21
31
06/17/80
26 copy to NTL

~~10/337~~

HEDL-TME 78-55

UC-79 b, d

MASTER

**A COMPUTER CODE
FOR THE ATOMISTIC SIMULATION
OF LATTICE DEFECTS AND DYNAMICS**

VOL. III

DO NOT MICROFILM
COVER

Hanford Engineering Development Laboratory

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

HANFORD ENGINEERING DEVELOPMENT LABORATORY
Operated by Westinghouse Hanford Company
P.O. Box 1970 Richland, WA 99352
A Subsidiary of Westinghouse Electric Corporation
Prepared for the U.S. Department of Energy
under Contract No. DE-AC14-76FF02170

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

SECTIONS 1 AND 2 MISSING

DO NOT MICROFILM COVER

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the U.S. DOE, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately-owned rights.

Printed in the United States of America
Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22161

NTIS Price Codes:
Printed: A10
Microfiche: A01

A COMPUTER CODE FOR THE ATOMISTIC SIMULATION OF LATTICE DEFECTS AND DYNAMICS VOL. III

Hanford Engineering Development Laboratory

J.O. Schiffgens
N.J. Graves

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

C.A. Oster
Pacific Northwest Laboratory

April 1980

NOTICE

PORTIONS OF THIS REPORT ARE ILLEGIBLE.

It has been reproduced from the best available copy to permit the broadest possible availability.

MN ONLY

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

HANFORD ENGINEERING DEVELOPMENT LABORATORY
Operated by Westinghouse Hanford Company
P.O. Box 1970 Richland, WA 99352
A Subsidiary of Westinghouse Electric Corporation
Prepared for the U.S. Department of Energy
under Contract No. DE-AC14-76FF02170

CONTENTS

	<u>Page</u>
VOLUME I. <u>INTRODUCTION</u>	
ACKNOWLEDGEMENTS	1
ABSTRACT	2
CODE ABSTRACT	3
VOLUME II. <u>USER'S GUIDE</u>	
FIGURES	xi
REFERENCES	xiii
1. <u>METHODS AND THEORY</u>	
1.1 <u>COMPUTER SIMULATION</u>	1-1
1.1.1 Review of Methods	1-1
1.1.1.1 Dynamical Method	1-2
1.1.1.2 Variational Method	1-2
1.1.1.3 Lattice Statics Method	1-3
1.1.1.4 Monte Carlo Method	1-3
1.1.1.5 Pattern Development Method	1-3
1.1.2 Methods Employed in COMENT	1-4
1.1.2.1 Objective	1-4
1.1.2.2 Scope	1-4
1.1.2.3 Model	1-5
1.2 <u>INTERATOMIC POTENTIAL</u>	1-8
1.2.1 Background	1-8
1.2.2 Formulation	1-9
1.3 <u>CALCULATIONAL PROCEDURES</u>	1-11

CONTENTS (Cont'd)

	<u>Page</u>
1.3.1 Interatomic Forces	1-11
1.3.1.1 Potential Function Forces	1-12
1.3.1.2 Pressure Forces	1-12
1.3.1.3 Hookeian Forces	1-13
1.3.1.4 Viscous Forces	1-13
1.3.2 Finite Difference Equations	1-14
1.3.2.1 Equations of Motion	1-14
1.3.2.2 Kinetic and Potential Energies	1-15
1.3.2.3 Dissipated and Stored Energies	1-16
1.4 <u>BOUNDARY CONDITIONS</u>	1-18
1.4.1 Discussion	1-18
1.4.1.1 Analysis of Defect Dynamics	1-18
1.4.1.2 Analysis of Static Defect Configurations	1-19
1.4.2 Determination of Special Forces	1-19
1.4.2.1 Conservative Forces	1-19
1.4.2.2 Non-Conservative Forces	1-21
2. <u>TECHNIQUES AND PROCEDURES</u>	2-1
2.1 <u>CODE CHARACTERISTICS</u>	2-1
2.1.1 Assembly Language Subroutines	2-1
2.1.2 Lattice Search Procedure	2-2
2.1.3 Crystal Lattices	2-4
2.1.3.1 Variable Volumes	2-25
2.1.3.2 Alloys	2-27
2.1.3.3 Temperature	2-30
2.1.4 Reliability	2-38
2.1.5 I/O Units	2-40
2.2 <u>INPUT VARIABLES</u>	2-42

CONTENTS (Cont'd)

	<u>Page</u>
2.2.1 DEFECT, TEMSET, ALLOY, RSTAPE, and CONT	2-42
2.2.2 Description of Other Variables	2-45
2.3 <u>SAMPLE CALCULATIONS</u>	2-68
2.3.1 A PKA in an Initially Ideal (fcc) Copper Lattice	2-68
2.3.2 A Gold Dumbbell Impurity Interstitial at Static Equilibrium in a Copper Lattice	2-93
2.3.3 A Copper Lattice Thermalization ($\sim 500^{\circ}\text{C}$)	2-108
2.3.3.1 A Lattice Expansion	2-134

VOLUME III. CODE DESCRIPTION

FIGURES	xi
REFERENCES	xiii

3. CALLING PROGRAM

3.1 <u>DISCUSSION</u>	3-1
3.1.1 Outline	3-1
3.1.2 Flow	3-2
3.2 <u>DEFINITIONS</u>	3-3
3.2.1 Variables and Constants	3-3
3.2.2 Units	3-49

4. SUBPROGRAMS

4.1 <u>FORTTRAN STATEMENT FUNCTIONS</u>	4-1
4.1.1 BOXNO (A,B,C)	4-1
4.1.2 RDIST (A,B,C)	4-1
4.1.3 TRANX (A,B), TRANY (A,B,C), TRANZ (A,B,C), and XTRAN (A,B,C), YTRAN (A,B,C), ZTRAN (B,C)	4-2

CONTENTS (Cont'd)

	<u>Page</u>
4.2 <u>FORTRAN EXTERNAL FUNCTIONS</u>	4-3
4.2.1 ERF (Y)	4-3
4.2.2 GAMIN (X,A)	4-4
4.2.3 INGINV (ALPHA, INGAM)	4-5
4.3 <u>FORTRAN SUBROUTINES</u>	4-6
4.3.1 ALOYORD (FA, FAA, FB, FBB, S, SO, INSERT, NPOT)	4-6
4.3.2 CRYSTL	4-9
4.3.3 DAMP (M, M2, M1, ENGH, DX1, DY1, FM, CCX, CCY, CCZ, IAFIX)	4-9
4.3.4 DASPKA (POLI, AXII, XG, YG, ZG, IPI, KEPKA, DT, DSP1, DSP2, DSP3, LPRNT1)	4-10
4.3.5 DFCT (IQ, DXD, DYD, DZD, LPD, DXE, DYE, DZE, LPE, IDV, N12S, IPS)	4-10
4.3.6 DIRECT (DBX, DBY, DBZ), RETURN (A1)	4-10
4.3.7 DSLOAN	4-10
4.3.8 DSLOIN	4-11
4.3.9 EXPAND (N1)	4-11
4.3.10 IDDEF (DR, ENG2, ECUT, IPI, UDCC, NB, LPRNT, LPRNT1, LPRNT2)	4-11
4.3.11 IEAF (NPOTA, NPOTB, NPOTAB, LPRNT2, ALLOY)	4-12
4.3.12 INCBOX	4-12
4.3.13 LABLE (S, FA, FB, NE)	4-13
4.3.14 MINILAT	4-13
4.3.15 POT1 (W, FO, F1, F2)	4-14
4.3.16 POT2 (W, FO, F1)	4-14
4.3.17 POT3 (W, FO, F1)	4-14

CONTENTS (Cont'd)

	<u>Page</u>
4.3.18 POT4 (W, FO, F1)	4-14
4.3.19 POT5 (W, FO, F1)	4-14
4.3.20 POT6 (W, FO, F1)	4-15
4.3.21 POT7 (W, FO, F1, NPOT)	4-15
4.3.22 SPEFORS (UDCC)	4-15
4.3.23 SQUEZ (IF)	4-16
4.3.24 THRMAL (GENVS)	4-16
4.3.24.1 Task One: Generation of the Velocity Distribution	4-16
4.3.24.2 Task Two: Statistical Analyses of Velocity and and Displacement Distributions	4-19
4.4 <u>COMPASS SUBPROGRAMS</u>	4-25
4.4.1 NEAF	4-25
4.4.2 Packing and Unpacking Real Numbers	4-29
4.4.2.1 Real Number Representation	4-29
4.4.2.2 PAC3 (A,X,Y,Z) and UNPAC3 (A,X,Y,Z)	4-31
4.4.2.3 PACF (A,P,H) and UNPACF (A,P,H)	4-32
4.4.3 Packing and Unpacking Integer Numbers	4-33
4.4.3.1 Integer Number Representation	4-33
4.4.3.2 INBIN (II,J1,ID), FILBIN (II,K,L,M,J) and MTBIN (II,K,L,M,J)	4-34
5. <u>COMENT LISTING</u>	5-1

FIGURES

<u>Figure</u>		<u>Page</u>
3.1.	A Flow Chart of the CALL Statements and Essential Elements of the CALLING PROGRAM.	3-4
4.1.	A Sketch of the Functional Relationship Between V_{ξ} and ξ .	4-18
4.2.	A Sketch of Gaussian and Maxwellian Probability Distribution Functions for $V_{\rho}=\sqrt{2}$.	4-20

REFERENCES

1. J. B. Gibson, A. N. Goland, M. Milgram, and G. H. Vineyard, Phys. Rev., 120, (1960) 1229.
2. P. C. Gehlen, A. R. Rosenfield, and G. T. Hahn, J. Appl. Phys., 39, (1968) 5264.
3. D. H. Tsai, R. Bullough, and R. C. Perrin, J. Phys. C., 3, (1970) 2022.
4. D. M. Schwartz, J. O. Schiffgens, D. G. Doran, G. R. Odette, and R. G. Ariyasu, in Computer Simulation for Materials Applications, eds., R. J. Arsenault, et al., NBS, 1, (1976) 75.
5. M. T. Robinson and J. M. Torrens, Phys. Rev. B, 9, (1974) 5008.
6. L. Tewordt, Phys. Rev., 109 (1958) 61.
7. R. A. Johnson and E. Brown, Phys. Rev., 127 (1962) 446.
8. R. M. J. Cotterill and M. Doyama, Phys. Rev., 145, (1966) 465.
9. H. Kanzaki, J. Phys. Chem. Solids, 2, (1957) 24.
10. P. A. Flinn and A. A. Maradudin, Annals of Phys., 18, (1962) 81.
11. J. W. Flocken and J. R. Hardy, Phys. Rev., 175 (1968) 919.
12. J. R. Beeler, U.S. Dept. of Comm. Report DC 59-12-83 (1959).
13. D. G. Besco, Gen. Elec. Co. Technical Report GEMP-644 (1967)/
14. A. I. Michaels and M. B. Ives, in Computer Simulation for Materials Applications, eds., R. J. Arsenault, et al., NBS 1, (1976) 133.
15. A. J. E. Foreman and M. J. Makin, Philosophical Magazine, 14 (1966) 911.
16. A. J. W. Moore, Philosophical Magazine, 16, (1967) 739.
17. R. C. Sanwald and J. J. Hren, Surface Sci., 7, (1967) 197.
18. L. H. Thomas, Proc. Cambridge Philosophical Soc., 23, (1927) 542.
19. E. Fermi, Zeitschrifter Phys., 48, (1928) 73.
20. J. O. Schiffgens, D. M. Schwartz, R. G. Ariyasu, and S. E. Cascadden, Rad. Eff. 39, (1979) 221.

REFERENCES (Cont'd)

21. J. O. Schiffgens and D. H. Ashton, J. Appl. Phys., 45 (1974) 1023.
22. C. Erginsoy, G. H. Vineyard, and A. Englert, Phys. Rev., 133, (1964) A595.
23. J. O. Schiffgens and K. E. Garrison, J. Appl. Phys., 43, (1972) 3240.
24. R. A. Johnson, Radiation Effects, 2, (1969) 1.
25. R. A. Johnson, Phys. Rev., 145, (1966) 423.
26. W. D. Wilson and R. A. Johnson, in Interatomic Potentials and Simulations of Lattice Defects, eds., P. C. Gehlin, et al., Plenum Press, (1972) 375.
27. R. A. Johnson, G. J. Dienes, and A. C. Damask, Acta Met., 12, (1964) 1215.
28. R. O. Jackson, J. P. Leighly, and D. R. Edwards, Phil. Mag., 25, (1972) 1169.
29. J. Aspeling, U. Gosele, and C. S. B. Piani, Rad. Eff., 38 (1978) 175.
30. H. R. Brager and F. A. Garner, J. Nucl. Mat., 73, (1978) 9.
31. T. H. Blewitt and R. R. Coltman, Acta Meta., 2, (1954) 549.
32. L. M. Howe and M. H. Rainville, J. Nucl. Mal., 68, (1977) 215.
33. M. L. Jenkins and M. Wilkens, Phil. Mag., 34, (1976) 1155.
34. J. O. Schiffgens and R. D. Bourquin, J. Nucl. Mat., 69 and 70 (1978) 790.
35. A. Larsen, ERAPE: A Computer Program for Classical Many-Body Problems in Radiation Damage, Brookhaven National Laboratory, BNL-7979 (1964).
36. A. T. DiBenedetto, The Structure and Properties of Materials, (McGraw-Hill Book Company, New York 1967) 176.
37. W. L. Bragg and E. J. Williams, Proc. Roy. Soc., A145 (1934) 699.
38. J. M. Cowley, Phys. Rev., 77 (1950) 669.
39. P. M. Morse, Thermal Physics (W. A. Benjamin, Inc., New York 1969), p. 276.

