   SSSSSS   IIIIIIII  NN    NN  RRRRRRR      |
*      RRRRRRRR  EEEEEEEE  SSSSSSSS  IIIIIIII  NNN   NN  RRRRRRRR     |
*      RR    RR  EE        SS           II     NNNN  NN  RR    RR     |
```

```
*      RR    RR  EEEEE     SSSSSSS      II     NN NN NN  RR    RR      |
*      RRRRRRRR  EEEEE      SSSSSSS     II     NN  NNNN  RRRRRRRR      |
*      RRRRRR    EE             SS     II     NN   NNN  RRRRRR        |
*      RR   RR   EEEEEEEE  SSSSSSSS  IIIIIIII  NN    NN  RR          |
*      RR    RR  EEEEEEEE   SSSSSS   IIIIIIII  NN    NN  RR          |
*                                                                      |
*     This subroutine inputs the restart file.                         |
*                                                                      |
*======================================================================+
      SUBROUTINE RESINP(DENOM1,DNOM2V,DNOM2I,DENOM3)

      INCLUDE (MAIN)

     OPEN(UNIT=11,STATUS='OLD',ACCESS='SEQUENTIAL',FORM='UNFORMATTED')

      READ(11) NET,NNELET,NBBLST
      PRINT *, ' NET - ',NET
      PRINT *, ' NNELET - ',NNELET
      PRINT *, ' NBBLST - ',NBBLST

      IF (NET.NE.NE.OR.NNELET.NE.NNELEM.OR.NBBLST.NE.NBBLSZ)
     +    CALL ERROR(' RESTART FILE DOES NOT MATCH INPUT FILE ')
      READ(11) RESATM
      READ(11) DENOM1,DNOM2V,DNOM2I,DENOM3
      DO 10 I = 1,NN
         IND = (I-1)*NBBLSZ
         READ(11) (CONC(J),J=IND+1,IND+NBBLSZ)
         PRINT *, ' CONC(',IND+2,') = ',CONC(IND+2)
         READ(11) (RAD(J),J=IND+1,IND+NBBLSZ)
         READ(11) (RADIUS(I,J),J=1,NBBLSZ)
         READ(11) (BBLPRS(I,J),J=1,NBBLSZ)
         READ(11) CONCV(I),CONCI(I)
  10     CONTINUE
      RETURN
      END
*======================================================================+
*                                                                      |
*      RRRRRRR   EEEEEEEE   SSSSSS    SSSSSS    AAAAAA   VV    VV     |
*      RRRRRRRR  EEEEEEEE  SSSSSSSS  SSSSSSSS  AAAAAAAA  VV    VV     |
*      RR    RR  EE        SS        SS        AA    AA  VV    VV     |
*      RR    RR  EEEEE     SSSSSSS   SSSSSSS   AA    AA  VV    VV     |
*      RRRRRRRR  EEEEE      SSSSSSS   SSSSSSS  AAAAAAAA  VV    VV     |
*      RRRRRR    FE             SS        SS  AAAAAAAA   VV  VV      |
*      RR   RR   EEEEEEEE  SSSSSSSS  SSSSSSSS  AA    AA    VVVV      |
*      RR    RR  EEEEEEEE   SSSSSS    SSSSSS   AA    AA     VV       |
*                                                                      |
*     This subroutine saves the restart file.                          |
*                                                                      |
*======================================================================+
      SUBROUTINE RESSAV(DENOM1,DNOM2V,DNOM2I,DENOM3)

      INCLUDE (MAIN)

      REWIND(10)

      WRITE(10) NE,NNELEM,NBBLSZ
      WRITE(10) TOTATM
      WRITE(10) DENOM1,DNOM2V,DNOM2I,DENOM3

      DO 10 I = 1,NN
         IND = (I-1)*NBBLSZ
         WRITE(10) (CONC(J),J=IND+1,IND+NBBLSZ)
         WRITE(10) (RAD(J),J=IND+1,IND+NBBLSZ)
         WRITE(10) (RADIUS(I,J),J=1,NBBLSZ)
         WRITE(10) (BBLPRS(I,J),J=1,NBBLSZ)
         WRITE(10) CONCV(I),CONCI(I)
  10     CONTINUE
      RETURN
      END
*======================================================================+
*                                                                      |
*          RRRRRRR   NN     NN   000000   RRRRRRR   MMMMMM             |
*          RRRRRRRR  NNN    NN  00000000  RRRRRRRR  MMMMMMM            |
*          RR    RR  NNNN   NN  00    00  RR    RR  MM MM MM           |
*          RR    RR  NN NN  NN  00    00  RR    RR  MM MM MM           |
*          RRRRRRRR  NN  NNNN  00    00  RRRRRRRR  MM MM MM           |
*          RRRRRR    NN   NNN  00    00  RRRRRR    MM MM MM           |
*          RR   RR   NN    NN  00000000  RR   RR   MM MM MM           |
*          RR    RR  NN    NN   000000   RR    RR  MM MM MM           |
*                                                                      |
*     Finds the norm of a vector.                                      |
*     Input is VECTOR - the array containing the vector                |
*              LENGTH - length of array (INTEGER)                      |
*======================================================================+

      REAL FUNCTION RNORM(VECTOR,LENGTH)

      REAL*16 VECTOR(1050)
      INTEGER LENGTH
```

```
      RNORM = 0.0Q0
      DO 10 I=1,LENGTH
         RNORM = RNORM + VECTOR(I)**2
   10    CONTINUE

      RETURN
      END
*=====================================================================+
*                                                                     |
*      SSSSSS   EEEEEEEE  MMMMMM   BBBBBBB   NN     NN  DDDDDDD     |
*     SSSSSSSS  EEEEEEEE  MMMMMMM  BBBBBBBB  NNN    NN  DDDDDDDD    |
*     SS        EE        MM MM MM  BB    BB  NNNN   NN  DD    DD    |
*     SSSSSSS   EEEEE     MM MM MM  BBBBBBB   NN NN NN  DD    DD    |
*      SSSSSSS  EEEEE     MM MM MM  BBBBBBB   NN  NNNN  DD    DD    |
*          SS   EE        MM MM MM  BB    BB  NN   NNN  DD    DD    |
*     SSSSSSSS  EEEEEEEE  MM MM MM  BBBBBBBB  NN    NN  DDDDDDDD    |
*      SSSSSS   EEEEEEEE  MM MM MM  BBBBBBB   NN    NN  DDDDDDD     |
*                                                                     |
*     This subroutine calculates the semi-bandwith                    |
*                                                                     |
*    Inputs are:                                                      |
*       NE - total number of elements                                 |
*       NDOF - total number of degrees of freedom                     |
*       NPD - total number of prescribed degrees of freedom           |
*       NDOFE(i) - number of degrees of freedom for element i         |
*       ICONDF(i,j) - connectivity array of local dofs to global dofs |
*                     i.e. global node number of local node J         |
*                     for element i                                   |
*       IDOFPD(i) - global degree of freedom number of ith            |
*                   prescribed displacement                           |
*       UP(i) -  prescribed displacement of global node i             |
*                (only set for nodes with prescribed displacements    |
*                                                                     |
*=====================================================================+

      SUBROUTINE SEMBND

*---------------------------------------------------------------------+
*  Get declarations                                                   |
*---------------------------------------------------------------------+
      INCLUDE (MAIN)

*---------------------------------------------------------------------+
*  Find the semi-bandwidth of the global K matrix.                    |
*   This is done by using the L array and known dofs per element.     |
*                                                                     |
*   NE - total number of elements                                     |
*   NDOFE(i) - number of dofs in ith element                          |
*   ICONDF(i,j) - connectivity array of jth local dof of ith element  |
*                  coverts local values to global values              |
*   IGNE,IGME - contains value of above array in loop                 |
*   LIGNE,LIGME - contains value of LARRAY(IGNE or IGME), i.e.        |
*                 value = 0 then don't include (prescribed dof)       |
*                 value <> 0 then find its position to compute H      |
*   MAXH - the semi-bandwidth                                         |
*---------------------------------------------------------------------+

      MAXH = 0
      NEDOF = NNELEM*NBBLSZ
      DO 105 IE=1,NE
         DO 106 M=1,NEDOF
            DO 107 N=1,NEDOF
               IGNE = ICONDF(IE,N)
               IGME = ICONDF(IE,M)
               LIGNE = LARRAY(IGNE)
               LIGME = LARRAY(IGME)

*-------------------------------------------------------------+
*  Do not include if either row or column number matches a    |
*   prescribed dof position                                   |
*-------------------------------------------------------------+

               IF (LIGME.EQ.0) GOTO 106
               IF (LIGNE.EQ.0) GOTO 107
               MTEST = LIGNE-LIGME+1
               MAXH = MAX(MTEST,MAXH)
  107          CONTINUE
  106       CONTINUE
  105    CONTINUE
      MAXH = NEDOF
      RETURN
      END
*=====================================================================+
*                                                                     |
*      SSSSSS   EEEEEEEE  TTTTTTTT   CCCCCC    000000   NN     NN    |
*     SSSSSSSS  EEEEEEEE  TTTTTTTT  CCCCCCCC  00000000  NNN    NN    |
*     SS        EE           TT     CC    CC  00    00  NNNN   NN    |
*     SSSSSSS   EEEEE        TT     CC        00    00  NN NN NN    |
*      SSSSSSS  EEEEE        TT     CC        00    00  NN  NNNN    |
*          SS   EE           TT     CC    CC  00    00  NN   NNN    |
```

```
*     SSSSSSSS  EEEEEEEE     TT     CCCCCCCC  00000000  NN    NN      |
*      SSSSSS   EEEEEEEE     TT      CCCCCC    000000   NN    NN      |
*                                                                     |
*    This subroutine sets the properties for all the elements in this |
*     subperiod.                                                      |
*                                                                     |
*=====================================================================+
      SUBROUTINE SETCON(IPRD,LINC,NLINC,ITYPE)

      INCLUDE (MAIN)
      INCLUDE (LATTICE)
      INCLUDE (PI)

      IF (ITYPE.EQ.1) THEN
         DO 101 I=1,NN*NBBLSZ
            CONC(I) = CONC(I) + DELCON(I)
            DELCON(I) = 0.0Q0
101         CONTINUE
         ENDIF

      IF (ITYPE.EQ.2) THEN
         DO 103 I=1,NN-1
            CONCV(I) = CONCV(I) + DELCNV(I)
            CONCI(I) = CONCI(I) + DELCNI(I)
            DELCNV(I) = 0.0Q0
            DELCNI(I) = 0.0Q0
103         CONTINUE
         ENDIF

      IF (ITYPE.EQ.3) THEN
         DO 104 I=1,NN-1
            DO 105 J=2,NBBLSZ
               IJ = (I-1)*NBBLSZ + J
               RAD(IJ) = RADIUS(I,J)
               DELRAD(IJ) = 0.0Q0
105            CONTINUE
104         CONTINUE
         ENDIF