REFERENCES (Cont'd)

40. G. Moliere, A. Naturforsch, 2a (1947) 133.
41. A. Englert, H. Tompa, and R. Bullough, in Fundamental Aspects of Dislocation Theory, NBS Special Publ. 317 (1971) 273.
42. M. Born and J. E. Mayer, Z. Phys., 75 (1932) 1.
43. J. O. Schiffgens and H. L. Heinisch, HEDL Magnetic Fusion Energy Programs Progress Report, July-September 1977, HEDL-TME 77-110, p. 40.
44. Milton Abramowitz and Irene A. Stegun, Editors, Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables, National Bureau of Standards Applied Mathematics Series No. 55, U.S. Government Printing Office, Washington, DC (1965).
45. V. Mangulis, Handbook of Series for Scientists and Engineers, Academic Press (1965).
46. Brice Camahan, H. A. Luther and James O. Wilkes, Applied Numerical Methods, John Wiley and Sons, Inc. (1969) 562.
47. Gerand J. Hahn and Samuel S. Shapiro, Statistical Models in Engineering, John Wiley and Sons, Inc. (1967), p. 87.

3. CALLING PROGRAM

3.1 DISCUSSION

3.1.1 Outline

This document has been prepared to satisfy the need for a detailed, up-to-date description of a computer code that can be used to simulate phenomena on an atomistic level. COMENT was written in FORTRAN IV and COMPASS (CDC assembly language) to solve the classical equations of motion for a large number of atoms interacting according to a given force law, and to perform the desired ancillary analysis of the resulting data. COMENT is a dual-purpose code intended to describe static defect configurations as well as the detailed motion of atoms in a crystal lattice. It can be used to simulate the effect of temperature, impurities, and pre-existing defects on radiation-induced defect production mechanisms, defect migration, and defect stability.

COMENT consists of a CALLING PROGRAM and thirty-two user-defined subroutines. Fourteen subroutines (ALOYORD, DASPKA, DFCT, DSLOAN, DSLOIN, EXPAND, POT1, POT2, POT3, POT4, POT5, POT6, POT7, and THRMAL) are associated with the selection of program options; only a few of these are used in any given analysis. Seven of the other subroutines (CRYSTL, IEAF, INCBOX, LABEL, MINILAT, SPEFORS, and SQUEZ) are used to establish a variety of arrays and conditions required for each analysis; most of them are used once in a given calculation. The remaining eleven subroutines (DAMP, DIRECT, IDDEF, NEAF, INBIN, FILBIN, MTBIN, PAC3, UNPAC3, PACF and UNPACF) are used many times in each calculation; the last eight of these are used many times in each time step during the integration, and therefore, are written in COMPASS. The COMPASS subroutines are described in this volume in sufficient detail to permit easy conversion to some other assembly language or to FORTRAN.

The CALLING PROGRAM reads INPUT data, dictates the order of the calculation, provides linkage between subroutines, and writes the information that is generated. While some INPUT data are always read through NAMELIST, data generated during a previous calculation and stored on disk (i.e., TAPE2) may also be read as INPUT data. COMENT INPUT includes:

- 1) Problem identification data (ADAMP, ALLOY, CONT, DEFCLS, DEFECT, DEFINT, DRLIM, IELE, IREP, ISKIP, LINDEF, PTDEF, RELAX, RSEQUE, RSTAPE, TEMP, TEMSET);
- 2) Crystal lattice and related data (AROE, CLP, CM, CO, DERL, ERLI, FB, IC, IP, MEA, N2, S, SL, TEST, XIPS, YIPS, ZIPS);
- 3) Analytical procedure data (CL, CN, COO, CP, CQ, DTN, ECUT, EOUT, ESET, ES1, ES2, FTAPE, IAFIX, ITSS, LPRNT, LPRNT1, LPRNT2, NFRZ, RBCUT, RESET, RMN, RMX, VECT);
- 4) Interatomic potential and related data (AP, FH, FV, IDC, ID1, ID2, ID3, ID4, ID5, ID6, MA, MB, NPOTA, NPOTB, NPOTAB, NSF, NWDH, NWDL, RE, RM, RP, SR, ST, ZA, ZB); and
- 5) PKA or defect specification data (AZII, IMP, IPI, IMPUR, IPV, KEPKA, LPG, LPI, NE, POLI, XG, YG, ZG, XI, YI, ZI).

3.1.2 Flow

COMENT has many program options and may be readily adapted for addition of more. COMENT has the following general flow: it

- a) reads INPUT data from NAMELIST,
- b) generates interatomic force and energy tables,
- c) defines increments used in the box search procedure,
- d) defines lattice atom identification arrays,

- e) generates an ideal fcc lattice or reads previously generated fcc lattice atom identification and coordinate arrays (e.g., for a crystal lattice at temperature),
- f) establishes an impurity distribution in the crystal lattice (if requested),
- g) calculates pressure forces and Hookeian force constants for atoms at the lattice-continuum interface,
- h) superposes a Maxwellian velocity distribution on the lattice atoms (if requested),
- i) initiates the analysis of one or more cascades or the analysis of isolated or interacting defects,
- j) generates a spherical or cylindrical mini-lattice (if requested),
- k) integrates the equations of motion for atoms within the mini-lattice,
- l) analyzes the energy and position of all significantly displaced atoms each time step during the integration,
- m) writes OUTPUT data in desired form,
- n) enlarges the mini-lattice if and when necessary, and
- o) stores the final state of the lattice on disk for future use (if requested).

Figure 3.1 is a flow chart of the CALL statements in the CALLING PROGRAM.

3.2 DEFINITIONS

3.2.1 Variables and Constants

It is convenient to think of computer storage as comprising a number of registers, more often called words, into which a fixed number of digits can be stored. Storage words, each of which can be referenced by a unique address, have a dual function. They may be used to store the problem data and intermediate results, or they may contain the individual instructions of a program. In this section we are concerned with describing the former.

FIGURE 3.1. A Flow Chart of the CALL Statements and Essential Elements of the CALLING PROGRAM. (Page 1 of 4).

FIGURE 3.1. (Page 2 of 4).

FIGURE 3.1. (Page 3 of 4).

HEDL 7907-159.4

FIGURE 3.1. (Page 4 of 4).

Below are listed storage word labels and corresponding descriptions. The words may be designated constants, variables, logical variables, or arrays. Here, the terms "constant" or "array of constants" merely mean that the quantities in the registers do not change during a given calculation, although they may change in going from one calculation or analysis to another. In the same context, the quantities referred to as variables or arrays may change value during an analysis. The logical variables may or may not change value during an analysis. All INPUT labels are marked with an asterisk.

<u>LABEL</u>	<u>DESCRIPTION</u>
A	A dummy variable used to transfer packed atom coordinates to and from subroutines UNPAC3 and PAC3, and packed force constants to and from subroutines UNPACF and PACF.
*ADAMP	A logical variable used to cause damping of the interior atoms of the crystal lattice. When ADAMP is <u>TRUE</u> the kinetic energy of each of the interior atoms of the lattice is set to zero each time step.
*ALLOY	A logical variable used to specify a calculation as an analysis of some phenomenon in a binary alloy. When ALLOY is <u>TRUE</u> a face-centered-cubic crystal lattice is formed with the specified chemical composition and degree of $L1_2$ superlattice order (see subroutine ALOYORD).
*AP	A constant parameter used in interatomic potential functions (see subroutines IEAF, POT4, and POT6).
*AROE	A constant used to change the chopping error associated with coordinate packing into a random error. This constant is equal to half the maximum chopping error.

LABELDESCRIPTION

AX(MX, M1)	An array of constants representing the packed x-components of the pressure forces and Hookeian force constants for interface atoms with the particular surrounding atom configuration identified by indices MX and M1 (see subroutine SPEFORS).
AY(MX, M1)	An array of constants representing the packed y-components of the pressure forces and Hookeian force constants for interface atoms with the particular surrounding atom configuration identified by indices MX and M1 (see subroutine SPEFORS).
AZ(MX, M1)	An array of constants representing the packed z-components of the pressure forces and Hookeian force constants for interface atoms with the particular surrounding atom configuration identified by indices MX and M1 (see subroutine SPEFORS).
AZI	A variable containing an azimuthal angle relative to the x-axis in the unit cell coordinate system, calculated for each atom each time step, in order to either (1) describe the direction of motion of atoms during the analysis of dynamic atom configurations, or (2) describe the position of atoms during the analysis of static atom configurations (see subroutine DIRECT).
*AZII(I)	An array of constants specifying the initial azimuthal directions of primary knock-on atoms (see subroutine DASPKA)
CCX	A variable used in calculating the x-component of the viscous damping force applied to an atom.
CCY	A variable used in calculating the y-component of the viscous damping force applied to an atom.

LABELDESCRIPTION

CCZ	A variable used in calculating the z-component of the viscous damping force applied to an atom.
*CL	A constant specifying the minimum distance an atom must travel from its lattice site before its coordinates and energies are recorded for filming or any additional analyses of position are made.
*CLP	A constant specifying the crystal lattice parameter for the metal or alloy being analyzed. It should be equal to the value of the lattice parameter used in deriving the interatomic potential or potential set designated by NPOTA, or NPOTA, NPOTB, and NPOTAB (see subroutine IEAF).
*CM	A constant specifying the minimum distance an atom must travel from its lattice site before a check is made on its proximity to the surface of the mini-lattice or other lattice sites.
*CN	A constant specifying the minimum distance an atom must travel from its lattice site before its coordinates and energies are recorded for listing.
*CO	A constant specifying the distance from the mini-lattice surface within which a net atom displacement from its lattice site of greater than CM will cause the mini-lattice to be enlarged.
*CONT	A logical variable used to designate a calculation as a continuation of a previously initiated analysis. When CONT is <u>TRUE</u> a previously initiated analysis is to be continued (note: RSTAPE must be <u>TRUE</u>).

LABELDESCRIPTION

*CP	A constant specifying the maximum distance of one atom from another atom's lattice site such that the former is associated with the site of the latter and listed as such.
*CQ	A constant specifying the radius of a region surrounding the origin (i.e., XIPS, YIPS, and ZIPS) within which atom coordinates and energies are listed or recorded for filming.
CXD	A variable used to calculate the effect of viscous damping on the x-coordinate of an atom (see subroutine DAMP).
CXN	A variable used to calculate the effect of viscous damping on the x-coordinate of an atom (see subroutine DAMP).
CYD	A variable used to calculate the effect of viscous damping on the y-coordinate of an atom (see subroutine DAMP).
CYN	A variable used to calculate the effect of viscous damping on the y-coordinate of an atom (see subroutine DAMP).
CZD	A variable used to calculate the effect of viscous damping on the z-coordinate of an atom (see subroutine DAMP).
CZN	A variable used to calculate the effect of viscous damping on the z-coordinate of an atom (see subroutine DAMP).
*C00	A constant specifying a distance such that when an atom is further from its lattice site than this distance, an additional search procedure (centered on the site) is employed to identify the defect at the site (see subroutine IDDEF).

LABELDESCRIPTION

DB	A variable containing the distance of an atom from its lattice site, in angstroms. This variable is calculated for each atom each time step.
DBX	A variable containing the average change in the x-coordinate of an atom during the time interval $t - \Delta t$ to $t + \Delta t$. This quantity is calculated for each atom each time step (see sub-routine DIRECT).
DBY	A variable containing the average change in the y-coordinate of an atom during the time interval $t - \Delta t$ to $t + \Delta t$. This quantity is calculated for each atom each time step (see sub-routine DIRECT).
DBZ	A variable containing the average change in the z-coordinate of an atom during the time interval $t - \Delta t$ to $t + \Delta t$. This quantity is calculated for each atom each time step (see sub-routine DIRECT).
DDT	A variable containing the square of the time step, Δt .
*DEFCLS	A logical variable used to specify a calculation as an analysis of a defect cluster. When DEFCLS is <u>TRUE</u> the atom configuration and configurational energy of a defect cluster are analyzed (DEFECT must be <u>TRUE</u> when DEFCLS is <u>TRUE</u>).
*DEFECT	A logical variable used to specify a calculation as an analysis of a defect or defects (not a cascade). When DEFECT is <u>TRUE</u> the atom configuration and configurational energy of the defect (defects or clusters, isolated or interacting) are analyzed.

LABELDESCRIPTION

*DEFINT	A logical variable used to specify a calculation as an analysis of defect interactions. When DEFINT is <u>TRUE</u> the atom configuration and configurational energy of the interaction are analyzed (DEFECT must be <u>TRUE</u> when DEFINT is <u>TRUE</u>).
DEL	A constant used in calculating the kinetic energy of atoms in the lattice. It is one-half the inverse of DDT.
DELX	A variable containing the change in the x-coordinate of an atom during time interval Δt . This quantity is calculated for each atom each time step.
DELY	A variable containing the change in the y-coordinate of an atom during time interval Δt . This quantity is calculated for each atom each time step.
DELZ	A variable containing the change in the z-coordinate of an atom during time interval Δt . This quantity is calculated for each atom each time step.
*DERL(I)	An array of constants specifying incremental increases in the radius of the mini-lattice.
DIV	A constant (of value 2) defined in COMENT for transfer to subroutine NEAF.
DR	A variable containing the distance of an atom from its lattice site. This quantity is calculated for each atom each time step.
*DRLIM	A constant set to prevent the violation of boundary conditions during calculations at elevated temperatures ($TEMP \neq .0$). It is the maximum distance that an interface atom is permitted to be from its lattice site during thermal vibrations.

<u>LABEL</u>	<u>DESCRIPTION</u>
DR1	A variable used to identify the position of lattice sites. It is the distance of a lattice site from the center of coordinates.
DR2	A variable used to identify the position of lattice atoms. It is the distance of a lattice atom from the center of coordinates.
DSP1	A constant calculated and used to provide an initial PKA momentum. It is the pre-initial (i.e., $t_0 - \Delta t$) and imaginary x-displacement of the I^{th} PKA from its initial position in order to give it the momentum prescribed by KEPKA(I), POLI(I), and AZII(I) at time $t_0 = 0$ (see subroutine DASPKA).
DSP2	A constant calculated and used to provide an initial PKA momentum. It is the pre-initial (i.e., $t_0 - \Delta t$) and imaginary y-displacement of the I^{th} PKA from its initial position in order to give it the momentum prescribed by KEPKA(I), POLI(I), and AZII(I) at time $t_0 = 0$ (see subroutine DASPKA).
DSP3	A constant calculated and used to provide an initial PKA momentum. It is the pre-initial (i.e., $t_0 - \Delta t$) and imaginary z-displacement of the I^{th} PKA from its initial position in order to give it the momentum prescribed by KEPKA(I), POLI(I), and AZII(I) at time $t_0 = 0$ (see subroutine DASPKA).
DT	A variable containing the time increment (i.e., the length of the time step used during the integration). The length of the time step is allowed to increase periodically as the kinetic energy of the most energetic atom decreases (i.e., DT is set equal to DTN).