      RETURN
      END
*=====================================================================+
*                                                                     |
*          SSSSSS   HH    HH   AAAAAA   PPPPPPP   EEEEEEEE           |
*         SSSSSSSS  HH    HH  AAAAAAAA  PPPPPPPP  EEEEEEEE           |
```

```
*          SS       HH    HH  AA    AA  PP    PP  EE                 |
*         SSSSSSS   HHHHHHHH  AA    AA  PP    PP  EEEEE              |
*          SSSSSSS  HHHHHHHH  AAAAAAAA  PPPPPPPP  EEEEE              |
*               SS  HH    HH  AAAAAAAA  PPPPPPP   EE                 |
*         SSSSSSSS  HH    HH  AA    AA  PP        EEEEEEEE           |
*          SSSSSS   HH    HH  AA    AA  PP        EEEEEEEE           |
*                                                                     |
*                                                                     |
*   This routine calculates the shape functions and derivitive        |
*    shape functions                                                  |
*=====================================================================+
      SUBROUTINE SHAPE(NNELEM,ITR,IGP,SHPFCT,DERSHP)

      INCLUDE (GAUSS)
*-----------------------------------------------------+
*  Zero shape factor and shape derivitive matricis    |
*-----------------------------------------------------+

      SHPFCT(1) = 0.0Q0
      SHPFCT(2) = 0.0Q0
      SHPFCT(3) = 0.0Q0
      SHPFCT(4) = 0.0Q0
      DERSHP(1) = 0.0Q0
      DERSHP(2) = 0.0Q0
      DERSHP(3) = 0.0Q0
      DERSHP(4) = 0.0Q0

*-------------------------------------------------------------------+
*  Calculate value of shape function at coordinates of integration  |
*   point                                                           |
*-------------------------------------------------------------------+
         IF (NNELEM.EQ.2) THEN
            SHPFCT(1) = 0.5Q0*(1.Q0-RNAT(ITR,IGP))
            SHPFCT(2) = 0.5Q0*(1.Q0+RNAT(ITR,IGP))
            ENDIF
         IF (NNELEM.EQ.3) THEN
            SHPFCT(2) = 1.Q0-RNAT(ITR,IGP)**2
            SHPFCT(1) = 0.5Q0*(1.Q0-RNAT(ITR,IGP)) - 0.5Q0*SHPFCT(2)
            SHPFCT(3) = 0.5Q0*(1.Q0+RNAT(ITR,IGP)) - 0.5Q0*SHPFCT(2)
            ENDIF
         IF (NNELEM.EQ.4) THEN
            SHPFCT(2) = 1.Q0-RNAT(ITR,IGP)**2
            SHPFCT(1) = 0.5Q0*(1.Q0-RNAT(ITR,IGP)) - 0.5Q0*SHPFCT(2)
            SHPFCT(4) = 0.5Q0*(1.Q0+RNAT(ITR,IGP)) - 0.5Q0*SHPFCT(2)
            SHPFCT(3) = -1.6875Q0 * RNAT(ITR,IGP)**3
```

```
     +                -0.5625Q0 * RNAT(ITR,IGP)**2
     +                +1.6875Q0 * RNAT(ITR,IGP)
     +                +0.5625Q0
           SHPFCT(1) = SHPFCT(1) -0.5625Q0 * RNAT(ITR,IGP)**3
     +                           +0.0625Q0 * RNAT(ITR,IGP)**2
     +                           +0.5625Q0 * RNAT(ITR,IGP)
     +                           -0.0625Q0
           SHPFCT(4) = SHPFCT(4) +0.5625Q0 * RNAT(ITR,IGP)**3
     +                           +0.0625Q0 * RNAT(ITR,IGP)**2
     +                           -0.5625Q0 * RNAT(ITR,IGP)
     +                           -0.0625Q0
           SHPFCT(2) = SHPFCT(2) +1.6875Q0 * RNAT(ITR,IGP)**3
     +                           +0.4375Q0 * RNAT(ITR,IGP)**2
     +                           -1.6875Q0 * RNAT(ITR,IGP)
     +                           -0.4375Q0
           ENDIF


        IF (NNELEM.EQ.2) THEN
           DERSHP(1) = -0.5Q0
           DERSHP(2) =  0.5Q0
           ENDIF
        IF (NNELEM.EQ.3) THEN
           DERSHP(2) =  -2.Q0 * RNAT(ITR,IGP)
           DERSHP(1) = -0.5Q0 - 0.5Q0*DERSHP(2)
           DERSHP(3) = 0.5Q0 - 0.5Q0*DERSHP(2)
           ENDIF
        IF (NNELEM.EQ.4) THEN
           DERSHP(2) =  -2.Q0 * RNAT(ITR,IGP)
           DERSHP(1) = -0.5Q0 - 0.5Q0*DERSHP(2)
           DERSHP(4) = 0.5Q0 - 0.5Q0*DERSHP(2)
           DERSHP(3) = -5.0625Q0 * RNAT(ITR,IGP)**2
     +                 -1.125Q0  * RNAT(ITR,IGP) + 1.6875Q0
           DERSHP(1) = DERSHP(1) -1.6875Q0 * RNAT(ITR,IGP)**2
     +                           +0.125Q0  * RNAT(ITR,IGP) + 0.5625Q0
           DERSHP(4) = DERSHP(4) +1.6875Q0 * RNAT(ITR,IGP)**2
     +                           +0.125Q0  * RNAT(ITR,IGP) - 0.5625Q0
           DERSHP(2) = DERSHP(2) +5.0625Q0 * RNAT(ITR,IGP)**2
     +                           +0.875Q0  * RNAT(ITR,IGP) - 1.6875Q0
           ENDIF
        RETURN
        END
*=============================================================+
*                                                             |
*          SSSSSS    000000   LL        VV    VV  EEEEEEEE   |
*         SSSSSSSS  00000000  LL        VV    VV  EEEEEEEE   |
*         SS        00    00  LL        VV    VV  EE         |
*         SSSSSSS   00    00  LL        VV    VV  EEEEE      |
*          SSSSSSS  00    00  LL        VV    VV  EEEEE      |
*               SS  00    00  LL         VV  VV   EE         |
*         SSSSSSSS  00000000  LLLLLLLL    VVVV    EEEEEEEE   |
*          SSSSSS    000000   LLLLLLLL     VV     EEEEEEEE   |
*                                                             |
*  This program will take a nonsymmetric banded matrix        |
*   and a right hand side vector                              |
*   and solve the system of equations for the unknowns.       |
*   In other words, solve for <A><X>=<R> where <A> and <R> are known
*                                                             |
*  The matrix is assumed to be stored in banded form using the|
*   whole bandwidth.  The original matrix is overwritten by   |
*   the separate lower and upper triangular parts.            |
*   The calculated answers replace the load vector.           |
*                                                             |
*  Solution is obtained using Guassian elimination and back   |
*   substitution.  Reference algorithms can be found in       |
*                                                             |
*    THE FINITE ELEMENT METHOD by                             |
*     O.C. Zienkiewicz                                        |
*                                                             |
*     NOTE:  No checking is done for positive definiteness    |
*            or zeros on the diagonal!                        |
*=============================================================+

      SUBROUTINE SOLVE (MAXHF,NONPDF)

*-------------------------------------------------------------+
*  Get declarations                                           |
*-------------------------------------------------------------+
      INCLUDE (MAIN)

      REAL*16 Y(1050)

*---------------------------------------+
*  Calculate full bandwidth.            |
*---------------------------------------+

 1    MAXBW = 2*MAXHF-1

*-----------------------------------------------+
*  Perform triangularization of stiffness matrix. |
```

```
*-----------------------------------------------+

      DO 10 J=2,NONPDF
         INCRJ = J-MAXHF
         INDEX1 = 1-INCRJ

         DO 20 I=1,J
            INCRI = I-MAXHF
            SUM1 = 0.Q0
            SUM2 = 0.Q0

            DO 30 M=1,I-1
               INCRM = M-MAXHF
               INDEX1 = M-INCRJ
               INDEX2 = I-INCRM
               INDEX3 = M-INCRI
               INDEX4 = J-INCRM

               IF (INDEX1.GT.0.AND.INDEX1.LE.MAXBW.AND.
     +              INDEX2.GT.0.AND.INDEX2.LE.MAXBW)
     +            SUM1 = SUM1+RKBC(J,INDEX1)*RKBC(M,INDEX2)

               IF (INDEX3.GT.0.AND.INDEX3.LE.MAXBW.AND.
     +              INDEX4.GT.0.AND.INDEX4.LE.MAXBW)
     +            SUM2 = SUM2+RKBC(I,INDEX3)*RKBC(M,INDEX4)

  30           CONTINUE
            INDEX1 = I-INCRJ
            INDEX2 = J-INCRI

            IF (I.NE.J.AND.INDEX1.GT.0.AND.INDEX1.LE.MAXBW) THEN
               RKBC(J,INDEX1) = (RKBC(J,INDEX1)-SUM1)/RKBC(I,MAXHF)
               ENDIF

            IF (INDEX2.GT.0.AND.INDEX2.LE.MAXBW)
     +         RKBC(I,INDEX2) = RKBC(I,INDEX2) - SUM2

  20        CONTINUE
  10     CONTINUE

*-----------------------------------------------+
*  Perform forward elimination.                 |
*-----------------------------------------------+

      Y(1) = RRBC(1)

      DO 40 I=2,NONPDF
         SUM1 = 0.Q0
         INCRI = I-MAXHF

         DO 50 J=MAX(1,INCRI),I-1
            IF (J.GT.INCRI) SUM1 = SUM1 + RKBC(I,J-INCRI)*Y(J)
  50        CONTINUE
         Y(I) = RRBC(I) - SUM1

  40     CONTINUE

*-----------------------------------------------+
*  Perform backward substitution.               |
*-----------------------------------------------+

      RRBC(NONPDF) = Y(NONPDF)/RKBC(NONPDF,MAXHF)

      DO 60 I=NONPDF-1,1,-1
         SUM1 = 0.Q0
         INCRI = I-MAXHF

         DO 70 J=I+1,MIN(NONPDF,INCRI+2*MAXHF-1)
            SUM1 = SUM1 + RKBC(I,J-INCRI)*RRBC(J)

  70        CONTINUE
         RRBC(I) = (Y(I) - SUM1)/RKBC(I,MAXHF)
  60     CONTINUE

      RETURN
      END
*===============================================================+
*                                                               |
*          SSSSSS   WW    WW  EEEEEEEE  LL        LL           |
*         SSSSSSSS  WW    WW  EEEEEEEE  LL        LL           |
*         SS        WW    WW  EE        LL        LL           |
*         SSSSSSS   WW WW WW  EEEEE     LL        LL           |
*          SSSSSSS  WW WW WW  EEEEE     LL        LL           |
*               SS  WW WW WW  EE        LL        LL           |
*         SSSSSSSS  WWWWWWWW  EEEEEEEE  LLLLLLLL  LLLLLLLL     |
*          SSSSSS    WW  WW   EEEEEEEE  LLLLLLLL  LLLLLLLL     |
*                                                               |
*                                                               |
*===============================================================+
      SUBROUTINE SWELL(IPRD,LINC,NLINC,PCTREL,PCTSWL,ATMREL)
```

```
      INCLUDE (MAIN)
      INCLUDE (ELKF)
      INCLUDE (GAUSS)
      INCLUDE (PI)
      INCLUDE (LATTICE)
      INCLUDE (BOLTZMAN)
      INCLUDE (SURFENGY)

      ATOMS = 0.Q0
      VOLUME = 0.Q0
      GRNVOL = 4.Q0*PI*GRNDIA**3/3.Q0
      DO 151 N=1,NBBLSZ
         ATOM(N) = 0.Q0
         VOLCUM(N) = 0.Q0
         VELCUM(N) = 0.Q0
 151     CONTINUE
      DO 100 IELEM=1,NE

*----------------------------+
*  Get global node numbers   |
*----------------------------+

      NODE1 = ICON(IELEM,1)
      NODE2 = ICON(IELEM,2)
      NODE3 = ICON(IELEM,3)

*-------------------------------------------------+
*  Zero shape factor and shape derivitive matricis |
*-------------------------------------------------+

      SHPFCT(1) = 0.0Q0
      SHPFCT(2) = 0.0Q0
      SHPFCT(3) = 0.0Q0

*--------------------------------+
*  Loop over integration points  |
*--------------------------------+
      OLDVOL = VOLUME
      DO 701 ITR=1,IGP

       CALL SHAPE(NNELEM,ITR,IGP,SHPFCT,DERSHP)

*-------------------------------------------------------+
*  Value of R in real coordinates of integration point  |
*-------------------------------------------------------+

        RGP = SHPFCT(1)*R(NODE1)+
     +        SHPFCT(2)*R(NODE2)+
     +        SHPFCT(3)*R(NODE3)

*-----------------------------------------+
*  Calculate Jacobian matrix and determinant  |
*-----------------------------------------+

      AJMAT = DERSHP(1) * R(ICON(IELEM,1))
     +      + DERSHP(2) * R(ICON(IELEM,2))
     +      + DERSHP(3) * R(ICON(IELEM,3))

      AJDET = AJMAT

*-----------------------------------------------------------+
*  Calculate numerical integration multipliers times R**2   |
*-----------------------------------------------------------+

      WMULT = 4.Q0*PI*WEIGHT(ITR,IGP)*RGP**2*AJDET

*-------------------------+
*  Loop over matrix rows  |
*-------------------------+

         DO 101 N=1,NBBLSZ
            NDOF1 = ICONDF(IELEM,N)
            NDOF2 = ICONDF(IELEM,N+NBBLSZ)
            NDOF3 = ICONDF(IELEM,N+2*NBBLSZ)

            CONCIP(N) = SHPFCT(1)*CONC(NDOF1)+
     +                  SHPFCT(2)*CONC(NDOF2)+
     +                  SHPFCT(3)*CONC(NDOF3)

            ATOM(N) = ATOM(N) + CONCIP(N)*QFLOAT(N)*WMULT

            RADIP(N) = SHPFCT(1)*RADIUS(NODE1,N) +
     +                 SHPFCT(2)*RADIUS(NODE2,N) +
     +                 SHPFCT(3)*RADIUS(NODE3,N)

            VELIP(N) = 3.Q0 / (BLTZMN*(TEMP(1)**2)*(LTPARM**3))
     +                 *DIFFS*TMPGR(1)*PI*(RADIP(N)**3)*SRFACT
            VOLCUM(N) = VOLCUM(N)
     +                  +4.Q0*PI*CONCIP(N)/3.Q0*RADIP(N)**3*WMULT
            VELCUM(N) = VELCUM(N) + VELIP(N)*WMULT
```

```
 101       CONTINUE

 701  CONTINUE

 100  CONTINUE

      DO 1032 IJ = 1,NBBLSZ
         ATOMS = ATOMS + ATOM(IJ)
1032     CONTINUE

     IF (LINC.EQ.1) THEN
        POWSUM = 0.Q0
        DO 800 I=1,IPRD-1
           POWSUM = POWSUM+(POWER1(I,1)+POWER2(I,1)
    +                      +(POWER2(I,1)-POWER1(I,1))
    +                      /QFLOAT(NLINC(I)))
    +                      *(ENDTIM(I)-ENDTIM(I-1))/2.Q0
800        CONTINUE
        ENDIF
      PERPOW = POWER(1)*DELTIM(IPRD)
      POWSUM = POWSUM+PERPOW

      TOTATM = POWSUM * 7.80275Q9 * GRNVOL

      IF (IREST.EQ.1) TOTATM = TOTATM+RESATM
      ATMKPT = QMIN1((TOTATM/ATOMS),1.Q0)
      BURATE = 1.72800Q7*FISSRC(1)*OMEGA
      BUAPPR = BURATE*DELTIM(IPRD)/8.6400Q4
      BUAPTL = BUAPTL + BUAPPR
      WTMOLE = ENRICH*235.043928Q0 + (1.Q0-ENRICH)*238.050788Q0
      BUMWTL = BUMWTL + 1.394Q19*OMEGA*POWER(1)*DELTIM(IPRD)/WTMOLE

       ATOMS = 0.Q0
       DO 1030 IJ = 1,NBBLSZ
          AVGVEL = VELCUM(IJ)/GRNVOL
          SHIFT = AVGVEL*DELTIM(IPRD)
          PCTKPT = (16.Q0*GRNDIA**3
    +             -12.Q0*SHIFT*GRNDIA**2+SHIFT**3)/(16.Q0*GRNDIA**3)
          ATOMS = ATOMS + ATOM(IJ)*ATMKPT*PCTKPT
          VOLUME = VOLUME + VOLCUM(IJ)*ATMKPT*PCTKPT
          DO 1031 IK = 1,NN
             INDEX = (IK-1)*NBBLSZ+IJ
             CONC(INDEX) = CONC(INDEX)*ATMKPT*PCTKPT
             DELCON(INDEX) = DELCON(INDEX)*ATMKPT*PCTKPT
1031         CONTINUE
1030      CONTINUE
      PCTSWL = 100.Q0*VOLUME/GRNVOL
      PCTSWL = QMAX1(PCTSWL,0.Q0)
      PCTSWL = QMIN1(PCTSWL,1.Q2)
      ATMREL = QMAX1((TOTATM-ATOMS),ATMREL)
      PCTREL = 100.Q0*ATMREL/TOTATM
      PCTREL = QMAX1(PCTREL,0.Q0)
      PCTREL = QMIN1(PCTREL,1.Q2)
       RETURN
      END
```

A2.1 Include Files

```
*=================================================================+
*                                                                 |
*              MMMMMM    AAAAAA   IIIIIIII  NN    NN             |
*              MMMMMMM  AAAAAAAA  IIIIIIII  NNN   NN             |
*              MM MM MM  AA    AA    II     NNNN  NN             |
*              MM MM MM  AA    AA    II     NN NN NN             |
*              MM MM MM  AAAAAAAA    II     NN  NNNN             |
*              MM MM MM  AAAAAAAA    II     NN   NNN             |
*              MM MM MM  AA    AA  IIIIIIII  NN    NN             |
*              MM MM MM  AA    AA  IIIIIIII  NN    NN             |
*                                                                 |
*=================================================================+
      IMPLICIT REAL*16 (A-H,O-Z)
      IMPLICIT INTEGER (I-N)

      COMMON/STORAG/R,CONC,DELCON,RAD,DELRAD,RADIUS,DIFFCF,VOLCUM
     +       ,BURATE,BUAPPR,BUAPTL,BUMWTL,RESOLP,RESADJ,RADMAX,RADMIN
     +       ,CONCV,CONCI,DELCNV,DELCNI,BBLPRS,CONCVE,ATOM,VELCUM
     +       ,DENSTY,POROS,VOL,SWEL,GRNDIA,SPDISP,DIFFS,YPRS,ENRICH
     +       ,LTPARM,OMEGA,LAMBDA,ZVAC,ZINT,DIFFN,DIFFU,DIFFV,DIFFI
     +       ,POWER1,POWER2,TEMP1,TEMP2,PRESS1,PRESS2,TMPGR1,TMPGR2
     +       ,POWER,TEMP,PRESS,TMPGR,PRSNGN,FISSRC,ATMSRC,VACSRC
     +       ,DELTIM,ENDTIM,POWSUM,RESATM,TOTATM
     +       ,RKBC,RRBC,ELK,ELF
     +       ,NE,NN,NSPC,NMAT,NNELEM,NBBLSZ,NPRD
     +       ,NNETYP,NEDOF,NDOF
     +       ,ICON,ICONDF
     +       ,JARRAY,LARRAY
     +       ,INONPD,IDOFPD,INONPV,INONPR
     +       ,IPRBTP,IMAT,NPD,IDIR,IREST
     +       ,MAXH,IGP,MAXHV,MAXHR
     +       ,HEADER

*----------------------+
*  Header variables.   |
*----------------------+
      CHARACTER*80 HEADER(10)

*----------------------+
*  Integer variables.  |
*----------------------+
```

```
*----------------------+
*-------------------------------------------------------------+
*    NE - number of elements                                  |
*         NOTE: Problem set up for maximum of 10 elements     |
*    NN - number of nodes                                     |
*         NOTE: Problem set up for maximum of 21 global nodes |
*               (4 local nodes for 10 elements)               |
*    NSPC - number of specified concentrations                |
*         NOTE: Only first and last node should have specified|
*               concentrations.  Thus NSPC =< 2*NBBLSZ        |
*    NMAT - number of material types                          |
*         NOTE: Currently only one material type allowed.     |
*    NELT - number of element types                           |
*         NOTE: Based only on number of nodes per element     |
*    NNELEM - number of nodes per element                     |
*         NOTE: Restricted to 2 or 3                          |
*    INONPD - number of non-prescribed degrees of freedom     |
*           for bubbles                                       |
*         NOTE: Simply NDOF minus NSPC                        |
*    INONPV - number of non-prescribed degrees of freedom     |
*           for vacancies                                     |
*         NOTE: Simply (NN-1)*2                               |
*    MAXH - the maximum half bandwidth of stiffness matrix    |
*           for bubbles                                       |
*    MAXHV - the maximum half bandwidth of stiffness matrix   |
*           for vacancies                                     |
*    NEDOF - number of degrees of freedom per element         |
*         NOTE: Simply NNELEM*NBBLSZ                          |
*    NBBLSZ - number of bubble groups per node                |
*         NOTE: Currently limited to 50 bubble groups         |
*               The bubble group index determines the number  |
*               of gas atoms in the bubble                    |
*    NDOF - number of degrees of freedom                      |
*         NOTE: Simply NN*NBBLSZ                              |
*    NPRD - number of time periods                            |
*         NOTE: Currently limited to 10                       |
*-------------------------------------------------------------+
      INTEGER*4 NE,NN,NSPC,NMAT,NELT,NNELEM
     +         ,INONPD,MAXH,NEDOF,NBBLSZ,NDOF
*-------------------------------------------------------------+
*    ICON(i,j) - mapping of global node number                |
*                to local node j of element i                 |
*           Dim: i = max number of global nodes               |
*                j = max number of local nodes                |
*-------------------------------------------------------------+
```