LABELDESCRIPTION

*DTN	The value of the time step increment as it is read in or calculated. During an analysis of dynamic atom configurations DTN is recalculated every time a mini-lattice is formed. DTN is recalculated only once during an analysis of static atom configurations (i.e., when DEFECT = <u>TRUE</u> and TEMP = 0, DTN is redefined at ITIME = ITSS + 50 to 5*DTN).
DTP	A variable containing the value of the time step increment calculated after the last time cycle in the integration procedure. This quantity, read from disk (i.e., TAPE2), was generated in a previous calculation.
DT0	A variable containing the value of the time step increment used during the last time cycle in the integration procedure. This quantity, read from disk (i.e., TAPE2), was generated in a previous calculation.
DX	A dummy variable usually used to contain the x-coordinate of an atom, in the COMENT reference system (see Section 2.1.3), relative to an arbitrarily chosen center of coordinates that may vary from analysis to analysis.
DX1	A variable usually used to contain the x-coordinate of a lattice site, in the unit cell reference system (see Section 2.1.3), relative to an arbitrarily chosen center of coordinates that may vary from analysis to analysis.
DX2	A variable usually used to contain the x-coordinate of an atom, in the unit cell reference system, relative to an arbitrarily chosen center of coordinates that may vary from analysis to analysis.

LABELDESCRIPTION

DX3	(Same as variable DX1)
DY	A dummy variable usually used to contain the y-coordinate of an atom, in the COMENT reference system, relative to an arbitrarily chosen center of coordinates that may vary from analysis to analysis.
DY1	A variable usually used to contain the y-coordinate of a lattice site, in the unit cell reference system, relative to an arbitrarily chosen center of coordinates that may vary from analysis to analysis.
DY2	A variable usually used to contain the y-coordinate of an atom, in the unit cell reference system, relative to an arbitrarily chosen center of coordinates that may vary from analysis to analysis.
DY3	(Same as variable DY1)
DZ	A dummy variable usually used to contain the z-coordinate of an atom, in the COMENT reference system, relative to an arbitrarily chosen center of coordinates that may vary from analysis to analysis.
DZ1	A variable usually used to contain the z-coordinate of a lattice site, in the unit cell reference system, relative to an arbitrarily chosen center of coordinates that may vary from analysis to analysis.
DZ2	A variable usually used to contain the z-coordinate of an atom, in the unit cell reference system, relative to an arbitrarily chosen center of coordinates that may vary from analysis to analysis.

LABELDESCRIPTION

DZ3	(Same as variable DZ1)
D1	A dummy variable used to transfer packed atom coordinates from disk storage to COMMENT. The coordinates define the atom position at time t .
D2	A dummy variable used to transfer packed atom coordinates from disk storage to COMMENT. The coordinates define the atom position at time $t - \Delta t$.
D3	A dummy variable used to transfer packed atom coordinates from disk storage to COMMENT. The coordinates define the lattice site.
EA(I)	An array of constants specifying the interatomic force per interatomic distance for interactions between A (host) atoms. This array is generated in IEAF from the potential function specified by NPOTA.
EAB(I)	An array of constants specifying the interatomic force per interatomic distance for interactions between A (host) and B (impurity) atoms. This array is generated in IEAF from the potential function specified by NPOTAB.
EB(I)	An array of constants specifying the interatomic force per interatomic distance for interactions between B (impurity) atoms. This array is generated in IEAF from the potential function specified by NPOTB.
*ECUT	A constant set to limit the quantity of listed OUTPUT. Energy and directional information is tabulated and analyzed for all atoms with kinetic energies greater than ECUT.

LABELDESCRIPTION

EEA(I)	An array of constants specifying the interatomic potential energy for interactions between A (host) atoms. This array is generated in IEAF from the potential function specified by NPOTA.
EEAB(I)	An array of constants specifying the interatomic potential energy for interactions between A (host) and B (impurity) atoms. This array is generated in IEAF from the potential function specified by NPOTAB.
EEB(I)	An array of constants specifying the interatomic potential energy for interactions between B (impurity) atoms. This array is generated in IEAF from the potential function specified by NPOTB.
ELOS	A variable, evaluated each time step, containing an estimate of the cumulative energy lost by the lattice during an analysis.
ENGH	A variable, evaluated for each interface atom each time step, containing an estimate of the energy stored by the atom in the surrounding continuum.
ENGKA	A variable, evaluated each time step, containing the total kinetic energy of all the atoms in the lattice at each point in time during an analysis.
ENGPI(ID)	An array containing the configurational energy (essentially, the total potential energy) of each atom in the lattice.
ENGPO	A variable, evaluated each time step, containing the total configurational (potential) energy of all the atoms in the lattice at each point in time during an analysis.

LABELDESCRIPTION

ENGSP	A variable, evaluated each time step, containing an estimate of the total energy stored in the interface atom region surrounding the lattice at each point in time during an analysis.
ENG2	A variable, evaluated for each atom each time step, containing the kinetic energy of the atom.
ENG2M	A variable containing the kinetic energy of the most energetic atom in the lattice.
ENG3	A variable, evaluated for each atom each time step, containing an estimate of the energy lost by the atom due to the application of viscous damping forces.
*EOUT	A constant set to limit the amount of energy (kinetic plus potential, ET2) an atom may have when near the lattice-continuum interface and still be considered within the lattice. When an atom near the interface has energy ET2 greater than EOUT it is removed from the lattice.
ERL	A dummy variable used to contain the radii of various regions in the mini-lattice during mini-lattice formation (see subroutine MINILAT).
*ERLI	A variable specifying the radius of a cylindrical or spherical mini-lattice. During the analysis of a cascade the radius of the mini-lattice increases as the cascade expands (see subroutine MINILAT).
ERLII	A variable equal to ERLI, expressed in angstroms, and listed in the OUTPUT. Although ERLI is entered through NAMELIST in angstroms, it is immediately converted to COMENT units for subsequent use.

<u>LABEL</u>	<u>DESCRIPTION</u>
ERLI1	A variable containing the last radius of the mini-lattice used in a calculation. This quantity, read from disk (i.e., TAPE2), was generated in a previous calculation.
*ESET	A constant set to define the end of the first phase in the development of a displacement cascade in terms of the kinetic energy (ENG2M) of the most energetic atom (MEA). When ENG2M falls below ESET, the size of the mini-lattice is increased for the last time.
*ES1	A constant set to warn of possible boundary effect distortions. When an atom near the lattice-continuum interface receives energy ET2 greater than ES1 but less than ES2 a warning is printed.
*ES2	A constant set to warn of possible boundary effect distortions. When an atom near the lattice-continuum interface receives energy ET2 greater than ES2 but less than EOUT, a message is printed and the calculation is terminated.
ET2	A variable, evaluated for each atom each time step, containing the sum of the kinetic and potential energies of the atom.
E1	A constant calculated in subroutine IEAF and used in subroutine NEAF for locating values of interatomic force and potential in the EA(I), EB(I), EAB(I), EEA(I), EEB(I), and EEAB(I) arrays.
E2	A constant calculated in subroutine IEAF and used in subroutine NEAF for locating values of interatomic force and potential in the EA(I), EB(I), EAB(I), EEA(I), EEB(I), and EEAB(I) arrays.
FA	A constant calculated when generating an alloy. This quantity is the fraction of A (host) atoms in the lattice.

LABELDESCRIPTION

FAA	A constant that reflects the degree of order in a nominal A_3B , $L1_2$ superlattice. This quantity is the fraction of A atoms on A lattice sites (see subroutine ALOYORD).
*FB	A constant used to specify the composition of an impure metal or alloy. This quantity is the fraction of B (impurity) atoms in the lattice.
FBB	A constant that reflects the degree of order in a nominal A_3B , $L1_2$ superlattice. This quantity is the fraction of B atoms on B lattice sites (see subroutine ALOYORD).
*FH	A constant used to modify the Hookeian force constants as calculated in the subroutine SPEFORS.
FM	A constant used to take into account the effect of different atomic masses when analyzing an impure metal or alloy.
FMA	A variable containing the calculated value of FM for A (host) atoms.
FMB	A variable containing the calculated value of FM for B (impurity) atoms.
*FTAPE	A logical variable used to cause data to be stored for filming. When FTAPE is <u>TRUE</u> the effective state of the lattice at the end of each time step is recorded on disk so that a filmed account of the processes being studied can be made.
*FV	A constant used to modify the viscous force constants as calculated in subroutine SPEFORS.

LABELDESCRIPTION

GENVS	A logical variable used to determine how subroutine THRMAL is to be employed. When GENVS is <u>TRUE</u> , THRMAL superposes a Maxwellian velocity distribution on the lattice atoms, then analyzes the velocity and displacement distributions. When GENVS is <u>FALSE</u> , only the analysis is performed.
HXC	A variable containing the effective x-component of a Hookeian force constant (averaged over the displacement range 0 to ST*NSF) for an interior atom with neighbors fixed at ideal lattice sites (see subroutine SPEFORS).
HYC	A variable containing the effective y-component of a Hookeian force constant (averaged over the displacement range 0 to ST*NSF) for an interior atom with neighbors fixed at ideal lattice sites (see subroutine SPEFORS).
HZC	A variable containing the effective z-component of a Hookeian force constant (averaged over the displacement range 0 to ST*NSF) for an interior atom with neighbors fixed at ideal lattice sites (see subroutine SPEFORS).
I	A dummy variable, usually used for indexing or counting.
*IAFIX	A logical variable used to specify the behavior of interface atoms. When IAFIX is <u>TRUE</u> , lattice-continuum interface atoms are not permitted to move (i.e., relax against the appropriate special forces).
IB	A dummy variable used as an array index.
IBOX(J)	An array containing packed identification numbers. Each element of this array contains the identification numbers of atoms located within the region (i.e., "box") identified by J.

LABELDESCRIPTION

*IC	A dummy variable used as a counting index. During an analysis this quantity reveals the number of times the size of the mini-lattice has been increased. Initially IC specifies the first element of the DERL array to be used during mini-lattice expansion.
IC1	A variable containing the value of IC at the termination of a calculation. This quantity, read from disk (i.e., TAPE2), was generated in a previous calculation.
ID	A variable used for indexing and counting. This variable always refers to an atom identification number for either the entire-lattice or a mini-lattice.
*IDC	A variable used to identify an interior atom for the purpose of determining effective Hookeian force constants for a given potential and setting the augmenting Hookeian force constants for interface atoms (see subroutine SPEFORS).
IDENT(ID)	An array containing packed atom identification data. Each element of the IDENT array contains the entire-lattice atom identification number (K), the element type number (L), the boundary type number (M), and the "box" identification number (J), for the atom with the current, perhaps mini-lattice, atom identification number ID (which may be different from the entire-lattice atom identification number).
IDL	A variable used to transfer packed atom identification data from disk storage to COMENT.

<u>LABEL</u>	<u>DESCRIPTION</u>
IDT	A variable used to identify the other atom when calculating its contribution to the configurational energy of, and resultant force on, atom ID.
IDTFL	A variable used to contain an alphanumeric film identification label of up to 80 characters.
*ID1	A variable used to specify an interface atom identification number. The data generated in the course of calculating special forces and force constants for this atom are printed (see subroutine SPEFORS).
*ID2	(Same as ID1)
*ID3	(Same as ID1)
*ID4	(Same as ID1)
*ID5	(Same as ID1)
*ID6	(Same as ID1)
*IELE	A constant used to specify the number of chemical elements in the metal or alloy being analyzed. No more than two elements can be considered; when IELE = 1 a pure metal is specified, and when IELE = 2 an impure metal or binary alloy is specified.
II	A dummy variable generally used to transfer packed atom identification data to and from subroutines MTBIN, INBIN, and FILBIN.

LABELDESCRIPTION

III	(Same as II)
IK(MX,J1)	Arrays of constants specifying increments in the "box" identification numbers that are used in the box search procedure. Adding these increments to J, the box identification number of atom ID, yields the box identification numbers of all surrounding boxes containing interacting neighbors (see subroutine INCBOX).
*IMP	A constant specifying the number of impurities being inserted into the metal lattice (used only when IELE = 2 and ALLOY is <u>FALSE</u>).
*IMPUR(K)	An array of constants specifying the entire-lattice atom identification numbers of atoms that are to be impurities during the analysis.
INIB	A constant (of value 5) defined in COMENT for transfer to subroutine NEAF.
INSERT	A logical variable used to determine how subroutine ALOYORD is to be employed. When INSERT is <u>TRUE</u> , ALOYORD generates an A_3B $L1_2$ superlattice of specified stoichiometry and degree of long- and short-range order. When INSERT is <u>FALSE</u> , only the degrees of long- and short-range order are analyzed.
*IP	A variable used to identify a lattice site that is to serve as the origin of a unit cell coordinate system. It is the identification number of the lattice site. This number specifies the origin only when XIPS, YIPS, and ZIPS are zero.

<u>LABEL</u>	<u>DESCRIPTION</u>
*IPI(ID)	An array of constants specifying the atom identification numbers of either the atoms that are to serve as PKAs (when DEFECT is <u>FALSE</u>) or the lattice sites that are to serve as interstitial sites (when DEFECT is <u>TRUE</u>).
IPS	A dummy variable containing the value of IP.
IPV(ID)	An array of constants specifying the atom identification numbers of lattice sites that are to serve as vacancies.
IPI	A dummy variable containing the value of IP. This quantity, read from disk (i.e., TAPE2), was generated in a previous calculation.
IQ	A dummy variable containing the identification number stored in an element of IPI(I).
*IREP(ID)	An array of constants specifying atom identification numbers that identify prospective members of replacement sequences. When the elements of this array are specified, all coordinate and energy data of the atoms involved are listed each time step.
IS	A dummy variable used for indexing and counting.
*ISKIP(ID)	An array of constants specifying atom identification numbers. The elements of this array are intended to specify those atoms near the lattice-continuum interface that are likely to be ejected from the lattice region during an analysis.
ISTOP	A variable used as a switch to "turn off" a calculation when the boundary conditions are violated.