```
      INTEGER*4 ICON(21,4)
*-------------------------------------------------------------+
*    NLINC(i) - number of time steps in period i              |
*           Dim: i = max number of periods                    |
*    LOUT(i) - time steps per printout in period i            |
*           Dim: i = max number of periods                    |
*-------------------------------------------------------------+
      INTEGER*4 NLINC(10)
      INTEGER*4 LOUT(10)
*-------------------------------------------------------------+
*    IMAT(i) - type of material for element i                 |
*         NOTE: Currently set to 1 for all elements           |
*           Dim: i = max number of elements                   |
*-------------------------------------------------------------+
      INTEGER*4 IMAT(10)
*-------------------------------------------------------------+
*    NPD(i) - node for prescribed concentration i             |
*           Dim: i = max number of prescribed concentrations  |
*    IDIR(i) - bubble group for prescribed concentration i    |
*           Dim: i = max number of prescribed concentrations  |
*-------------------------------------------------------------+
      INTEGER*4 NPD(50)
      INTEGER*4 IDIR(50)
*-------------------------------------------------------------+
*    ICONDF(i,j) - mapping of global degrees of freedom       |
*                  to local degrees of freedom (i.e. bubble grou|
*           Dim: i = max number of elements                   |
*           Dim: i = max number of degrees of freedom / element |
*-------------------------------------------------------------+
      INTEGER*4 ICONDF(10,150)
*-------------------------------------------------------------+
*    JARRAY(i) - mapping of non-prescribed degrees of freedom |
*                to row i of global matrix                    |
*           Dim: i = max number of global degrees of freedom  |
*    LARRAY(i) - mapping of global matrix rows to             |
*                degree of freedom i                          |
*           Dim: i = max number of global degrees of freedom  |
*-------------------------------------------------------------+
      INTEGER*4 JARRAY(1050)
      INTEGER*4 LARRAY(1050)
*-------------------------------------------------------------+
*    IDOFPD(i) - connectivity array for prescribed degree of  |
*                freedom                                      |
*           Dim: i = max number of prescribed concentrations  |
*-------------------------------------------------------------+
      INTEGER*4 IDOFPD(50)

*----------------------+
*  Real variables.     |
*----------------------+
      REAL*16 LTPARM,OMEGA,LAMBDA,ZVAC,ZINT
*-------------------------------------------------------------+
*    R(i) - radial position of i'th node (cm)                 |
*           Dim: i = max number of global nodes               |
*-------------------------------------------------------------+
      REAL*16 R(21)
*-------------------------------------------------------------+
*    VOLUME - volume of grain (cm3)                           |
*    GRNDIA - fuel grain diameter (cm)                        |
*-------------------------------------------------------------+
      REAL*16 VOLUME,GRNDIA
*-------------------------------------------------------------+
*    DENSTY(i) - density of i'th element (g/cm3)              |
*           Dim: i = max number of elements                   |
*    POROS(i) - porosity of i'th element (%)                  |
*           Dim: i = max number of elements                   |
*-------------------------------------------------------------+
      REAL*16 DENSTY(10)
      REAL*16 POROS(10)
*-------------------------------------------------------------+
*    VOL(i) - volume of i'th element (i.e. sphere) (cm3)      |
*           Dim: i = max number of elements                   |
*    SWEL(i) - percent swelling of i'th element (i.e. sphere) |
*           Dim: i = max number of elements                   |
*-------------------------------------------------------------+
      REAL*16 VOL(10),SWEL(10)
*-------------------------------------------------------------+
*    DELTIM(i) - time step size of i'th period (s)            |
*           Dim: i = max number of periods                    |
*-------------------------------------------------------------+
      REAL*16 DELTIM(10)
*-------------------------------------------------------------+
*    RADMAX(i) - equilibrium radius for i'th bubble group (cm) |
*           Dim: i = max number of bubble groups              |
*    RADMIN(i) - minimum radius for i'th bubble group (cm)    |
*           Dim: i = max number of bubble groups              |
*    ATOM(i) - atoms retained for i'th bubble group           |
*           Dim: i = max number of bubble groups              |
*    VELCUM(i) - integrated velocities for i'th bubble group  |
*           Dim: i = max number of bubble groups              |
```

```
*    VOLCUM(i) - integrated volumes for i'th bubble group
*           Dim: i = max number of bubble groups
*-----------------------------------------------------------------+
      REAL*16 RADMAX(50), RADMIN(50), ATOM(50), VELCUM(50),VOLCUM(50)
*-----------------------------------------------------------------+
*    POWER1(i,j) - thermal initial power of i'th period
*                  and j'th material
*           Dim: i = max number of periods
*           Dim: j = max number of materials
*    POWER2(i,j) - thermal final power of i'th period
*                  and j'th material
*           Dim: i = max number of periods
*           Dim: j = max number of materials
*    PRESS1(i,j) - initial pressure of i'th period
*                  and j'th material
*           Dim: i = max number of periods
*           Dim: j = max number of materials
*    PRESS2(i,j) - final pressure of i'th period
*                  and j'th material
*           Dim: i = max number of periods
*           Dim: j = max number of materials
*    TEMP1(i,j) - initial temperature of i'th period
*                  and j'th material
*           Dim: i = max number of periods
*           Dim: j = max number of materials
*    TEMP2(i,j) - final temperature of i'th period
*                  and j'th material
*           Dim: i = max number of periods
*           Dim: j = max number of materials
*    TMPGR1(i,j) - initial temperature gradient of i'th period
*                  and j'th material
*           Dim: i = max number of periods
*           Dim: j = max number of materials
*    TMPGR2(i,j) - final temperature gradient of i'th period
*                  and j'th material
*           Dim: i = max number of periods
*           Dim: j = max number of materials
*-----------------------------------------------------------------+
      REAL*16 POWER1(10,1),POWER2(10,1)
     +       ,TEMP1(10,1),TEMP2(10,1)
     +       ,TMPGR1(10,1),TMPGR2(10,1)
     +       ,PRESS1(10,1),PRESS2(10,1)
*-----------------------------------------------------------------+
*    POWER(i) - thermal power for i'th material
*           Dim: i = max number of materials
*    TEMP(i) - temperature for i'th material
*           Dim: i = max number of materials
*    TMPGR(i) - temperature gradient for i'th material
*           Dim: i = max number of materials
*    PRESS(i) - hydrostatic pressure for i'th material
*           Dim: i = max number of materials
*    PRSNGN(i) - partial pressure of nitrogen
*           Dim: i = max number of materials
*           NOTE:  set to 1
*    FISSRC(i) - fission rate density for i'th material
*           Dim: i = max number of materials
*    ATMSRC(i) - gas atom production rate for i'th material
*           Dim: i = max number of materials
*    VACSRC(i) - vacancy production rate for i'th material
*           Dim: i = max number of materials
*    CONCVE(i,j) - equilibrium vacancy (j=2) and
*                  equilibrium interstitial (j=1) concentrations
*           Dim: i = max number of materials
*           Dim: j = 2
*-----------------------------------------------------------------+
      REAL*16 POWER(1),TEMP(1),PRESS(1),PRSNGN(1),FISSRC(1),ATMSRC(1)
      REAL*16 TMPGR(1),VACSRC(1),CONCVE(1,2)
*-----------------------------------------------------------------+
*    DIFFCF(i,j) - diffusion coefficient for i'th node
*                  and j'th bubble group
*           Dim: i = max number of nodes
*           Dim: j = max number of bubble groups
*    RADIUS(i,j) - bubble radius for i'th node
*                  and j'th bubble group
*           Dim: i = max number of nodes
*           Dim: j = max number of bubble groups
*    BBLPRS(i,j) - bubble pressure for i'th node
*                  and j'th bubble group
*           Dim: i = max number of nodes
*           Dim: j = max number of bubble groups
*-----------------------------------------------------------------+
      REAL*16 DIFFCF(21,50), RADIUS(21,50), BBLPRS(21,50)
*-----------------------------------------------------------------+
*    CONC(i) - concentration at end of last time step
*              for i'th global degree of freedom
*           Dim: i = max number of global degree of freedom
*    DELCON(i) - change in concentration since last time step
*              for i'th global degree of freedom
*           Dim: i = max number of global degree of freedom
*    RAD(i) - vacancy concentration at end of last time step
```

```
*            for i'th global degree of freedom                  |
*          Dim: i = max number of global degree of freedom      |
*    DELRAD(i) - change in concentration since last time step   |
*                for i'th global degree of freedom              |
*          Dim: i = max number of global degree of freedom      |
*    CONCV(i) - vacancy concentration at end of last time step  |
*               for i'th node                                   |
*          Dim: i = max number of nodes                         |
*    DELCNV(i) - change in concentration since last time step   |
*                for i'th node                                  |
*          Dim: i = max number of nodes                         |
*    CONCI(i) - interstitial concentration at end of last time  |
*               step for i'th node                              |
*          Dim: i = max number of nodes                         |
*    DELCNI(i) - change in concentration since last time step   |
*                for i'th node                                  |
*          Dim: i = max number of nodes                         |
*-----------------------------------------------------------------+
      REAL*16 CONC(1050),DELCON(1050)
      REAL*16 RAD(1050),DELRAD(1050)
      REAL*16 CONCV(21),DELCNV(21)
      REAL*16 CONCI(21),DELCNI(21)
*-----------------------------------------------------------------+
*    SPDISP(i) - value of specified concentration i               |
*          Dim: i = max number of prescribed concentrations       |
*-----------------------------------------------------------------+
      REAL*16 SPDISP(50)
*-----------------------------------------------------------------+
*    RKBC(i,j) - stiffness matrix                                 |
*          Dim: i = max number of global degrees of freedom       |
*          Dim: j = max number of local degrees of freedom        |
*    RRBC(i) - force vector                                       |
*          Dim: i = max number of global degrees of freedom       |
*-----------------------------------------------------------------+
      REAL*16 RKBC(1050,300), RRBC(1050)
*----------------------+
*  RKBC Dimensions     |
*----------------------+
      DATA IDMRB1,IDMRB2/1050,300/
*-----------------------------------------------------------------+
*    ELK(i,j) - elemental stiffness matrix                        |
*          Dim: i = max number of local degrees of freedom        |
*          Dim: j = max number of local degrees of freedom        |
*    ELF(i) - elemental force vector                              |
*          Dim: i = max number of local degrees of freedom        |
*-----------------------------------------------------------------+
      REAL*16 ELK(150,150),ELF(150)
*----------------------+
*  ELK Dimensions      |
*----------------------+
      DATA IDMEK1,IDMEK2,IDMEK3/150,150,1/
*----------------------+
*  ELF Dimensions      |
*----------------------+
      DATA IDMEF1,IDMEF2,IDMEF3/1,150,1/
*-----------------------------------------------------------------+
*    ENDTIM(i) - end time for period i in real time (seconds)     |
*          Dim: i = max number of periods                         |
*          NOTE: '0' index needed for initial time period         |
*-----------------------------------------------------------------+
      REAL*16 ENDTIM(0:10)

*=================================================================+
*                                                                 |
*BBBBBBB   000000  LL       TTTTTTTT ZZZZZZZZ MMMMMM   AAAAAA  NN    NN
*BBBBBBBB 00000000 LL       TTTTTTTT ZZZZZZZZ MMMMMMM AAAAAAAA NNN   NN
*BB    BB 00    00 LL          TT        ZZZ  MM MM MM AA    AA NNNN  NN
*BBBBBBB  00    00 LL          TT       ZZZ   MM MM MM AA    AA NN NN NN
*BBBBBBB  00    00 LL          TT      ZZZ    MM MM MM AAAAAAAA NN  NNNN
*BB    BB 00    00 LL          TT     ZZZ     MM MM MM AAAAAAAA NN   NNN
*BBBBBBBB 00000000 LLLLLLLL    TT    ZZZZZZZZ MM MM MM AA    AA NN    NN
*BBBBBBB   000000  LLLLLLLL    TT    ZZZZZZZZ MM MM MM AA    AA NN    NN
*                                                                 |
*=================================================================+
      REAL*16 BLTZMN
*---------------------+
*  Units are in erg/K |
*---------------------+
      DATA BLTZMN/1.380662Q-16/

*=================================================================+
*                                                                 |
*DDDDDDD  IIIIIIII FFFFFFFF FFFFFFFF  CCCCCC   000000  EEEEEEEE FFFFFFFF
*DDDDDDDD IIIIIIII FFFFFFFF FFFFFFFF CCCCCCCC 00000000 EEEEEEEE FFFFFFFF
*DD    DD    II    FF       FF       CC    CC 00    00 EE       FF      |
*DD    DD    II    FFFFF    FFFFF    CC       00    00 EEEEE    FFFFF   |
*DD    DD    II    FFFFF    FFFFF    CC       00    00 EEEEE    FFFFF   |
*DD    DD    II    FF       FF       CC    CC 00    00 EE       FF      |
*DDDDDDDD IIIIIIII FF       FF       CCCCCCCC 00000000 EEEEEEEE FF      |
*DDDDDDD  IIIIIIII FF       FF        CCCCCC   000000  EEEEEEEE FF      |
```

```
*                                                                      |
*  DIFFVA = pre-exponential for vacancy diffusion coefficient (cm2/s)  |
*  DIFFVB = exponential term for vacancy diffusion coefficient (ergs*K/|
*  DIFFIA = pre-exponential for interst diffusion coefficient (cm2/s)  |
*  DIFFIB = exponential term for interst diffusion coefficient (ergs*K/|
*  DIFFXA = pre-exponential for xenon diffusion coefficient (cm2/s)    |
*  DIFFXB = exponential term for xenon diffusion coefficient (ergs*K/er|
*  DIFFSA = pre-exponential for surface diffusion coefficient (cm2/s)  |
*  DIFFSB = exponential term for surface diffusion coefficient (ergs*K/|
*  DIFFLM = mean jump distance of surface atom (cm)                    |
*  DIFFIE = irradiation enhanced diffusion multiplier                  |
*  DIFFUA = pre-exponential for uranium self diffusion (cm2/s)         |
*  DIFFUB = partial pressure exponential                               |
*  DIFFUC = exponential term for uranium self diffusion (ergs*K/erg)   |
*  DIFFNA = pre-exponential for nitrogen self diffusion (cm2/s)        |
*  DIFFNB = partial pressure exponential                               |
*  DIFFNC = exponential term for nitrogen self diffusion (ergs*K/erg)  |
*  DIFND1 = constant exponent for grain size                           |
*  DIFND2 = temperature dependent exponent for grain size              |
*                                                                      |
*======================================================================+
      REAL*16 DIFFXA,DIFFXB,DIFFSA,DIFFSB
       DATA DIFFVA,DIFFVB/0.6Q0,1.53Q4/
       DATA DIFFIA,DIFFIB/1.Q-3,2.32Q3/
       DATA DIFFXA,DIFFXB/4.6536Q-04,3.6073Q4/
       DATA DIFFSA,DIFFSB/1.9200Q+03,6.5188Q4/
       DATA DIFFLM/1.12Q-6/
       DATA DIFFIE/8.22Q-31/
       DATA DIFFUA,DIFFUB,DIFFUC/2.2146Q-11,0.6414Q0,7.9893Q3/
       DATA DIFFNA,DIFFNB,DIFFNC/2.251Q-05,0.4134Q0,1.92147Q4/
       DATA DIFND1,DIFND2/-0.737Q0,2.179Q-4/

*======================================================================+
*                                                                      |
*               EEEEEEEE  LL        KK    KK  FFFFFFFF                |
*               EEEEEEEE  LL        KK   KK   FFFFFFFF                |
*               EE        LL        KK  KK    FF                      |
*               EEEEE     LL        KKKKK     FFFFF                   |
*               EEEEE     LL        KKKKK     FFFFF                   |
*               EE        LL        KK  KK    FF                      |
*               EEEEEEEE  LLLLLLLL  KK   KK   FF                      |
*               EEEEEEEE  LLLLLLLL  KK    KK  FF                      |
*                                                                      |
*    TELK(i,j) - temporary elemental stiffness matrix                  |
*           Dim: i = max number of local degrees of freedom            |
*           Dim: j = max number of local degrees of freedom            |
*    TELF(i,j) - temporary elemental force vector                      |
*           Dim: i = max number of local degrees of freedom            |
*======================================================================+
      REAL*16 TELK(150,150),TELF(150)
      REAL*16 CONCIP(50), DELCIP(50), DERCIP(50), SRCFRC(50)
      REAL*16 CNRDIP(50), DLRDIP(50), DRRDIP(50)
      REAL*16 RADIP(50), DFCFIP(50), BPRSIP(50), VELIP(50), RESOL(50)
      REAL*16 BPRSA(50), BPRSB(50),BPTMIP(50)
*--------------------+
*  TELK Dimensions   |
*--------------------+
      DATA IDMCM1,IDMCM2,IDMCM3/150,150,1/
*--------------------+
*  TELF Dimensions
*--------------------+
      DATA IDMCV1,IDMCV2,IDMCV3/1,150,1/

*======================================================================+
*                                                                      |
*            GGGGGG    AAAAAA   UU    UU   SSSSSS    SSSSSS          |
*           GGGGGGGG  AAAAAAAA  UU    UU  SSSSSSSS  SSSSSSSS         |
*           GG        AA    AA  UU    UU  SS        SS                |
*           GG  GGGG  AA    AA  UU    UU  SSSSSSS   SSSSSSS           |
*           GG  GGGG  AAAAAAAA  UU    UU   SSSSSSS   SSSSSSS          |
*           GG    GG  AAAAAAAA  UU    UU        SS        SS         |
*           GGGGGGGG  AA    AA  UUUUUUUU  SSSSSSSS  SSSSSSSS         |
*            GGGGGG   AA    AA   UUUUUU    SSSSSS    SSSSSS          |
*                                                                      |
*======================================================================+
*----------------------------------------------------------------------+
*    SHPFCT(i) - shape factor array                                    |
*           Dim: i = max number of local nodes                         |
*    DERSHP(i) - shape factor array                                    |
*           Dim: i = max number of local nodes                         |
*----------------------------------------------------------------------+
      REAL*16 SHPFCT(4),DERSHP(4)
*----------------------------------------------------------------------+
*    RNAT(i,j) - Gauss quadrature roots                                |
*           Dim: i = max number of choices for integration pts         |
*    WEIGHT(i,j) - Gauss quadrature weights                            |
*           Dim: i = max number of choices for integration pts         |
*----------------------------------------------------------------------+
      REAL*16 RNAT(6,6),WEIGHT(6,6)
*----------------------------------------------------------------------+
```

```
*    Gauss quadrature roots                                          |
*-----------------------------------------------------------------+
      DATA RNAT/       0.0Q0,                0.0Q0,
     +                 0.0Q0,                0.0Q0,
     +                 0.0Q0,                0.0Q0,
     +  -0.5773502691896260Q0,  0.5773502691896260Q0,
     +                 0.0Q0,                0.0Q0,
     +                 0.0Q0,                0.0Q0,
     +  -0.7745966924148300Q0,                0.0Q0,
     +   0.7745966924148300Q0,                0.0Q0,
     +                 0.0Q0,                0.0Q0,
     +  -0.8611363115940530Q0, -0.3399810435848560Q0,
     +   0.3399810435848560Q0,  0.8611363115940530Q0,
     +                 0.0Q0,                0.0Q0,
     +  -0.9061798459386640Q0, -0.5384693101056830Q0,
     +                 0.0Q0,  0.5384693101056830Q0,
     +   0.9061798459386640Q0,                0.0Q0,
     +  -0.9324695142031520Q0, -0.6612093864662650Q0,
     +  -0.2386191860831970Q0,  0.2386191860831970Q0,
     +   0.6612093864662650Q0,  0.9324695142031520Q0/
*-----------------------------------------------------------------+
*    Gauss quadrature weights                                        |
*-----------------------------------------------------------------+
      DATA WEIGHT/     0.0Q0,                0.0Q0,
     +                 0.0Q0,                0.0Q0,
     +                 0.0Q0,                0.0Q0,
     +                 1.0Q0,                1.0Q0,
     +                 0.0Q0,                0.0Q0,
     +                 0.0Q0,                0.0Q0,
     +   0.5555555555555560Q0,  0.8888888888888890Q0,
     +   0.5555555555555560Q0,                0.0Q0,
     +                 0.0Q0,                0.0Q0,
     +   0.3478548451374540Q0,  0.6521451548625460Q0,
     +   0.6521451548625460Q0,  0.3478548451374540Q0,
     +                 0.0Q0,                0.0Q0,
     +   0.2369268850561890Q0,  0.4786286704993660Q0,
     +   0.5688888888888890Q0,  0.4786286704993660Q0,
     +   0.2369268850561890Q0,                0.0Q0,
     +   0.1713244923791700Q0,  0.3607615730481390Q0,
     +   0.4679139345726910Q0,  0.4679139345726910Q0,
     +   0.3607615730481390Q0,  0.1713244923791700Q0/