LABELDESCRIPTION

ITIME	A variable containing the number of time steps taken up to any point during the integration.
ITS	A variable containing the time step (i.e., the value of ITIME) at which an analysis is initiated or a new mini-lattice is formed (or enlarged).
*ITSS	A variable containing the value of ITS (i.e., the value of ITIME) at which an analysis is initiated.
ITO	A dummy variable used as a switch to check the time required to integrate over one time step in the integration procedure.
I1	A dummy variable usually used for indexing or counting.
I2	A dummy variable usually used for indexing or counting.
I3	A dummy variable usually used for indexing or counting.
J	A variable used for indexing and counting. This variable always refers to a box identification number.
J1	A dummy variable usually used in the process of packing the IBOX(J) array (see Section INBIN).
J2	A dummy variable usually used in the box search procedure.
J3	A dummy variable usually used to transfer the packed content of the element IBOX(J2) from COMENT to NEAF.
KEA	A variable containing the entire-lattice atom identification number for the most energetic atom in the lattice.

<u>LABEL</u>	<u>DESCRIPTION</u>
*KEPKA(ID)	An array of constants specifying the initial kinetic energies of all primary knock-on atoms.
KO	A dummy variable used as a switch signal.
L	A variable used to specify the chemical identity of an atom. When L = 1 the atom is a host atom (A), when L = 2 the atom is an impurity atom (B).
LDO(MX)	An array of constants specifying the total number of boxes to be searched for each boundary type (specified by MX).
LFIRST	A logical variable defined in COMENT for transfer to subroutine MINILAT. When LFIRST is <u>TRUE</u> , the mini-lattice formed at the beginning of a continuation calculation has the same origin as that used at the end of the previous calculation (which is part of the same analysis).
*LINDEF	A logical variable used to specify a calculation as an analysis of a line defect (i.e., a dislocation). When LINDEF is <u>TRUE</u> the atom configuration and configurational energy of the dislocation are analyzed (DEFECT must be <u>TRUE</u> when LINDEF is <u>TRUE</u>).
*LPG(ID)	An array of constants specifying the chemical identification numbers (L's) of one member of each dumbbell interstitial inserted for analysis.
*LPI(ID)	An array of constants specifying the chemical identification numbers (L's) of the other member of each dumbbell interstitial inserted for analysis [see Section LPG(ID)].

LABELDESCRIPTION

LPOT	A logical variable used to determine whether or not special forces need to be calculated for a given continuation calculation. When LPOT is <u>TRUE</u> the "restart" calculation uses the same potential set as that used in the previous calculation, and the special forces from that calculation (stored on disk) are used.
*LPRNT	A logical variable used as a switch to control the amount of printed OUTPUT. LPRNT changes several times during a calculation.
*LPRNT1	A logical variable used as a switch to control the amount of additional OUTPUT, over and above that dictated by LPRNT. LPRNT1 does not change during a calculation.
*LPRNT2	A logical variable used as a switch to control the amount of additional OUTPUT, over and above that dictated by LPRNT and LPRNT1. LPRNT2 does not change during a calculation.
LT	A variable used to specify the chemical identity of the atom IDT that interacts with atom ID (for which the interatomic forces and energies are being summed).
M	A variable used to specify the boundary type of an atom. The value of this variable uniquely identifies the symmetry of atoms (or missing atoms) surrounding the given interface atoms.
*MA	A variable containing the mass of A (host) atoms.
*MB	A variable containing the mass of B (impurity) atoms.
*MEA	A variable containing the identification number of the most energetic atom in the entire-lattice or mini-lattice.

<u>LABEL</u>	<u>DESCRIPTION</u>
MEA1	A variable containing the identification number of the most energetic atom in the lattice. It is used to transfer the value of MEA from one time step to another.
MEE	A variable containing the identification number of the most energetic atom in the lattice. This quantity may be stored for use in a continuation calculation.
ML	A dummy variable used to determine the character of the search procedure. When ML = 1 a forward search procedure is called (i.e., only atoms for which IDT is greater than ID are considered). When ML = 0 a total search procedure is called (i.e., all values of IDT are considered).
MPOTA	A variable that contains the potential identification number of A-A (host-host) atom interactions (as does NPOTA). This quantity, read from disk (i.e., TAPE2), was generated in a previous calculation.
MPOTAB	A variable that contains the potential identification number of A-B (host-impurity) atom interactions (as does NPOTAB). This quantity, read from disk (i.e., TAPE2), was generated in a previous calculation.
MPOTB	A variable that contains the potential identification number of B-B (impurity-impurity) atom interactions (as does NPOTB). This quantity, read from disk (i.e., TAPE2), was generated in a previous calculation.
MX	A variable that contains the search array index for an interface atom. The search array index determines the appropriate search pattern and corresponding IK(MX,J1) box number increment array.

LABELDESCRIPTION

MXKPKA	A variable that contains the identification number of the most energetic PKA introduced into the lattice. This quantity is used in the initial calculation of DTN.
M1	A dummy variable usually used to contain the minor interface atom special force identification index.
M2	A dummy variable usually used to contain the major interface atom special force identification index.
M3	A dummy variable usually used in calculating MX.
M4	A dummy variable usually used to transfer elements of the LDO(MX) array [i.e., the lengths of given IK(MX,J1) arrays] within the CALLING PROGRAM.
N	A variable used to contain the total number of atoms in the lattice (or mini-lattice).
NAT	A variable containing the total amount of atoms making up the previous mini-lattice [i.e., ITIME < ITS].
NA	A variable used to contain the total number of atoms in the interior of the lattice (or mini-lattice). This quantity contains the total number of atoms with a complete set of interacting neighbors at any time during a calculation.
NB	A variable used to contain the number of boxes required to encompass the entire-lattice [i.e., the length of the IBOX(J) array].
*NE	A variable containing the number of PKAs (when DEFECT is <u>FALSE</u>) or the number of point defects (when DEFECT is <u>TRUE</u>).

LABELDESCRIPTION

*NFRZ	A logical variable used to specify whether or not interface atoms are to be damped. When NFRZ is <u>FALSE</u> interface atoms are damped, and when NFRZ is <u>TRUE</u> no damping forces are applied (see subroutine DAMP).
NI	A variable containing the identification number of the first atom in a continuous set that includes all atom boundary types. This quantity is defined in COMENT for use in subroutine SPEFORS.
NID	A dummy variable used in subroutine MINILAT. This variable is used in summing the atoms in the mini-lattice being generated.
NL	A variable containing the identification number of the last atom in the continuous series ranging from 1 to NL. Inter-atomic forces are summed and stored for all atoms in this series. Atoms with identification numbers between NL + 1 (i.e., N12) and N are not permitted to move.
NLS	A variable containing the initial value of NL.
NP	A dummy variable used to contain the value of NPOTA, NPOTB, and NPOTAB when selecting elements of some arrays.
*NPOTA	A variable obtained from NAMELIST that contains the potential identification number of A-A (host-host) atom interactions.
*NPOTAB	A variable obtained from NAMELIST that contains the potential identification number of A-B (host-impurity) atom interactions.
*NPOTB	A variable obtained from NAMELIST that contains the potential identification number of B-B (impurity-impurity) atom interactions.

LABELDESCRIPTION

NR	A dummy variable used in the subroutine MINILAT. This variable is used to contain the number of atoms in various regions of the lattice during mini-lattice formation.
NS	A variable in which the total number of atoms in the entire-lattice is stored for use throughout the calculation.
*NSF	A constant used when determining the Hookeian force constants for interface atoms. This quantity is the number of points at which the restoring force is calculated when determining the special forces (see subroutine SPEFORS).
*NWDH	A constant containing an identification number that determines the number of (112) interface end planes, nearest the interior of the lattice, whose atoms are to experience weak damping. This number refers to the planes containing atoms with high z-coordinates
*NWDL	A constant containing an identification number that determines the number of (112) interface end planes, nearest the interior of the lattice, whose atoms are to experience weak damping. This number refers to the planes containing atoms with low z-coordinates.
N1	A constant containing the number of atoms in the entire-lattice.
N12	A variable containing the identification number of the first atom in the continuous series ranging from $N12 = NL + 1$ to N . Interatomic forces are not summed for the atoms in this series, since these atoms are not permitted to move.
N12S	A constant containing the value of N12 for the entire-lattice.

LABELDESCRIPTION

*N2	A constant containing the number of atoms in the entire-lattice that are permitted to move (i.e., $N2 = NL$ when $SL = 5$).
PE0	A dummy variable used to contain the total configurational energy of the lattice and interface regions before and after a new mini-lattice is formed.
PE01	A variable containing the value of PE0. This quantity, read from disk (i.e., TAPE2), was generated in a previous calculation.
PEPA	A constant containing the configurational energy of an atom, with a full complement of neighbors, in a static ideal lattice.
POL	A variable containing a polar angle, calculated for each atom each time step, in order to either (1) describe the direction of motion of atoms during the analysis of dynamic atom configurations, or (2) describe the position of atoms during the analysis of static atom configurations (see subroutine DIRECT).
*POLI(I)	An array of constants specifying the initial polar directions of primary knock-on atoms (see subroutine DASPKA).
*PTDEF	A logical variable used to specify a calculation as an analysis of point defects. When PTDEF is <u>TRUE</u> the atom configuration and configurational energy of the point defects are analyzed (DEFECT must be <u>TRUE</u> when PTDEF is <u>TRUE</u>).
PXC	A constant used when calculating special forces. This is the x-component of the residual restoring force on an interior atom at its lattice site in an ideal lattice. Ideally, it should be zero (i.e., no force imbalance).

LABELDESCRIPTION

PYC	A constant used when calculating special forces. This is the y-component of the residual restoring force on an interior atom at its lattice site in an ideal lattice. Ideally, it should be zero (i.e., no force imbalance).
PZC	A constant used when calculating special forces. This is the z-component of the residual restoring force on an interior atom at its lattice site in an ideal lattice. Ideally, it should be zero (i.e., no force imbalance).
*RBCUT	A constant used to define the interior region of the entire-lattice. Atoms with R11 greater than RBCUT are regularly monitored to determine if they violate the interface boundary conditions.
RCUT	A constant used to alter the boundary type identification number during a calculation. When an interior atom moves outside the region defined by RCUT (i.e., such that R11 is greater than RCUT) and RESET = 3, its boundary type identification number changes.
*RE	A constant parameter used in an interatomic potential function (see subroutines IEAF and POT3).
*RELAX	A logical variable used to determine how subroutine DSLOAN repositions atoms on the (112) interface end planes at the end of each time step. When RELAX is <u>FALSE</u> , DSLOAN analyzes the strain energy as a function of distance from the dislocation line.
*RESET	A variable used as a switch to cause the mini-lattice to expand. When RESET = 0, the size of the mini-lattice does not change. When RESET = 1 or 2, the size of the mini-lattice is increased,

LABELDESCRIPTION

ERLI = ERLI + DERL (IC + 1). When RESET = 3 the size of the mini-lattice is increased for the last time.

RESET1	A variable that contains the value of RESET after the last time cycle in the integration procedure. This quantity, read from disk (i.e., TAPE2), was generated in a previous calculation.
*RM	A constant parameter used in an interatomic potential function (see subroutines IEAF and POT3).
RMAX	A constant containing a value of RMX(NPOTA) (see subroutine IEAF).
RMAX2	A constant containing a value of RMAX squared.
RMIN	A constant containing the value of RMN(NPOTA) (see subroutine IEAF).
*RMN(NPOT)	An array of constants specifying the minimum value of the interatomic distance for each of the interatomic potentials (identified by NPOTA, NPOTB, or NPOTAB).
*RMX(NPOT)	An array of constants specifying the maximum value of the interatomic distance for each of the interatomic potentials (identified by NPOTA, NPOTB, or NPOTAB). This is generally the distance at which the interatomic potential and force go smoothly to zero.
*RP	A constant parameter used in an interatomic potential function (see subroutines IEAF, POT4, and POT6).

LABELDESCRIPTION

*RSEQUE	A logical variable used to indicate that certain atoms in the lattice will be involved in linear replacement sequences. When RSEQUE is <u>TRUE</u> a specific replacement sequence is anticipated.
*RSTAPE	A logical variable used to indicate that the calculation will begin with previously generated lattice data. When RSTAPE is <u>TRUE</u> INPUT data are read from TAPE2 (note: CONT may or may not be <u>TRUE</u>).
R11	A variable containing the distance of an atom from the z-axis (in the COMENT coordinate system) that lies at the center of the entire-lattice.
*S	A variable containing a long-range order parameter. This parameter reflects the extent to which the alloy lattice is an ordered $L1_2$ (i.e., A_3B type) superlattice.
SFM	A constant containing a Hookeian force constant. This force constant corresponds to the smallest restoring force constant correction required for a displaced interface atom to make it behave like an interior atom.
*SL	A constant used to determine the shape and orientation of the mini-lattice (for further discussion see section 2.2.1).
S0	A constant used to contain the initial value of the long-range order parameter.
SPE	A variable containing the increase in the potential energy stored in an ideal interface region when the mini-lattice is formed or its size is increased.

<u>LABEL</u>	<u>DESCRIPTION</u>
*SR	A constant containing a parameter used in an interatomic potential function (see subroutines IEAF and POT1).
*ST	A constant containing the distance between points at which the restoring forces are calculated when determining the Hookeian force constants for interface atoms (see subroutine SPEFORS).
STEMP	A constant used to contain the simulated temperature of the lattice. This quantity, read from disk (i.e., TAPE2), was generated in a previous calculation.
TB(ID)	An array of constants specifying the packed x-, y-, and z-coordinates of all lattice sites (see subroutine CRYSTL). The coordinates of the lattice sites of all atoms are maintained throughout an analysis.
TBX1	A variable used to contain the x-coordinate of a lattice site relative to the COMENT entire-lattice center of coordinates. The quantity in TBX1 comes from unpacking a TB(ID) array element.
TBY1	A variable used to contain the y-coordinate of a lattice site relative to the COMENT entire-lattice center of coordinates. The quantity in TBY1 comes from unpacking a TB(ID) array element.
TBZ1	A variable used to contain the z-coordinate of a lattice site relative to the COMENT entire-lattice center of coordinates. The quantity in TBZ1 comes from unpacking a TB(ID) array element.