*=================================================================+
*                                                                 |
*          IIIIIIII  NN    NN   000000   UU    UU  TTTTTTTT       |
*          IIIIIIII  NNN   NN  00000000  UU    UU  TTTTTTTT       |
*             II     NNNN  NN  00    00  UU    UU     TT          |
*             II     NN NN NN  00    00  UU    UU     TT          |
*             II     NN  NNNN  00    00  UU    UU     TT          |
*             II     NN   NNN  00    00  UU    UU     TT          |
*          IIIIIIII  NN    NN  00000000  UUUUUUUU     TT          |
*          IIIIIIII  NN    NN   000000    UUUUUU      TT          |
*                                                                 |
*=================================================================+
      INTEGER*4 IFLD(40),IDATE(14)
      LOGICAL ITEND
      CHARACTER*80 BUFFER
      CHARACTER*50 PROBLM(5)
      CHARACTER*13 TEMPTP(5)
      CHARACTER*30 ELEMNT(6),COORD(5)
      CHARACTER*60 ELPRB1(5),ELPRB2(5,2),ELPRB3(5)
      CHARACTER*10 UNITS(0:8,2),CRDSYS(3,5),CRDUNT(3,2,5)
      CHARACTER*23 DATE
      CHARACTER*11 TIME
      CHARACTER*36 MONTHS
      CHARACTER*63 DAYS
      CHARACTER*4 AMPM

      DATA DATE/'                    '/
      DATA TIME/'          '/
      DATA MONTHS/'JanFebMarAprMayJunJulAugSepOctNovDec'/
      DATA DAYS(1:36)/'Sunday   Monday   Tuesday  Wednesday'/
      DATA DAYS(37:63) /'Thursday Friday   Saturday '/
      DATA AMPM/'AMPM'/

      DATA PROBLM/'Single Fuel Grain                                ',
     +            '                                                 ',
     +            '                                                 ',
     +            '                                                 ',
     +            '                                                 '/
      DATA COORD/' (1=R)                                 ',
     +           '                                       ',
     +           '                                       ',
     +           '                                       ',
     +           '                                       '/
      DATA ELEMNT/'Spherical                             ',
     +            '                                      ',
     +            '                                      ',
     +            '                                      ',
     +            '                                      ',
```

```
     +          '                              '/
      DATA UNITS/'SI       ',' m',' N/m**2 ',' m/m-C ','meters','N ','C',
     +  ' N/m**3 ','   N/N ',
     +          'BRITISH','in','lb/in**2','ft/ft-F','inches','lb','F',
     +  'lb/in**3','  in/in'/
      DATA CRDSYS/'  R(I)  ','        ','        ',
     +            '        ','        ','        ',
     +            '        ','        ','        ',
     +            '        ','        ','        ',
     +            '        ','        ','        '/
      DATA CRDUNT/'meters','meters','meters',
     +            'inches','inches','inches',
     +            'meters','meters','radian',
     +            'inches','inches','radian',
     +            'meters','meters','radian',
     +            'inches','inches','radian',
     +            'meters','meters','meters',
     +            'inches','inches','inches',
     +            '      ','      ','      ',
     +            '      ','      ','      '/
*=============================================================+
*                                                             |
* LL         AAAAAA   TTTTTTTT  TTTTTTTT  IIIIIIII   CCCCCC   EEEEEEEE |
* LL        AAAAAAAA  TTTTTTTT  TTTTTTTT  IIIIIIII  CCCCCCCC  EEEEEEEE |
* LL        AA    AA     TT        TT        II     CC    CC  EE       |
* LL        AA    AA     TT        TT        II     CC        EEEEE    |
* LL        AAAAAAAA     TT        TT        II     CC        EEEEE    |
* LL        AAAAAAAA     TT        TT        II     CC    CC  EE       |
* LLLLLLLL  AA    AA     TT        TT     IIIIIIII  CCCCCCCC  EEEEEEEE |
* LLLLLLLL  AA    AA     TT        TT     IIIIIIII   CCCCCC   EEEEEEEE |
*                                                             |
*=============================================================+
      REAL*16 LTPRMA,LTPRMB,LTPRMC
      REAL*16 EFORMV,DISDEN,DISRAD
      REAL*16 YPSLPE,YPINT

      DATA LTPRMA,LTPRMB,LTPRMC/4.879Q-8,3.264Q-13,6.889Q-17/
      DATA EFORMV,DISDEN,DISRAD/1.96Q4,1.Q8,25.Q-8/
      DATA YPSLPE,YPINT/-4.628196154965516 4Q6,8.673268330830046 3Q9/

*=============================================================+
*                                                             |
*                      PPPPPPP   IIIIIIII                     |
*                      PPPPPPPP  IIIIIIII                     |
*                      PP    PP     II                        |
*                      PP    PP     II                        |
*                      PPPPPPPP     II                        |
*                      PPPPPPP      II                        |
*                      PP        IIIIIIII                     |
*                      PP        IIIIIIII                     |
*                                                             |
*=============================================================+
      REAL*16 PI
      DATA PI/3.14159265358979323846264338327950288Q0/

*=============================================================+
*                                                             |
* SSSSSS  UU    UU RRRRRRR  FFFFFFFF EEEEEEEE NN    NN  GGGGGG  YY    YY
*SSSSSSSS UU    UU RRRRRRRR FFFFFFFF EEEEEEEE NNN   NN GGGGGGGG YY    YY
*SS       UU    UU RR    RR FF       EE       NNNN  NN GG       YY    YY
*SSSSSSS  UU    UU RR    RR FFFFF    EEEEE    NN NN NN GG  GGGG  YY  YY
* SSSSSSS UU    UU RRRRRRRR FFFFF    EEEEE    NN  NNNN GG  GGGG   YYYY
*      SS UU    UU RRRRRR   FF       EE       NN   NNN GG    GG    YY
*SSSSSSSS UUUUUUUU RR   RR  FF       EEEEEEEE NN    NN GGGGGGGG    YY
* SSSSSS   UUUUUU  RR    RR FF       EEEEEEEE NN    NN  GGGGGG     YY
*                                                             |
*=============================================================+
      REAL*16 SRFENG,SRFACT,RESLMH,RESLML

*-------------------------------------------------------------+
*  Units are in dyn/cm                                        |
*     equal to 1 erg                                          |
*  SRFACT is activation energy for surface atoms to break free|
*    in units of ergs                                         |
*  RESLMH and RESLML are high temp and low temp resolution    |
*    factors respectively.  High temp is defined as >1500 K   |
*    while low temp is <500 K                                 |
*-------------------------------------------------------------+
      DATA SRFENG,SRFACT/1.Q3,9.0003Q-12/
      DATA RESLMH,RESLML/1.1Q-19,4.1Q-19/

*=============================================================+
*                                                             |
*VV    VV  AAAAAA  NN    NN DDDDDDD  WW    WW  AAAAAA   AAAAAA  LL      |
*VV    VV AAAAAAAA NNN   NN DDDDDDDD WW    WW AAAAAAAA AAAAAAAA LL      |
*VV    VV AA    AA NNNN  NN DD    DD WW    WW AA    AA AA    AA LL      |
*VV    VV AA    AA NN NN NN DD    DD WW WW WW AA    AA AA    AA LL      |
*VV    VV AAAAAAAA NN  NNNN DD    DD WW WW WW AAAAAAAA AAAAAAAA LL      |
* VV  VV  AAAAAAAA NN   NNN DD    DD WW WW WW AAAAAAAA AAAAAAAA LL      |
*  VVVV   AA    AA NN    NN DDDDDDDD WWWWWWWW AA    AA AA    AA LLLLLLLL |
```

```
*   VV     AA     AA NN     NN DDDDDDD   WW  WW  AA     AA AA     AA LLLLLLLL
*                                                                          |
*==========================================================================+
      REAL*16 VANDWL(2)
*----------------------------+
*  Units are in (1) - g cm4  |
*               (2) - cm3    |
*----------------------------+
      DATA VANDWL/7.1953543Q-33,84.76616659Q-24/
```

APPENDIX 2

SAMPLE INPUT

```
*--------------------------------+
* OUPUT DECK ? - 0: NO 1: YES    |
*--------------------------------+
000   0
*--------------------------------+
* HEADER CARDS, MAX 80 CHARACTERS |
*--------------------------------+
100   UN Fuel Grain Modelling Using Nonlinear Finite Element
Analysis
101               By Capt. Daniel L. DeForest, USAF
102
103                   Sample Input Deck
105                *****************************
106                         CONDITIONS
107                *****************************
108                Power Density = 700.00 W/cm3
109                        Temp = 1200 K
110                   Temp Grad  = 1000 K/cm
111             External Pressure = 1033 Pa
112                   Time Length = 1.0E5 s
*-----------------+
*  TYPE OF UNITS  |
*-----------------+
201 2
*-------------------------------------+
* NUMBER OF GUASS INTEGRATION POINTS  |
*-------------------------------------+
202 3
*-------------------------------------+
* LIST INPUT DECK FLAG                |
*-------------------------------------+
203 0
*-------------------------------------+
* RESTART FLAG                        |
*-------------------------------------+
204 0
*----------------------------------------+
*   NE,NBBLSZ,NNELEM,NSPC,NPRD,MAXITER   |
*----------------------------------------+
300  10   10     3    10   2      50
*---------------------------------+
* R COORDINATES IN ORDER OF NODE  |
*---------------------------------+
501    0.0Q0
502    1.25Q-4
503    2.5Q-4
504    3.5Q-4
505    4.5Q-4
506    5.5Q-4
507    6.5Q-4
508    7.0Q-4
509    7.5Q-4
510    8.0Q-4
511    8.5Q-4
```

```
512    8.9Q-4
513    9.3Q-4
514    9.5Q-4
515    9.7Q-4
516    9.8Q-4
517    9.9Q-4
518    9.925Q-4
519    9.95Q-4
520    9.9875Q-4
521   10.0Q-4
*-------------------------------------------------+
*  SPECIFIED CONCENTRATIONS - NODE #, BBLGP, VALUE |
*-------------------------------------------------+
601 21   1       0.0Q0
602 21   2       0.0Q0
603 21   3       0.0Q0
604 21   4       0.0Q0
605 21   5       0.0Q0
606 21   6       0.0Q0
607 21   7       0.0Q0
608 21   8       0.0Q0
609 21   9       0.0Q0
610 21  10       0.0Q0
*-----------------+
*  GRAIN RADIUS   |
*-----------------+
701 10.Q-4
*-----------------------------------------------------------+
* INITIAL POROSITY, INITIAL ENRICHMENT, CONVERGENCE LIMIT  |
*-----------------------------------------------------------+
901  0.057  0.10  1.Q-4
*---------------------------------------------+
* PERIOD LENGTH, #TIME STEPS, PRINTOUT INTERVAL |
*---------------------------------------------+
1001  1.0Q5  5     5
*-------------------------------------------------+
*  MATERIALS - POWER,TEMP,TEMP GRADIENT,PRESSURE  |
*-------------------------------------------------+
1102  700.0  1200.0 1000.0 1033.23  700.0  1200.0 1000.0 1033.23
```

APPENDIX 3

SAMPLE OUTPUT

1