LABELDESCRIPTION

TE	A variable evaluated each time step, containing the total energy (i.e., the sum of kinetic, potential, stored, and lost energies) of all the atoms in the lattice.
TEDIF	A variable evaluated each time step, containing the deviation of the total energy at any time, t, from the value of the total energy immediately after the last mini-lattice was formed (or after the entire-lattice was created if no mini-lattice was formed).
*TEMP	A constant used to contain the simulated temperature of the lattice (see STEMP).
*TEMSET	A logical variable used to specify whether or not a lattice at temperature is to be formed. When TEMSET is <u>TRUE</u> an appropriate Maxwellian velocity distribution is superposed on the lattice atoms to simulate the atom vibrations associated with the specified temperature.
TE0	A variable evaluated when the entire-lattice and mini-lattice are formed, and after each mini-lattice expansion, containing the total energy of all atoms in the mini-lattice.
TE01	A constant used to contain the total energy of all the atoms in the lattice (see TE0). This quantity, read from disk (i.e., TAPE2), was generated in a previous calculation.
*TEST	A constant used to distinguish displacements due to uncertainties from those due to simulated physical disturbances. This quantity is usually set to three times the displacement that would normally be caused by PXC, PYC, or PZC (which turns out to be about twice the coordinate packing error).

LABELDESCRIPTION

TIME	A variable used to contain the simulated time elapsed during the unfolding of the phenomenon being analyzed.
TT1	A variable containing the actual clock computation time required to complete one integration cycle.
TX1	(Same as TBX1)
TY1	(Same as TBY1)
TZ1	(Same as TBZ1)
T1	A dummy variable used in calculating TT1 that contains the clock time in seconds at the start of the integration loop.
T2	A dummy variable used in calculating TT1 that contains the clock time in seconds at the end of the integration loop.
UD	A constant containing the COMENT <u>unit distance</u> to <u>meters</u> multiplicative conversion factor (m/ud).
UDCC	A constant containing the COMENT <u>unit distance</u> to <u>angstroms</u> multiplicative conversion factor ($\text{\AA}/\text{ud}$).
UE	A constant containing the COMENT <u>unit energy</u> to <u>joules</u> multiplicative conversion factor (joules/ue). Note that in COMENT the unit of energy is electron volts (eV).
UF	A constant containing the COMENT <u>unit force</u> to <u>newtons</u> multiplicative conversion factor (N/uf).

<u>LABEL</u>	<u>DESCRIPTION</u>
UHC	A constant containing the COMENT <u>unit Hookeian force constant</u> to <u>newtons per meter</u> multiplicative conversion factor $[(N/m)/(uf/ud)]$.
UM	A constant containing the COMENT <u>unit mass</u> to <u>kilogram mass</u> multiplicative conversion factor (kgm/um).
UT	A constant containing the COMENT <u>unit time</u> to <u>seconds</u> multiplicative conversion factor (sec/ut).
UV	A constant containing the COMENT <u>unit velocity</u> to <u>meters per second</u> multiplicative conversion factor $[(m/s)/uv]$.
UVC	A constant containing the COMENT <u>unit viscous force constant</u> to <u>newtons per meter per second</u> multiplicative conversion factor $[(N/m/s)/(uf/uv)]$.
VCX	A constant containing the x-component of the viscous force constant.
VCY	A constant containing the y-component of the viscous force constant.
VCZ	A constant containing the z-component of the viscous force constant.
*VECT	A constant containing the maximum distance the most energetic atom travels during a single time step.
VX(ID)	An array containing the packed, x-, y-, and z-coordinates of all atoms in the lattice at time $t-\Delta t$. These "past" positions

LABELDESCRIPTION

are required to calculate the velocities and changes in momentum of the atoms at any time during an analysis.

VXN	A variable containing the x-coordinate of an atom that has been repositioned to conserve momentum after a change in time step length.
VX0	A variable used to contain the x-coordinate of an atom relative to the COMENT entire-lattice center of coordinates at time $t-\Delta t$. The quantity in VX0 comes from unpacking a VX(ID) array element.
VX11	(Same as VX0)
VYN	A variable containing the y-coordinate of an atom that has been repositioned to conserve momentum after a change in time step length.
VY0	A variable used to contain the y-coordinate of an atom relative to the COMENT entire-lattice center of coordinates at time $t-\Delta t$. The quantity in VY0 comes from unpacking a VX(ID) array element.
VY11	(Same as VY0)
VZN	A variable containing the z-coordinate of an atom that has been repositioned to conserve momentum after a change in time step length.
VZ0	A variable used to contain the z-coordinate of an atom relative to the COMENT entire-lattice center of coordinates at time $t-\Delta t$. The quantity in VZ0 comes from unpacking a VX(ID) array element.
VZ11	(Same as VZ0)

LABELDESCRIPTION

X(ID)	An array containing the packed x-, y-, and z-coordinates of all atoms in the lattice at time t. These "present" positions are required to calculate the velocities, changes in momentum, and restoring forces on the atoms at any time during an analysis.
XB	A variable used to contain the x-coordinate of an atom relative to the z-axis (in the COMENT reference system) that lies along the center of the entire-lattice.
XDM	A dummy variable.
XEA	A variable used to contain the x-coordinate of the most energetic atom in the mini-lattice, relative to the COMENT entire-lattice center of coordinates.
XEA1	A variable used to contain the value of XEA at the time the mini-lattice is formed or enlarged. After the mini-lattice is enlarged for the last time during a given calculation, this quantity may be stored for use in a future calculation.
*XG(I)	An array of constants used to specify values of the x-components of either (1) the Miller indices used to describe the initial direction of motion of PKAs with identification numbers IPI(I), or (2) the displacement from lattice sites that describe the initial orientation of one atom of the pair comprising dumbbell interstitials associated with the lattice sites IPI(I).
*XI(I)	An array of constants used to specify values of the x-components of the displacements from lattice sites to describe the initial orientation of the other atom of each pair comprising the dumbbell interstitials [see XG(I)].

<u>LABEL</u>	<u>DESCRIPTION</u>
XIP	A variable used to contain the pre-initial (i.e., $t_0 - \Delta t$) and imaginary x-coordinate of the I^{th} PKA in order to give it the momentum prescribed by KEPKA(I), POLI(I), and AZII(I) at time t_0 .
*XIPS	A variable containing the x-component of an arbitrarily chosen center of coordinates in the COMENT reference system.
XPKA	A variable used to contain the initial (i.e., t_0) x-coordinate of the I^{th} PKA.
XTS	A variable used to contain the x-coordinate of an atom relative to the COMENT entire-lattice center of coordinates at time $t + \Delta t$. This quantity is calculated for each atom each time step.
XX1	A dummy variable used when summing the interatomic forces to obtain the x-component of the restoring force on an atom.
X1	A variable used to contain the x-coordinate of an atom relative to the COMENT entire-lattice center of coordinates at time t . The quantity in X1 comes from unpacking an X(ID) array element.
X11	(Same as X1)
YB	A variable used to contain the y-component of an atom relative to the z-axis (in the COMENT reference system) that lies along the center of the entire-lattice.
YEA	A variable used to contain the y-coordinate of the most energetic atom in the mini-lattice, relative to the COMENT entire-lattice center of coordinates.

LABELDESCRIPTION

YEA1	A variable used to contain the value of YEA at the time the mini-lattice is formed or enlarged. After the mini-lattice is enlarged for the last time during a given calculation, this quantity may be stored for use in some future calculation.
*YG(I)	An array of constants used to specify values of the y-components of either (1) the Miller indices used to describe the initial direction of motion of PKAs with identification numbers IPI(I), or (2) the displacements from lattice sites that describe the initial orientation of one atom of the pair comprising dumbbell interstitials associated with the lattice sites IPI(I).
*YI(I)	An array of constants used to specify values of the y-components of the displacement from lattice sites to describe the initial orientation of the other atom of each pair comprising the dumbbell interstitial [see YG(I)].
YIP	A variable used to contain the pre-initial (i.e., $t_0 - \Delta t$) and imaginary y-coordinate of the I^{th} PKA in order to give it the momentum prescribed by KEPKA(I), POLI(I), and AZII(I) at time t_0 .
*YIPS	A variable containing the y-component of an arbitrarily chosen center of coordinates in the COMENT reference system.
YPKA	A variable used to contain the initial (i.e., t_0) y-coordinate of the I^{th} PKA.
YTS	A variable used to contain the y-coordinate of an atom relative to the COMENT entire-lattice center of coordinates at time $t + \Delta t$. This quantity is calculated for each atom each time step.

LABELDESCRIPTION

YY1	A dummy variable used when summing the interatomic forces to obtain the y-component of the restoring force on an atom.
Y1	A variable used to contain the y-coordinate of an atom relative to the COMENT entire-lattice center of coordinates at time t. The quantity in Y1 comes from unpacking a X(ID) array element.
Y11	(Same as Y1)
*ZA	A constant containing the atomic number of A (host) atoms.
*ZB	A constant containing the atomic number of B (impurity) atoms.
ZBH	A constant used in defining the effective size of the entire-lattice (along with RBCUT). Atoms with z-coordinates (Z11) larger than ZBH are regularly monitored to determine if they violate the interface boundary conditions.
ZBL	A constant used in defining the effective size of the entire-lattice (along with RBCUT). Atoms with z-coordinates (Z11) smaller than ZBL are regularly monitored to determine if they violate the interface boundary conditions.
ZEA	A variable used to contain the z-component of the most energetic atom in the mini-lattice, relative to the COMENT entire-lattice center of coordinates.
ZEA1	A variable used to contain the value of ZEA at the time the mini-lattice is formed or enlarged. After the mini-lattice is enlarged for the last time during a given calculation, this quantity may be stored for use in some future calculation.

LABELDESCRIPTION

*ZG(I)	An array of constants used to specify values of the z-components of either (1) the Miller indices used to describe the initial direction of motion of PKAs with identification numbers IPI(I), or (2) the displacements from lattice sites that describe the initial orientation of one atom of the pair comprising the dumbbell interstitials associated with the lattice sites IPI(I).
ZH	A constant used to alter the boundary type identification number during a calculation. When the z-coordinate of an interior atom becomes larger than ZH, its boundary type identification number changes (see RCUT).
*ZI(I)	An array of constants used to specify values of the z-components of the displacement from lattice sites to describe the initial orientation of the other atom of the pair comprising the dumbbell interstitial [see ZG(I)].
ZIP	A variable used to contain the pre-initial (i.e., $t_0 - \Delta t$) and imaginary z-coordinate of the I^{th} PKA in order to give it the momentum prescribed by KEPKA(I), POLI(I) and AZII(I) at time t_0 .
*ZIPS	A variable containing the z-component of an arbitrarily chosen center of coordinates in the COMENT reference system.
ZL	A constant used to alter the boundary type identification number during a calculation. When the z-coordinate of an interior atom becomes smaller than ZL, its boundary type identification number changes (see RCUT).

<u>LABEL</u>	<u>DESCRIPTION</u>
ZPE	A dummy variable containing the total zero point configurational energy for all atoms at the lattice-continuum interface when a new mini-lattice is formed.
ZPKA	A variable used to contain the initial (i.e., t_0) z-coordinate of the I^{th} PKA.
ZTS	A variable used to contain the z-coordinate of an atom relative to the COMENT entire-lattice center of coordinates at time $t + \Delta t$. This quantity is calculated for each atom each time step.
ZZX(ID)	An array containing the x-components of the restoring forces on the atoms at time t . This quantity is calculated for each atom each time step (see subroutine NEAF).
ZZY(ID)	An array containing the y-components of the restoring forces on the atoms at time t . This quantity is calculated for each atom each time step (see subroutine NEAF).
ZZZ(ID)	An array containing the z-components of the restoring forces on the atoms at time t . This quantity is calculated for each atom each time step (see subroutine NEAF).
ZZ1	A dummy variable used when summing the interatomic forces to obtain the z-component of the restoring forces on an atom.
Z1	A variable used to contain the z-coordinate of an atom relative to the COMENT entire-lattice center of coordinates at time t . The quantity in Z1 comes from unpacking a X(ID) array element.
Z11	(Same as Z1)

3.2.2 Units

In order to conveniently manipulate the small quantities encountered when performing atomistic simulations, a system of COMENT units was devised for all variables. First, COMENT units of mass, length, and energy were chosen. Then these were used to define the COMENT units of velocity, time and force. The following list relates the COMENT system of units to the mks system of units.

(Note: A is the atomic mass in atomic mass units and a is the crystal parameter in angstrom units. N is the abbreviation for newtons.)

COMENT unit mass (um)

$$1 \text{ um} = A(\text{amu}) \cdot 1.6604 \times 10^{-27} \text{ kgm} \quad (3.1)$$

COMENT unit distance (ud)

$$1 \text{ ud} = a (\text{\AA}) \cdot 10^{-10} \text{ m} \quad (3.2)$$

COMENT unit energy (ue)

$$1 \text{ ue} = 1 \text{ eV} = 1.6021 \times 10^{-19} \text{ joules} \quad (3.3)$$

COMENT unit velocity (uv)

$$1 \text{ uv} = [1 \text{ ue}/1 \text{ ud}]^{1/2} = [9.8229 \times 10^3 / A^{1/2} (\text{amu})] \text{ m/s} \quad (3.4)$$

COMENT unit time (ut)

$$1 \text{ ut} = [1 \text{ ud}/1 \text{ uv}] = [a (\text{\AA}) A^{1/2} (\text{amu}) 6.1082 \times 10^{-15}] \text{ s} \quad (3.5)$$

COMENT unit force (uf)

$$1 \text{ uf} = [1 \text{ ue}/1 \text{ ud}] = [2.6702 \times 10^{-9} / a (\text{\AA})] \text{ N} \quad (3.6)$$

COMENT unit Hookeian force constant

$$1 \text{ uhc} = [1 \text{ uf}/1 \text{ ud}] = [4.4503 \times 10^1 / a^2 (\text{\AA})] \text{ N/m} \quad (3.7)$$

COMENT unit viscous force constant

$$\begin{aligned} 1 \text{ uvc} &= [(1 \text{ uhc}) (1 \text{ um})]^{1/2} \\ &= [\sqrt{A} \text{ (amu)} \times 2.7183 \times 10^{-13} / a \text{ (\AA)}] \text{ N/m/s} \end{aligned} \quad (3.8)$$

4. SUBPROGRAMS

4.1 FORTRAN STATEMENT FUNCTIONS

Individual segments of the COMENT code are described in this section. These segments consist of FORTRAN subprograms, categorized by procedure type, and COMPASS subprograms. The procedure name and list of arguments are given for each FORTRAN subprogram. In the case of subroutine DIRECT (see Section 4.3.6), the RETURNS argument list is also given. Definitions of these arguments can be found in Section 3.2.1 and in the program listings. The COMPASS subprograms are described in Section 4.4.

4.1.1 BOXNO (A,B,C)

In the box search method (see Section 2.1.2), the crystal lattice is divided into cubical boxes $f a$ angstroms on edge, where f is taken to be 0.60 for fcc lattices and a is the lattice parameter. Associated with each box is a unique identification number J , defined by

$$J = \text{BOXNO} (A,B,C) \quad , \quad (4.1)$$

which is calculated frequently in the code. The function $\text{BOXNO} (A,B,C)$ is defined to yield an integer number,

$$\text{BOXNO} (A,B,C) = \text{IFIX}(A-X) + \alpha \text{IFIX}(B-Y) + \alpha \beta \text{IFIX}(C-Z), \quad (4.2)$$

where A , B , and C are coordinates of a point in the box, and (for lattices generated in COMENT) α and β are both 31 while X , Y , and Z are 31.0, 32.0, and 32.0, respectively.

4.1.2 RDIST (A,B,C)

Analyses performed with COMENT are invariably concerned with defect orientations and/or the directions of motion of atoms. Consequently, since the magnitudes of vectors

$$\bar{R} = \bar{i} A + \bar{j} B + \bar{k} C \quad (4.3)$$

are calculated frequently, the function RDIST (A,B,C) was defined such that

$$\begin{aligned} R &= \text{RDIST} (A,B,C) \\ &= (A^2 + B^2 + C^2)^{1/2} \end{aligned} \quad (4.4)$$

where A, B, and C are the x, y, and z components of R.

4.1.3 TRANX (A,B), TRANY (A,B,C), TRANZ (A,B,C), and XTRAN (A,B,C), YTRAN (A,B,C), ZTRAN (B,C)

COMENT utilizes two coordinate systems, the "COMENT coordinate system" and the "unit cell coordinate system." The coordinate axes of the COMENT coordinate system relative to the orientation of the crystal lattice are shown in Figure 2.1. Axes of the unit cell coordinate system lie along the edges of a face-centered-cubic unit cell as outlined by the dashed lines in the same figure. For a cell edge length of $a = 1$, the first neighbor distance is $r_0 = \sqrt{2}/2$, the distance between (111) planes is $b = \sqrt{3}/3$ and the distance between (112) planes is $c = \sqrt{1.5}/6$. The rotational transformation of rectangular space coordinates is accomplished with the equations

$$X = \text{XTRAN} (A,B,C) = \lambda_1 A + \mu_1 B + \gamma_1 C \quad , \quad (4.5)$$

$$Y = \text{YTRAN} (A,B,C) = \lambda_2 A + \mu_2 B + \gamma_2 C \quad , \quad (4.6)$$

$$Z = \text{ZTRAN} (A,B,C) = \lambda_3 A + \mu_3 B + \gamma_3 C \quad , \quad (4.7)$$

$$A = \text{TRANX} (X,Y,Z) = (\lambda_1 X + \lambda_2 Y + \lambda_3 Z)/(0.6)^2 \quad , \quad (4.8)$$

$$B = \text{TRANY} (X,Y,Z) = (\mu_1 X + \mu_2 Y + \mu_3 Z)/(0.6)^2 \quad , \text{ and } (4.9)$$

$$C = \text{TRANZ} (X,Y,Z) = (\gamma_1 X + \gamma_2 Y + \gamma_3 Z)/(0.6)^2 \quad , \quad (4.10)$$

where the coordinates of point P are A, B, C referred to the COMENT coordinate system (i.e., the original axes) and X, Y, Z referred to the unit cell coordinate system (i.e., the new rotated axes). The parameters $\lambda_1, \mu_1, \gamma_1$; $\lambda_2, \mu_2, \gamma_2$; and $\lambda_3, \mu_3, \gamma_3$ are the direction cosines of the new (unit cell) axes with respect to the old (COMENT) axes, i.e., for XTRAN, YTRAN, and ZTRAN

$$\lambda_1 = .6(r_0/a) = .424264,$$

$$\mu_1 = .6(b/a) = .346411,$$

$$\gamma_1 = .6(2c/a) = .244943,$$

$$\lambda_2 = .6(-r_0/a) = -.424264$$

$$\mu_2 = .6(b/a) = .346411,$$

$$\gamma_2 = .6(2c/a) = .244943,$$

$$\lambda_3 = .6(o/a) = .0$$

$$\mu_3 = .6(-b/a) = -.346411, \text{ and}$$

$$\gamma_3 = .6(4c/a) = .489886,$$

The reader should keep in mind that a unit distance in the unit cell system is a, and a unit distance in the COMENT system is .6a. Hence, the constants listed above are used in XTRAN, YTRAN, and ZTRAN, but are divided by $(.6)^2$ for use in TRANX, TRANY, and TRANZ. XTRAN, YTRAN, and ZTRAN transform from COMENT unit distances to lattice units, while TRANX, TRANY, and TRANZ do the reverse.

4.2 FORTRAN EXTERNAL FUNCTIONS

4.2.1 ERF(Y)

The error function is defined by the relation

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2) dt \quad (4.11)$$

which must be approximated for use in numerical computations. Since $\text{erf}(x) = -\text{erf}(-x)$, only positive values are used in the approximations. Three approximations are used to span the positive real axis, i.e., the range of x . Let $F(x)$ denote the approximation to $\text{erf}(x)$ in each range:

$$F(x) = \frac{2}{\sqrt{\pi}} x, \quad \text{for } 0 \leq x < 10^{-5} \quad (4.12)$$

$$F(x) = 1 - [1 + a_1 x + a_2 x^2 + \dots + a_6 x^6]^{-16}, \quad 10^{-5} \leq x < 6 \quad (4.13)$$

$$F(x) = 1, \quad \text{for } 6 \leq x \quad (4.14)$$

The first approximation, Equation (4.12), consists of the first term of the series expansion given by Equation (7.1.5) on p. 297 of reference 44. The second approximation, Equation (4.13), is a rational approximation given by Equation (7.1.28) on p. 299 of reference 44, where the coefficients $a_1, a_2, \dots, a_6$ are also listed. The maximum error incurred by using these approximations is less than or equal to 3×10^{-7} .

4.2.2 GAMIN(X,A)

The incomplete gamma function is defined by the relation

$$P(x,A) = \frac{1}{\Gamma(A)} \int_0^x z^{A-1} \exp(-z) dz \quad (4.15)$$

where $\Gamma(A)$ is the gamma function defined as the above integral evaluated at $x = \infty$ (see Equation 6.1.1 on p. 225 of reference 44). $\Gamma(A)$ is approximated by writing it as

$$\Gamma(A) = (A-1)(A-2)(A-3)\dots(A-j+1) \Gamma(A-j), \quad (4.16)$$

where $0 \leq (A-j) \leq 1$, then approximating $\Gamma(A-j)$ by the polynomial

$$\Gamma(A-j) \approx 1 + a_1(A-j-1) + a_2(A-j-1)^2 + \dots + a_8(A-j-1)^8 \quad (4.17)$$

(See Equation 6.1.36 on p. 257 of reference 44 for the values of $a_1, a_2, \dots, a_8$.) This approximation has a maximum error of 3×10^{-7} for any $(A-j)$ in the range $0 \leq (A-j) \leq 1$.

The integral part of $P(X, A)$ is approximated by the series

$$\int_0^X Z^{A-1} \exp(-Z) dZ = \exp(-X) \sum_{n=0}^{\infty} \frac{X^{n+A}}{(A)_n} \quad (4.18)$$

where $(A)_n = A(A+1)\dots(A+n-1)$ (see reference 45, Equations 18 and 20 on page 41). The above series is truncated when the next term to be added contributes less than one-half unit in the eighth significant digit of the approximation.

4.2.3 INGINV (ALPHA, INGAM)

This real function computes the inverse of the incomplete gamma function with parameter ALPHA whose value is INGAM. That is, if $\alpha = \text{ALPHA}$ and $\xi = \text{INGAM}$, then **INGINV** returns X such that $\xi = P(\alpha, X)$ where

$$P(\alpha, X) = \frac{1}{\Gamma(\alpha)} \int_0^X Z^{\alpha-1} \exp(-Z) dZ \quad (4.19)$$

and where $\Gamma(\alpha)$ is the gamma function, which is defined as the above integral with X replaced by ∞ (see Equation 6.1.1 on p. 255 of reference 44). Newton's method (see Equation 3.9.5 on p. 18 of reference 44) is used to produce X by iteration. The iteration function is given by

$$X_{n+1} = X_n - \frac{P(\alpha, X_n) - \xi}{\frac{1}{\Gamma(\alpha)} X_n^{\alpha-1} e^{-X_n}} \quad (4.20)$$

for $n = 0, 1, 2 \dots$. The starting value X_0 is chosen as

$$X_0 = \alpha + \begin{cases} -\sqrt{\alpha} & \text{for } \xi < \alpha - 1 \\ 0 & \text{for } \xi = \alpha - 1 \\ \sqrt{\alpha} & \text{for } \xi > \alpha - 1 \end{cases} \quad (4.21)$$

The sequence is terminated whenever $|X_{n+1} - X_n| \leq 0.5 \times 10^{-7}$. If n exceeds 100 and the convergence test has not been satisfied, the value returned is the negative of the last element X_n generated.

The function contains a constant named BIGNO, which is used to designate special conditions encountered. The value used in the current version is 9.99×10^{99} . This number is larger than any expected, but it also must be in the range permitted by the computer hardware limitations. The value returned is $\pm \text{BIGNO}$ depending on the following conditions:

$$\text{INGINV} = \begin{cases} -\text{BIGNO} & \begin{array}{l} \text{if } \text{ALPHA} \leq 0, \text{ and } \text{INGAM} < 0 \text{ or} \\ \text{INGAM} > 1, \text{ underflow occurs} \\ \text{during iteration} \end{array} \\ +\text{BIGNO} & \text{if } \text{INGAM} = 1. \end{cases} \quad (4.22)$$

4.3 FORTRAN SUBROUTINES

4.3.1 ALOYORD (FA, FAA, FB, FBB, S, SO, INSERT, NPOT)

Subroutine ALOYORD generates a binary alloy with an $L1_2$ superlattice structure, the degree of long-range order specified by S , and the chemical composition designated by FB , the fraction of B atoms in the alloy. There

are two main tasks performed by ALOYORD. The first task is to distribute the A and B atoms on lattice sites so as to obtain the desired degree of long-range order with the given composition. The second task performed by ALOYORD is the statistical analysis, which characterizes the element distributions in terms of long- and short-range order. The degree of order is either specified by S, the long-range order parameter, or by FAA and FBB, the fraction of A-atom ideal lattice sites occupied by A-atoms and the fraction of B-atom ideal lattice sites occupied by B atoms, respectively (see Section 2.1.3.2). If S is specified, then

$$FAA = \left(\frac{4}{3}\right) FA FB S + FA, \text{ and} \quad (4.23)$$

$$FBB = 3 FAA + 4 FB - 3, \quad (4.24)$$

where $FA = 1 - FB$. If FAA and FBB are specified, then

$$S = \frac{3}{4} \left(\frac{FAA - FA}{FB} \right) + \frac{1}{4} \left(\frac{FBB - FB}{FA} \right) \quad (4.25)$$

The two methods are distinguished by the INPUT argument SO.

If $SO = 1$, then FAA and FBB are given and the resulting ordered alloy falls into one of three categories for which S must be computed with Equation (4.25). These categories are characterized as follows:

- a) When $FB = 0.25$ (i.e., $FA = 0.75$) the parameters FAA, FBB and S are each set equal to unity, i.e., a perfectly ordered A_3B , $L1_2$ superlattice is generated.
- b) When $FB < 0.25$ the order-specifying parameters are set such that $FAA = 1$, $FBB = 4FB$, and S is calculated; i.e., an ordered alloy is generated whose only deviation from a perfectly ordered $L1_2$ superlattice is that due to an excess of A atoms randomly distributed on B atom lattice sites.

- c) When $FB > 0.25$ the order-specifying parameters are set such that $FBB = 1$, $FAA = 4(1 - FB)/3$, and S is calculated; i.e., an ordered alloy is generated whose only deviation from a perfectly ordered $L1_2$ superlattice is that due to an excess of B atoms randomly distributed on A atom lattice sites.

If $S \neq 1$, then S is given and the resulting ordered alloy falls into one of two categories for which FAA and FBB must be computed with Equations (4.23) and (4.24). These categories are characterized as follows:

- a) If $FB = 0.25$ (i.e., $FA = 0.75$) the parameter S is specified ($S \neq 1$) and FAA and FBB are calculated; i.e., a disordered A_3B $L1_2$ superlattice is generated.
- b) If $FB \neq 0.25$ the parameter S is specified ($S \neq 1$) and FAA and FBB are calculated; i.e., a disordered $L1_2$ superlattice is generated with $S < 3/[4(1-FB)]$.

Alloy crystal lattice generation is accomplished in three stages. First, the lattice is generated with approximately the stoichiometry and disorder specified. Second, the numbers of A and B-atoms are adjusted sequentially by choosing an atom at random and changing L , its element identification number, so as to move the atom in the direction of the prescribed mix. The third step involves choosing an atom at random, randomly selecting one of its neighbors that differs in kind, and exchanging the two. As in the second stage, this choice is made so as to move toward the specified level of disorder. The order-disorder operation, as the third stage is called, may fail in the sense that an excessive number of atoms may be selected and rejected, leaving the disorder level unchanged. If this occurs it is assumed that the candidate atoms are very sparsely located in the crystal and a systematic sequential search is conducted. A diagnostic message is written when this action is taken.