```
RRRRRRR    EEEEEEEE  DDDDDDD    SSSSSS   TTTTTTTT   000000   NN    NN  EEEEEEEE
RRRRRRRR   EEEEEEEE  DDDDDDDD  SSSSSSSS  TTTTTTTT  00000000  NNN   NN  EEEEEEEE
RR    RR   EE        DD    DD  SS           TT     00    00  NNNN  NN  EE
RR    RR   EEEEEE    DD    DD  SSSSSSS      TT     00    00  NN NN NN  EEEEEE
RRRRRRRR   EEEEEE    DD    DD   SSSSSSS     TT     00    00  NN  NNNN  EEEEEE
RRRRRR     EE        DD    DD        SS     TT     00    00  NN   NNN  EE
RR   RR    EEEEEEEE  DDDDDDDD  SSSSSSSS     TT     00000000  NN    NN  EEEEEEEE
RR    RR   EEEEEEEE  DDDDDDD    SSSSSS      TT      000000   NN    NN  EEEEEEEE
```

PROGRAM REDSTONE

UN Fuel Grain Modelling Using Nonlinear Finite Element Analysis
By Capt. Daniel L. DeForest, USAF

Sample Input Deck

****

CONDITIONS

****

Power Density = 700.00 W/cm3
Temp = 1200 K
Temp Grad = 1000 K/cm
External Pressure = 1033 Pa
Time Length = 1.0E5 s

1

Total number of elements	=	10
Total number of nodes	=	21
Maximum bubble groups per node	=	10
Total number of material types	=	1
Total number of specified displacements	=	10
Total number of time periods	=	2
Maximum number of iterations	=	50

Units are in BRITISH

Number of Guass integration points is 3

Grain Diameter = 1.000E-3 microns

NODAL POSITIONS FROM CENTER

Global Node>I	R(I) (inches)
1	0.000000E+0
2	1.250000E-4
3	2.500000E-4
4	3.500000E-4
5	4.500000E-4
6	5.500000E-4
7	6.500000E-4
8	7.000000E-4
9	7.500000E-4
10	8.000000E-4
11	8.500000E-4
12	8.900000E-4
13	9.300000E-4
14	9.500000E-4
15	9.700000E-4
16	9.800000E-4
17	9.900000E-4
18	9.925000E-4
19	9.950000E-4
20	9.987500E-4
21	1.000000E-3

ELEMENT CONNECTIVITY

	Global Node Number for the Local Node Numbers Below			
Element	1	2	3	4
1	1	2	3	0
2	3	4	5	0
3	5	6	7	0
4	7	8	9	0
5	9	10	11	0
6	11	12	13	0
7	13	14	15	0
8	15	16	17	0
9	17	18	19	0
10	19	20	21	0

ELEMENT MATERIAL PROPERTIES

Element>I	Material Type ID	Initial Porosity
1	1	5.7000E-2
2	1	5.7000E-2
3	1	5.7000E-2
4	1	5.7000E-2
5	1	5.7000E-2
6	1	5.7000E-2
7	1	5.7000E-2
8	1	5.7000E-2
9	1	5.7000E-2
10	1	5.7000E-2

1 TIME HISTORY CONDITIONS

Time Incr	Initial Power	Initial Temp	Initial Press	Final Power	Final Temp	Final Press
1	700.	1.200E+3	1.033E+3	700.	1.200E+3	1.033E+3
2	0.000E+0	0.000E+0	0.000E+0	0.000E+0	0.000E+0	0.000E+0

1 TIME STEP INFORMATION

Time Incr	Length (seconds)	Number of Time Steps	Time Step (seconds)	Printout Interval
1	1.000E+05	5	2.000E+04	5
2	0.000E+00	0	-1.000E+05	0

1

SPECIFIED CONCENTRATIONS

Global Node #	Bubble Gp	Value
21	1	0.000E+00
21	2	0.000E+00
21	3	0.000E+00
21	4	0.000E+00
21	5	0.000E+00
21	6	0.000E+00
21	7	0.000E+00
21	8	0.000E+00
21	9	0.000E+00
21	10	0.000E+00

1

TIME PERIOD 1
Time Step 5

Time Step Size = 2.0000E+4 seconds
Time in Period = 1.0000E+5 seconds
New Time = 1.0000E+5 seconds

Time step represents 100% of full time period.

Percent fission gas released = 0.4815 %
Percent swelling = 4.2423E-03%

Burnup rate = 1.129E-02 atom percent/day
Burnup for period = 2.614E-03 atom percent
Burnup (total) = 1.307E-02 atom percent
= 122.80 MWD/MT

Power density = 700.0 W/cm3
Pressure = 1033. Pa
Temperature = 1200. K

NODAL CONCENTRATIONS

Bubble Group	Concentration (cm-3)	Bubble Radius (cm)	Bubble Pressure (Pa)
--------- Node = 1 ----- Radius = 0.000E+00----------			
1	1.907419E+17	2.426457E-08	0.000000E+00
2	1.152346E+17	3.471774E-08	5.760742E+10
3	2.970138E+16	3.980550E-08	5.024431E+10
4	6.723207E+15	4.386735E-08	4.559199E+10
5	1.396060E+15	4.730536E-08	4.227850E+10
6	2.723131E+14	5.031653E-08	3.974836E+10
7	5.061627E+13	5.301388E-08	3.772596E+10
8	9.051949E+12	5.546897E-08	3.605619E+10
9	1.577364E+12	5.773034E-08	3.464382E+10
10	2.973562E+11	5.983260E-08	3.342659E+10
--------- Node = 2 ----- Radius = 1.250E-04----------			
1	1.907419E+17	2.426457E-08	0.000000E+00
2	1.152346E+17	3.471774E-08	5.760742E+10
3	2.970138E+16	3.980550E-08	5.024431E+10
4	6.723206E+15	4.386735E-08	4.559199E+10
5	1.396059E+15	4.730536E-08	4.227850E+10
6	2.723131E+14	5.031653E-08	3.974836E+10
7	5.061627E+13	5.301388E-08	3.772596E+10
8	9.051950E+12	5.546897E-08	3.605619E+10
9	1.577365E+12	5.773034E-08	3.464382E+10
10	2.973562E+11	5.983260E-08	3.342659E+10
--------- Node = 3 ----- Radius = 2.500E-04----------			
1	1.907419E+17	2.426457E-08	0.000000E+00
2	1.152346E+17	3.471774E-08	5.760742E+10
3	2.970138E+16	3.980550E-08	5.024431E+10
4	6.723206E+15	4.386735E-08	4.559199E+10
5	1.396059E+15	4.730536E-08	4.227850E+10
6	2.723131E+14	5.031653E-08	3.974836E+10
7	5.061627E+13	5.301388E-08	3.772596E+10
8	9.051950E+12	5.546897E-08	3.605619E+10
9	1.577364E+12	5.773034E-08	3.464382E+10
10	2.973562E+11	5.983260E-08	3.342659E+10
--------- Node = 4 ----- Radius = 3.500E-04----------			
1	1.907419E+17	2.426457E-08	0.000000E+00
2	1.152346E+17	3.471774E-08	5.760742E+10
3	2.970138E+16	3.980550E-08	5.024431E+10
4	6.723206E+15	4.386735E-08	4.559199E+10
5	1.396059E+15	4.730536E-08	4.227850E+10
6	2.723131E+14	5.031653E-08	3.974836E+10
7	5.061627E+13	5.301388E-08	3.772596E+10
8	9.051950E+12	5.546897E-08	3.605619E+10
9	1.577365E+12	5.773034E-08	3.464382E+10
10	2.973562E+11	5.983260E-08	3.342659E+10
--------- Node = 5 ----- Radius = 4.500E-04----------			
1	1.907419E+17	2.426457E-08	0.000000E+00
2	1.152346E+17	3.471774E-08	5.760742E+10
3	2.970138E+16	3.980550E-08	5.024431E+10

4	6.723207E+15	4.386735E-08	4.559199E+10
5	1.396060E+15	4.730536E-08	4.227850E+10
6	2.723131E+14	5.031653E-08	3.974836E+10
7	5.061627E+13	5.301388E-08	3.772596E+10
8	9.051949E+12	5.546897E-08	3.605619E+10
9	1.577364E+12	5.773034E-08	3.464382E+10
10	2.973562E+11	5.983260E-08	3.342659E+10

--------- Node = 6 ----- Radius = 5.500E-04----------

1	1.907419E+17	2.426457E-08	0.000000E+00
2	1.152346E+17	3.471774E-08	5.760742E+10
3	2.970138E+16	3.980550E-08	5.024431E+10
4	6.723206E+15	4.386735E-08	4.559199E+10
5	1.396059E+15	4.730536E-08	4.227850E+10
6	2.723131E+14	5.031653E-08	3.974836E+10
7	5.061627E+13	5.301388E-08	3.772596E+10
8	9.051950E+12	5.546897E-08	3.605619E+10
9	1.577365E+12	5.773034E-08	3.464382E+10
10	2.973563E+11	5.983260E-08	3.342659E+10

--------- Node = 7 ----- Radius = 6.500E-04----------

1	1.907419E+17	2.426457E-08	0.000000E+00
2	1.152346E+17	3.471774E-08	5.760742E+10
3	2.970139E+16	3.980550E-08	5.024431E+10
4	6.723208E+15	4.386735E-08	4.559199E+10
5	1.396060E+15	4.730536E-08	4.227850E+10
6	2.723132E+14	5.031653E-08	3.974836E+10
7	5.061626E+13	5.301388E-08	3.772596E+10
8	9.051948E+12	5.546897E-08	3.605619E+10
9	1.577364E+12	5.773034E-08	3.464382E+10
10	2.973560E+11	5.983260E-08	3.342659E+10

--------- Node = 8 ----- Radius = 7.000E-04----------

1	1.907419E+17	2.426457E-08	0.000000E+00
2	1.152345E+17	3.471774E-08	5.760742E+10
3	2.970137E+16	3.980550E-08	5.024431E+10
4	6.723204E+15	4.386735E-08	4.559199E+10
5	1.396059E+15	4.730536E-08	4.227850E+10
6	2.723131E+14	5.031653E-08	3.974836E+10
7	5.061627E+13	5.301388E-08	3.772596E+10
8	9.051952E+12	5.546897E-08	3.605619E+10
9	1.577365E+12	5.773034E-08	3.464382E+10
10	2.973565E+11	5.983260E-08	3.342659E+10

--------- Node = 9 ----- Radius = 7.500E-04----------

1	1.907420E+17	2.426457E-08	0.000000E+00
2	1.152348E+17	3.471774E-08	5.760742E+10
3	2.970145E+16	3.980550E-08	5.024431E+10
4	6.723221E+15	4.386735E-08	4.559199E+10
5	1.396062E+15	4.730536E-08	4.227850E+10
6	2.723133E+14	5.031653E-08	3.974836E+10
7	5.061624E+13	5.301388E-08	3.772596E+10
8	9.051934E+12	5.546897E-08	3.605619E+10
9	1.577360E+12	5.773034E-08	3.464382E+10
10	2.973548E+11	5.983260E-08	3.342659E+10

--------- Node = 10 ----- Radius = 8.000E-04----------

1	1.907418E+17	2.426457E-08	0.000000E+00
2	1.152344E+17	3.471774E-08	5.760742E+10
3	2.970132E+16	3.980550E-08	5.024431E+10
4	6.723194E+15	4.386735E-08	4.559199E+10
5	1.396058E+15	4.730536E-08	4.227850E+10
6	2.723130E+14	5.031653E-08	3.974836E+10
7	5.061629E+13	5.301388E-08	3.772596E+10
8	9.051963E+12	5.546897E-08	3.605619E+10
9	1.577369E+12	5.773034E-08	3.464382E+10
10	2.973574E+11	5.983260E-08	3.342659E+10

--------- Node = 11 ----- Radius = 8.500E-04----------

1	1.907426E+17	2.426457E-08	0.000000E+00
2	1.152358E+17	3.471774E-08	5.760742E+10
3	2.970175E+16	3.980550E-08	5.024431E+10
4	6.723284E+15	4.386735E-08	4.559199E+10
5	1.396071E+15	4.730536E-08	4.227850E+10
6	2.723141E+14	5.031653E-08	3.974836E+10
7	5.061614E+13	5.301388E-08	3.772596E+10
8	9.051868E+12	5.546897E-08	3.605619E+10
9	1.577339E+12	5.773034E-08	3.464382E+10
10	2.973490E+11	5.983260E-08	3.342659E+10

--------- Node = 12 ----- Radius = 8.900E-04----------

1	1.907413E+17	2.426457E-08	0.000000E+00
2	1.152335E+17	3.471774E-08	5.760742E+10
3	2.970104E+16	3.980550E-08	5.024431E+10
4	6.723135E+15	4.386735E-08	4.559199E+10
5	1.396049E+15	4.730536E-08	4.227850E+10
6	2.723123E+14	5.031653E-08	3.974836E+10
7	5.061639E+13	5.301388E-08	3.772596E+10
8	9.052025E+12	5.546897E-08	3.605619E+10
9	1.577388E+12	5.773034E-08	3.464382E+10
10	2.973629E+11	5.983260E-08	3.342659E+10

--------- Node = 13 ----- Radius = 9.300E-04----------

1	1.907460E+17	2.426457E-08	0.000000E+00
2	1.152416E+17	3.471774E-08	5.760742E+10
3	2.970360E+16	3.980550E-08	5.024431E+10
4	6.723668E+15	4.386735E-08	4.559199E+10
5	1.396128E+15	4.730536E-08	4.227850E+10
6	2.723188E+14	5.031653E-08	3.974836E+10
7	5.061556E+13	5.301388E-08	3.772596E+10
8	9.051475E+12	5.546897E-08	3.605619E+10
9	1.577216E+12	5.773034E-08	3.464382E+10
10	2.973139E+11	5.983260E-08	3.342659E+10

--------- Node = 14 ----- Radius = 9.500E-04----------

1	1.907370E+17	2.426457E-08	0.000000E+00
2	1.152263E+17	3.471774E-08	5.760742E+10
3	2.969876E+16	3.980550E-08	5.024431E+10
4	6.722662E+15	4.386735E-08	4.559199E+10
5	1.395979E+15	4.730536E-08	4.227850E+10
6	2.723068E+14	5.031653E-08	3.974836E+10

7	5.061717E+13	5.301388E-08	3.772596E+10
8	9.052528E+12	5.546897E-08	3.605619E+10
9	1.577544E+12	5.773034E-08	3.464382E+10
10	2.974071E+11	5.983260E-08	3.342659E+10

--------- Node = 15 ----- Radius = 9.700E-04----------

1	1.907768E+17	2.426457E-08	0.000000E+00
2	1.152929E+17	3.471774E-08	5.760742E+10
3	2.971989E+16	3.980550E-08	5.024431E+10
4	6.727095E+15	4.386735E-08	4.559199E+10
5	1.396649E+15	4.730536E-08	4.227850E+10
6	2.723653E+14	5.031653E-08	3.974836E+10
7	5.061180E+13	5.301388E-08	3.772596E+10
8	9.048326E+12	5.546897E-08	3.605619E+10
9	1.576200E+12	5.773034E-08	3.464382E+10
10	2.970200E+11	5.983260E-08	3.342659E+10

--------- Node = 16 ----- Radius = 9.800E-04----------

1	1.907013E+17	2.426457E-08	0.000000E+00
2	1.151657E+17	3.471774E-08	5.760742E+10
3	2.967965E+16	3.980550E-08	5.024431E+10
4	6.718704E+15	4.386735E-08	4.559199E+10
5	1.395400E+15	4.730536E-08	4.227850E+10
6	2.722627E+14	5.031653E-08	3.974836E+10
7	5.062442E+13	5.301388E-08	3.772596E+10
8	9.056896E+12	5.546897E-08	3.605619E+10
9	1.578883E+12	5.773034E-08	3.464382E+10
10	2.977862E+11	5.983260E-08	3.342659E+10

--------- Node = 17 ----- Radius = 9.900E-04----------

1	1.910678E+17	2.426457E-08	0.000000E+00
2	1.157624E+17	3.471774E-08	5.760742E+10
3	2.987072E+16	3.980550E-08	5.024431E+10
4	6.759842E+15	4.386735E-08	4.559199E+10
5	1.401990E+15	4.730536E-08	4.227850E+10
6	2.729704E+14	5.031653E-08	3.974836E+10
7	5.062448E+13	5.301388E-08	3.772596E+10
8	9.030537E+12	5.546897E-08	3.605619E+10
9	1.569303E+12	5.773034E-08	3.464382E+10
10	2.948855E+11	5.983260E-08	3.342659E+10

--------- Node = 18 ----- Radius = 9.925E-04----------

1	1.912319E+17	2.426457E-08	0.000000E+00
2	1.154131E+17	3.471774E-08	5.760742E+10
3	2.983335E+16	3.980550E-08	5.024431E+10
4	6.792335E+15	4.386735E-08	4.559199E+10
5	1.421386E+15	4.730536E-08	4.227850E+10
6	2.797620E+14	5.031653E-08	3.974836E+10
7	5.251273E+13	5.301388E-08	3.772596E+10
8	9.488197E+12	5.546897E-08	3.605619E+10
9	1.671186E+12	5.773034E-08	3.464382E+10
10	3.188721E+11	5.983260E-08	3.342659E+10

--------- Node = 19 ----- Radius = 9.950E-04----------

1	2.012295E+17	2.426457E-08	0.000000E+00

2	1.291832E+17	3.471774E-08	5.760742E+10
3	3.460673E+16	3.980550E-08	5.024431E+10
4	8.009758E+15	4.386735E-08	4.559199E+10
5	1.682402E+15	4.730536E-08	4.227850E+10
6	3.296526E+14	5.031653E-08	3.974836E+10
7	6.127661E+13	5.301388E-08	3.772596E+10
8	1.092718E+13	5.546897E-08	3.605619E+10
9	1.895029E+12	5.773034E-08	3.464382E+10
10	3.547876E+11	5.983260E-08	3.342659E+10

--------- Node = 20 ----- Radius = 9.987E-04----------

1	1.107092E+17	2.426457E-08	0.000000E+00
2	4.447289E+16	3.471774E-08	5.760742E+10
3	7.309087E+15	3.980550E-08	5.024431E+10
4	1.026271E+15	4.386735E-08	4.559199E+10
5	1.281519E+14	4.730536E-08	4.227850E+10
6	1.434489E+13	5.031653E-08	3.974836E+10
7	1.403750E+12	5.301388E-08	3.772596E+10
8	1.078410E+11	5.546897E-08	3.605619E+10
9	3.013661E+09	5.773034E-08	3.464382E+10
10	0.000000E+00	5.983260E-08	3.342659E+10

--------- Node = 21 ----- Radius = 1.000E-03----------

1	0.000000E+00	2.426457E-08	0.000000E+00
2	0.000000E+00	3.471774E-08	5.760742E+10
3	0.000000E+00	3.980550E-08	5.024431E+10
4	0.000000E+00	4.386735E-08	4.559199E+10
5	0.000000E+00	4.730536E-08	4.227850E+10
6	0.000000E+00	5.031653E-08	3.974836E+10
7	0.000000E+00	5.301388E-08	3.772596E+10
8	0.000000E+00	5.546897E-08	3.605619E+10
9	0.000000E+00	5.773034E-08	3.464382E+10
10	0.000000E+00	5.983260E-08	3.342659E+10

VITA

Daniel Lee DeForest is a Captain in the United States Air Force. He first joined the Air Force in 1980 soon after completing his undergraduate work. His undergraduate degree was a Bachelor of Science in Chemistry Cum Laude received from Baylor University in 1980. He subsequently received a Masters of Science degree in Nuclear Science from the Air Force Institute of Technology in 1982. From there he spent four years in the Technical Operations Division, Air Force Technical Applications Center, McClellan AFB, CA as a nuclear research officer. In 1986, he was selected to obtain his Ph.D. in nuclear engineering, and subsequently entered the program at Texas A&M University. While there, he presented a paper at the 1987 American Nuclear Society convention on comparisons of running the RELAP5 modeling code on various computers. In 1989, he was assigned to the Air Force Technical Applications Center Headquarters at Patrick AFB, FL where he is a branch chief in the Nuclear Technology Directorate over seven other physicists and nuclear engineers. His permanent address is:

50A E. Poinciana Dr.

Satellite Beach, FL 32937