In addition to the analysis of long-range order, there is also an analysis of short-range order. This consists of examining the neighbors of each interior atom (those with boundary type M less than 10) and constructing four histograms, one each for A-atoms with A-neighbors, A-atoms with B-neighbors, B-atoms with A-neighbors, and B-atoms with B-neighbors. Each histogram contains 14 bins. Atoms having 0, 1, 2, ..., 12, and >12 neighbor atoms of a given type lying within 0.8 CLP (where CLP is the lattice parameter) are counted, and each total is stored in a bin. The bin counts are converted to fractions of the total number of atoms of the specified type, and are listed in the OUTPUT.

4.3.2 CRYSTL

Subroutine CRYSTL generates an ideal face-centered-cubic lattice with x-, y-, and z-coordinate axes in the $\langle 1\bar{1}0 \rangle$, $\langle 11\bar{1} \rangle$, and $\langle 112 \rangle$ crystallographic directions, respectively. As is typical in atomistic simulations, the crystals analyzed by COMENT are assumed to consist of three regions. There is an inner region of discrete atoms, each with a full complement of neighboring lattice sites within the range of the potential, and an outer region of infinite elastic continuum. Between these two, in the lattice-continuum interface region, atoms lack a full complement of neighbors and hence require special (constant pressure and Hookeian) forces to augment the interatomic forces. Boundary type numbers (M) are used to distinguish between atoms at the interface and those in the interior. This subroutine assigns a boundary type identification number (M) to each atom in the lattice. Atoms that are to experience identical pressure, Hookeian, and viscous forces (i.e., have the same surrounding atom configurations) are given the same unique boundary type identification number. For atoms in the interior, $M = 1$, while for atoms at the interface, $M > 1$.

4.3.3 DAMP (M, M2, M1, ENGH, DX1, DY1, DZ1, FM, CCX, CCY, CCZ, IAFIX)

Subroutine DAMP calculates special forces for interface atoms and, if requested, sets viscous damping force constants for interior atoms each time step. Pressure forces and Hookeian force constants are unpacked in DAMP.

4.3.4 DASPKA (POLI, AZII, XG, YG, ZG, IPI, KEPKA, DT, DSP1, DSP2, DSP3, LPRNT1)

Subroutine DASPKA calculates the data required to initiate a PKA of identification number IPI with kinetic energy KEPKA in a direction defined by either the initial polar (POLI) and azimuthal (AZII) angles or the functional directional indices (XG, YG, and ZG). If the calculation is to begin with the initiation of more than one PKA, DASPKA is called more than once. This subroutine may be called NE times for a given calculation.

4.3.5 DFCT (IQ, DXD, DYD, DZD, LPD, DXE, DYE, DZE, LPE, IDV, N12S, IPS)

Subroutine DFCT inserts either a vacancy or an interstitial into the crystal lattice. If the calculation is to begin with the initiation of more than one point defect, DFCT is called more than once. This is the case when defect clusters or defect interactions are to be analyzed. This subroutine may be called NE times for a given calculation.

4.3.6 DIRECT (DBX, DBY, DBZ), RETURNS (A1)

Subroutine DIRECT calculates parameters characterizing atoms significantly displaced from their lattice sites. During the analysis of a PKA, this subroutine is called each time step for each atom that has traveled a distance CL from its lattice site.

4.3.7 DSLOAN

Subroutine DSLOAN was not completed in time to be included in this report. It is anticipated that this subroutine will eventually be added to the code. Reference to DSLOAN in COMENT was made to suggest where the subroutine would be called and what it would do (i.e., analyze the strain field and energy associated with an $(a/2) \langle 110 \rangle$ edge dislocation in a face-centered-cubic lattice).

4.3.8 DSLOIN

Subroutine DSLOIN was not completed in time to be included in this report. It is anticipated that this subroutine will eventually be added to the code. Reference to DSLOIN in COMENT was made to suggest where the subroutine would be called and what it would do (i.e., insert an $(a/2) \langle 110 \rangle$ edge dislocation in a face-centered-cubic lattice).

4.3.9 EXPAND (N1)

Subroutine EXPAND effectively expands a small thermalized lattice into either a medium or large thermalized lattice. Note that when forming a lattice at a specified temperature, a Maxwellian velocity distribution is superposed on atoms at ideal lattice sites. They are then allowed to equilibrate (i.e., acquire an equilibrium displacement distribution) before an analysis of the effect of temperature on a phenomenon of interest is begun. In the interest of good economy, this subroutine takes the equilibrium displacement distribution data obtained with a small lattice and uses lattice symmetry to translate these displacement data to the atoms of a medium or large lattice. This subroutine is automatically called when the thermalized lattice read from TAPE2 has a different number of atoms than are specified for the calculation.

4.3.10 IDDEF (DR, ENG2, ECUT, IPI, UDCC, NB, LPRNT, LPRNT1, LPRNT2)

Subroutine IDDEF identifies defects, locates disturbances near the lattice-continuum interface, and determines whether or not to increase the size of the mini-lattice. If there is no atom within CN of a lattice site, the site is recorded as a vacancy and labeled JDT = 4 for filming. If only one atom is within CN of some site other than its own, it is recorded as a replacement and labeled JDT = 1 for filming. In addition, if two atoms are within CN of some site they are recorded as an interstitial and labeled JDT = 2 or 3 for filming.

4.3.11 IEAF (NPOTA, NPOTB, NPOTAB, LPRNT2, ALLOY)

Subroutine IEAF generates, as a function of interatomic distance, one or three interatomic potential energy arrays (EEA or EEA, EEB, and EEAB) and one or three interatomic-force-per-interatomic-distance arrays (EA or EA, EB, and EAB) depending on whether a pure metal (IELE = 1) or alloy (IELE = 2) is specified. The arrays are 1000 elements long and, together with the interpolation scheme contained in subroutine NEAF, are used to obtain the interatomic potential energy and force between interacting atoms with an interatomic separation in the range RMN, RMX. Interpolating between values in the arrays, as opposed to calculating the energy and force each time they are needed, is considered to yield satisfactory accuracy and make more efficient use of computer resources (sacrificing core storage for computational speed). When long-range potentials (extending beyond the third-nearest neighbor distances) are used, it is sometimes desirable to increase the number of elements in these arrays to 2,000.

At present, there are 12 potential (and 12 corresponding force) functions that may be employed by this subroutine. The potential(s) of interest are selected by specifying the potential identification number(s) NPOTA (or NPOTA, NPOTB, and NPOTAB). Seven of the potentials stored in COMENT, NPOT = 1 through 7, describe the Cu-Cu interaction; the remainder, NPOT = 8 through 12, describe Au-Au, Cu-Au, Ni-Ni, Al-Al, and Ni-Al interactions, respectively. As many additional functions as desired may be added.

4.3.12 INCBOX

Subroutine INCBOX generates the incremented IK arrays used in the box search method. In the box search method, the lattice is divided into cubical boxes, each of which has a unique identification number J. The atom identification numbers of all atoms in each box are stored in each of the corresponding J elements of the IBOX array. Hence, neighboring atoms are

identified by sorting through the IBOX array and effectively searching the contents of surrounding boxes within the range of the potential (see Section 2.1.2). To avoid searching boxes that cannot contain atoms (i.e., those that are outside the lattice), numerous IK arrays are derived, each of which defines a unique search pattern. The search pattern for each atom is obtained from the boundary type number (M) assigned to it.

4.3.13 TABLE (S, FA, FB, NE)

Subroutine TABLE labels all calculations, both those that initiate and those that continue the analysis of a given phenomenon.

4.3.14 MINILAT

Subroutine MINILAT effectively generates a cylindrical or spherical "mini-lattice" within the entire-lattice (see Section 2.1.3.1). When analyzing various phenomena, displacement cascades in particular, it is very inefficient to integrate the equations of motion for all the atoms in the "entire-lattice" each time step, since during most of the analysis either only a small fraction of the atoms move or the atoms distant from the disturbance center move very little. To increase efficiency this subroutine was introduced into COMENT, enabling a user to specify the size and shape of a mini-lattice within the entire-lattice. This is accomplished by assigning temporary identification numbers to atoms that are within a given volume that is to become a mini-lattice, so that the equations of motion are solved only for the atoms within this volume. If the mini-lattice encompasses lattice-continuum interface atoms, COMENT will treat these as specified by the INPUT for the calculation and apply designated special forces when and wherever appropriate. The mini-lattice is expanded as needed. This subroutine is called each time the mini-lattice is to be expanded.

4.3.15 POT1 (W, FO, F1, F2)

Subroutine POT1 calculates the Moliere interatomic potential energy,⁴⁰ its slope (the interatomic force), and its curvature for a given interatomic separation W. This subroutine may be applied to collisions between any two atoms.

4.3.16 POT2 (W, FO, F1)

Subroutine POT2 calculates the Englert interatomic potential energy⁴¹ and corresponding force for an interatomic separation W. This subroutine applies only to Cu-Cu interactions.

4.3.17 POT3 (W, FO, F1)

Subroutine POT3 calculates the interatomic potential energy and corresponding force for the interatomic separation W, where $RM \leq W < RE$; i.e., where W falls in the region between that treated by the Moliere function and that treated by the Englert function.³⁴ This subroutine is used only when NPOT = 3, i.e., only when a mixed Moliere-Englert potential is used to treat Cu-Cu interactions.

4.3.18 POT4 (W, FO, F1)

Subroutine POT4 calculates the Born-Mayer interatomic potential energy⁴² and corresponding force for an interatomic separation W. This subroutine may be applied to collisions between any two atoms.

4.3.19 POT5 (W, FO, F1)

Subroutine POT5 calculates the Johnson interatomic potential energy²¹ and corresponding force for an interatomic separation W. This subroutine may be applied to collisions between any two atoms.

4.3.20 POT6 (W, F0, F1)

Subroutine POT6 directs the subroutines POT4 and POT5 in calculating the interatomic potential energy and corresponding force for an interatomic separation W when NPOT = 6, i.e., only when a mixed Born-Mayer/Johnson potential is used to treat Cu-Cu interactions.

4.3.21 POT7 (W, F0, F1, NPOT)

Subroutine POT7 calculates the Moliere-Heinisch interatomic potential energy⁴³ and corresponding force for an interatomic separation W. This subroutine applies to Cu-Cu, Au-Au, Cu-Au, Ni-Ni, Ni-Al, and Al-Al interactions. NOTE: The potential functions given in the code for these interactions (i.e., NPOT = 8 through 12) are not considered meaningful potentials. They have been included merely to show how the code operates. Researchers should substitute potentials of their own for these materials.

4.3.22 SPEFORS (UDCC)

Subroutine SPEFORS generates the unique special (pressure) forces and force constants (Hookeian and viscous) used to augment interatomic forces at the lattice boundary (i.e., at the lattice-continuum interface). Atoms at the interface, by definition, lack a full complement of neighbors within the range of the potential and must be distinguished from atoms at the lattice interior. Boundary type numbers (M) are used to label atoms and distinguish between those at the interface and those in the interior. Each boundary type number contains major and minor special force indices, M2 and M1, respectively. Atoms with identical configurations of missing neighbors have identical special force indices and share common special force data. The x-, y-, and z-components of the pressure force and Hookeian force constants for all atoms with boundary type M are packed in the AX(M2, M1), AY(M2, M1), and AZ(M2, M1) arrays, respectively. Only one set of viscous force constants VCX, VCY, and VCZ are calculated, and they or some fraction of them are applied to all interface atoms.

4.3.23 SQUEZ (IF)

Subroutine SQUEZ is called by LABEL to suppress extra blanks in the generated alphanumeric formats.

4.3.24 THRMAL (GENVS)

Two tasks are performed in subroutine THRMAL (see Section 2.1.3.3). The first task is to produce a velocity V for each atom from a Maxwellian random number generator, then calculate the positions of the atoms Δt seconds before initiation time based on the assumption that V is the average speed over the Δt time period. The second task is to perform a statistical analysis of the collection of speeds and displacements from the ideal lattice sites (at the current time) associated with the collection of atoms.

4.3.24.1 Task One: Generation of the Velocity Distribution

The generation of a velocity distribution is a two-step process. First, the speed V for each atom is generated from a Maxwellian distribution with the properties

$$V_p = \sqrt{2kT/m} = \text{most probable value of } V, \quad (4.26)$$

$$\langle V \rangle = \frac{2}{\sqrt{\pi}} V_p = \text{expected value of } V, \text{ and} \quad (4.27)$$

$$\langle KE \rangle = \frac{3}{4} V_p^2 = \text{expected kinetic energy}, \quad (4.28)$$

where k is the Boltzmann constant, T is the temperature, and m is the atomic mass. The Maxwellian distribution function is given by

$$f(V) = 4 \pi \left(\frac{m}{2\pi kT} \right)^{\frac{3}{2}} V^2 \exp(-mV^2/2kT) \quad (4.29)$$

with the probability function

$$F(V) = \int_0^V f(V) dV \quad (4.30)$$

The speed V is obtained by choosing ξ from the uniform (0,1) random distribution, then solving the integral equation (see Figure 4.1) for V_ξ

$$\xi = \int_0^{V_\xi} f(V) dV \quad (4.31)$$

By writing $V_p^2 = \frac{2kT}{m}$, we may rewrite this equation as

$$\xi = \frac{2}{\sqrt{\pi}} \int_0^{(V_\xi/V_p)^2} x^{(3/2-1)} \exp(-x) dx = P\left[\frac{3}{2}, \left(\frac{V_\xi}{V_p}\right)^2\right] \quad (4.32)$$

where $P(\alpha, x)$ is the incomplete gamma function (see Section 4.2.2). The function subprogram INGINV (see Section 4.2.3) is used to compute $(V_\xi/V_p)^2$, which is then solved for V_ξ .

The second step consists of selecting a direction for the atom's motion. The direction is determined by generating two random numbers $\cos \theta$ and ϕ from uniform distributions over the ranges $-1 \leq \cos \theta \leq 1$ and $0 \leq \phi \leq 2\pi$, respectively. The velocity components are then obtained from

$$V_x = V \sin \theta \cos \phi, \quad (4.33)$$

$$V_y = V \sin \theta \sin \phi, \quad \text{and} \quad (4.34)$$

$$V_z = V \cos \theta. \quad (4.35)$$

Thus the velocity vector is given by

$$\vec{V} = \vec{i} V_x + \vec{j} V_y + \vec{k} V_z, \quad (4.36)$$

HEDL 7907-159.1

FIGURE 4.1. A Sketch of the Functional Relationship Between V_ξ and ξ .

where $\bar{i}$, $\bar{j}$, $\bar{k}$ are unit vectors parallel to the x, y, and z axes, respectively.

4.3.24.2 Task Two: Statistical Analyses of Velocity and Displacement Distributions

In the following discussion, for the sake of brevity, the variable V may be a velocity component (V_x , V_y , or V_z), the speed, a displacement component (Δx , Δy or Δz), or the total displacement. Displacement is defined relative to the ideal lattice site for the atom. There are two underlying distributions for V . The components of both velocity and displacement have Gaussian distributions, while speed and total displacement have Maxwellian distributions (see Figure 4.2).

A histogram is constructed for the V 's by placing each V in an appropriate "bin" and counting the number in each bin. There are 41 bins in all. Each bin, except for the first and last, are of uniform width. If the right-hand (high-value) endpoint of bin j is denoted by V_j , then $V_{41} = \infty$. For the Maxwellian distributions the bins are defined as

$$V_0 = 0, \quad (4.37)$$

$$V_j = \left[1 + \left(j - 15 + \frac{1}{2}\right)\left(\frac{2}{\sqrt{\pi}} - 1\right)/2\right]V_p, \quad j = 1, 2, \dots, 40, \text{ and} \quad (4.38)$$

$$V_{41} = \infty, \quad (4.39)$$

while for the Gaussian distributions, the bins are defined as

$$V_0 = -\infty, \quad (4.40)$$

$$V_j = \left(j - 21 + \frac{1}{2}\right) \frac{7}{39\sqrt{2}} V_p, \quad j = 1, 2, \dots, 40, \text{ and} \quad (4.41)$$

$$V_{41} = \infty. \quad (4.42)$$

FIGURE 4.2. A Sketch of Gaussian and Maxwellian Probability Distribution Functions for $V_p = \sqrt{2}$.

For the Maxwellian distributions the histograms are situated so that the midpoint of bin 15 is V_p . For the Gaussian distributions the histograms have 0 as the midpoint of bin 21, and excluding the end bins extend 3.5 standard deviations on either side of 0.

The expected number of V 's in bin j is given by

$$\langle N_j \rangle = \Pr[V_{j-1} \leq V \leq V_j] N, \quad j = 1, 2, \dots, 41 \quad (4.43)$$

where N is the total number of atoms under consideration and $\Pr[A]$ is the probability of condition "A" being true. If the V 's are from a Maxwellian distribution then

$$\begin{aligned} \Pr[V_{j-1} \leq V \leq V_j] &= 4\pi \left(\frac{m}{2\pi kT}\right)^{\frac{3}{2}} \int_{V_{j-1}}^{V_j} v^2 \exp[-mv^2/2kT] dv \\ &= P\left[\frac{3}{2}, \left(\frac{V_j}{V_p}\right)^2\right] - P\left[\frac{3}{2}, \left(\frac{V_{j-1}}{V_p}\right)^2\right], \end{aligned} \quad (4.44)$$

where $P(\alpha, x)$ is the incomplete gamma function (see Section 4.2.2). If, on the other hand, the V 's are from a Gaussian distribution, then

$$\Pr[V_{j-1} \leq V \leq V_j] = \frac{1}{\sqrt{\pi}} V_p \int_{V_{j-1}}^{V_j} \exp(-(V/V_p)^2) dV, \quad (4.45)$$

$$= \frac{1}{2} \left[\operatorname{erf}\left(\frac{V_j}{V_p}\right) - \operatorname{erf}\left(\frac{V_{j-1}}{V_p}\right) \right], \quad (4.46)$$

where $\operatorname{erf}(x)$ is the error function (see Section 4.2.1).

The χ^2 statistic is computed from a histogram as follows. First, any bin that in theory has fewer than five V 's is combined with its immediate neighbor bin to the right (i.e., of higher value). This is repeated until

at least five values are to be expected in the resulting bin. If $\langle N_{41} \rangle$ is less than five, it is combined with its neighbor on the left. For the χ^2 to be useful, N should be at least 10 (N being the number of atoms), and there should be at least two bins after the combination occurs.

Yates' correction form⁴⁶ of the χ^2 is used here:

$$\chi^2 = \sum_j \frac{[|N_j - \langle N_j \rangle| - \frac{1}{2}]^2}{N_j - \langle N_j \rangle}, \quad (4.46)$$

where j runs from 1 to the number of bins remaining after combining them (to ensure that N_j is greater than 4). N_j is the observed number in bin j and $\langle N_j \rangle$ is the expected number. The number of degrees of freedom associated with the calculated χ^2 is one less than the number of bins used in the calculation.

The statistical analysis consists of one of two types of calculations. If the INPUT parameter GENVS is TRUE the V's are generated from known theoretical distributions. However, if GENVS is FALSE, the V's do not necessarily adhere to the theoretical distributions. In this case one needs to estimate the parameter defining the distribution functions. The Gaussian distribution is defined by the mean and variance. Unbiased estimates for these two parameters are

$$\langle X \rangle = \frac{1}{N} \sum_{j=1}^N X_j, \quad (4.47)$$

and

$$s_x^2 = \frac{1}{N-1} \sum_{j=1}^N (X_j - \langle X \rangle)^2, \quad (4.48)$$

respectively, where X_j denotes a velocity or displacement component. Once $\langle X \rangle$ and s_x are known, the expected number of atoms with velocity (or displacement) V in bin j is

$$\langle N_j \rangle = \frac{1}{2} \left[\operatorname{erf} \left(\frac{V_j - \langle X \rangle}{s_x \sqrt{2}} \right) - \operatorname{erf} \left(\frac{V_{j-1} - \langle X \rangle}{s_x \sqrt{2}} \right) \right] N \quad (4.49)$$

The Maxwellian distribution is a special case of the two-parameter incomplete gamma distribution

$$P[\alpha, (\frac{X}{V_p})^2] = \frac{1}{\Gamma(\alpha)} \int_0^{X/V_p} z^{\alpha-1} \exp(-z) dz \quad (4.50)$$

Parameters α and V_p are estimated by the following procedure⁴⁶:

$$V_p^2 = \frac{s_x^2}{\langle X^2 \rangle} \quad (4.51)$$

where

$$\langle X^2 \rangle = \frac{1}{N} \sum_{j=1}^N x_j^2 \quad , \text{ and} \quad (4.52)$$

$$s_x^2 = \frac{1}{N-1} \sum_{j=1}^N (x_j^2 - \langle X^2 \rangle)^2 \quad (4.53)$$

Hence, the quantity α may be estimated from

$$\hat{\alpha} = \frac{\langle X^2 \rangle}{V_p^2} \quad (4.54)$$

such that if the data are from a Maxwellian distribution, $\hat{\alpha} \sim \frac{3}{2}$.

The following procedure is used to estimate the mode of the histogram of velocity and displacement data. Let f_j denote the number count or frequency in bin j of the histogram, excluding the tails of the histogram from consideration, i.e., $j = 2, 3, \dots, 40$. Thus, all bins considered have equal width ΔV . The procedure consists of choosing a subset J of the bins, fitting a least-squares parabola to the counts for this subset, then finding the maximum of this parabola. The midpoint of bin j is given by

$$V_{j-1/2} = V_{21} + (j - 21.5) \Delta V \quad j = 2, 3, \dots, 40. \quad (4.55)$$

Similarly we may write any V in the range $V_1 < V < V_{40}$ as

$$V(\xi) = V_{21} + (\xi - \frac{1}{2}) \Delta V, \quad (4.56)$$

where ξ is continuous over the range $-19.5 \leq \xi \leq 19.5$. Notice that when ξ is an integer, say k , then $V(k) = V_{k+20.5}$, the midpoint of bin $k+21$. Thus the polynomial

$$f(\xi) = a + b\xi + c\xi^2 \quad (4.56)$$

that fits the points $(V_{j-1/2}, f_j)$ in subset J chosen from the histogram is sought. A least-squares procedure is used for obtaining coefficients b and c . The coefficient a is not required, since the maximum of $f(\xi)$ occurs at the point

$$\xi_m = \frac{b}{2c} \quad (4.58)$$

Once ξ_m is determined, the mode of the histogram is given by

$$V_m = V_{21} - (\frac{b}{c} + 1) \frac{\Delta V}{2}. \quad (4.59)$$

If $c = 0$, this procedure fails and V_m is taken as the midpoint of the bin having the largest count. If more than one bin has the largest count (i.e., ties exist), V_m is the left-most candidate.

The determination of the set of bins J used in the mode calculation is as follows: let M denote the largest count in bins $2, 3, 4, \dots, 40$. The set J of bins used in determining the mode consists of all bins whose count is greater than or equal to $M/2$. The tails of the histogram, bins 1 and 41, have been excluded because they have widths different from ΔV . Unless the histogram is badly skewed, these bins have small counts and neglecting them should have no effect on the estimate of the mode of the histogram.

4.4 COMPASS SUBPROGRAMS

Eight COMENT subprograms are written in COMPASS, the CDC assembly language. These subprograms maximize computational speed and make highly efficient use of the machine storage bank. With regard to NEAF, the emphasis is on efficient use of computation time. Since this subroutine is involved in the process of summing forces and energies and is called approximately $N \times M/2$ times per time step (where N is the total number of atoms in the lattice and M is the number of neighboring atoms within the range of the interatomic potential) computation speed is of the essence. While efficient computation is also important for PAC3, UNPAC3, PACF, UNPACF, INBIN, FILBIN, and MTBIN, these subroutines are primarily concerned with effectively expanding machine storage capacity. The first four of these subroutines are used for packing and unpacking real numbers (two or three to a word) and the last three are used for packing and unpacking integer numbers (four to a word).

4.4.1 NEAF

The primary function of NEAF is to perform the force and potential energy calculations for every pair of atoms of interest. For a given atom ID, NEAF checks the occupancy of each box in the scan array J2 identified by the addition of the scan increments (IK(MX,J1)) to the box number J of the atom ID. Each element in the IBOX array that is scanned, J3 = IBOX(J2), is unpacked to obtain the identification numbers (IDT) of atoms interacting with atom ID. When an atom IDT is located in the box being investigated, the distance between atoms ID and IDT is calculated. If this distance does not exceed the maximum interaction distance, the force and potential energy contributions of the pair are calculated and stored for use in the CALLING PROGRAM.

The following is a detailed description, expressed in FORTRAN-like statements, of the operations performed by this subroutine:

Statement Numbers	Description
2 - 11	-Identify Subroutine
12 - 59	-Allocate Storage COMMON/ELMT/ENGPO, ... , IDT COMMON/CENTR/X11, ... , EXTRA COMMON/ALPHA/X, ... , EEAB DIMENSION IPF, Z22, ... , NEAF
60 - 81	-Unpack J3 for as Many as Four IDT's COMPT KMARK = 1 COMPT1 DUM1 = ID - IDT IF(ML.EQ.0) GO TO JUMP
91	-Forward Search (when ML=1) IF(DUM1.GT.0) GO TO SWTH
92	-Complete Search (when ML=0) JUMP IF(DUM1.EQ.0) GO TO SWTH
93 - 99	-Identify Material IPF = 1 DUM2 = IELE - 1 IF(DUM2.EQ.0) GO TO MET
100 - 110	-Unpack IDENT (for LT only) and Determine Potential Function Identification Number IPF for the ID-IDT Pair MTBIN (IDENT(IDT), , LT, ,) IPF = L + LT - 1
111 - 127	-Unpack X for Neighboring Atom IDT MET UNPAC3 (X(IDT),X22,Y22,Z22)
128 - 156	-Calculate RR, the Square of the Distance Between ID and IDT DUM3 = X11 - X22 DUM4 = DUM3**2 DX = DUM3

Statement
Numbers

Description

	<pre> DUM3 = Y11 - Y22 DUM4 = DUM4 + DUM3**2 DY = DUM3 DUM3 = Z11 - Z22 RR = DUM4 + DUM3**2 DZ = DUM3 DUM4 = RMAX2 - RR IF(DUM4.LT.0) GO TO SWTH </pre>
157 - 176	<pre> -Locate Registers IE and IE1, in the Force and Energy Arrays that Bracket Values of Interest, and Calculate Linear Interpolation Constant DELH DUM5 = E1*RR + E2 IE = IFIX(DUM5) DELH = DUM5 - AINT(DUM5) DUM6(IE1) = IE - 1 </pre>
177 - 199	<pre> -Calculate the Interatomic Force F1 per Unit Distance for the ID - IDT Pair DUM7 = IPF - 1 DUM8 = EA(IE1) DUM9 = EA(IE) IF(DUM7.EQ.0) GO TO META DUM8 = EAB(IE1) DUM9 = EAB(IE) DUM7 = DUM7 - 1 IF(DUM7.EQ.0) GO TO META DUM8 = EB(IE1) DUM9 = EB(IE) META F1 = DUM9 + (DUM8 - DUM9)*DELH </pre>
200 - 224	<pre> -Calculate the Interatomic Potential Energy P1 for the ID - IDT Pair DUM7 = IPF - 1 DUM8 = EEA(IE1) DUM9 = EEA(IE) IF(DUM7.EQ.0) GO TO METAL DUM8 = EEAB(IE1) DUM9 = EEAB(IE) DUM7 = DUM7 - 1 IF(DUM7.EQ.0) GO TO METAL DUM8 = EEB(IE1) DUM9 = EEB(IE) METAL P1 = DUM9 + (DUM8 - DUM9)*DELH </pre>

Statement Numbers	Description
225 - 276	<p>-Calculate the Force Components and Transfer them to the CALLING PROGRAM Via XX1, YY1, ZZ1 for Atom ID and Store them in the ZZX, ZZY, ZZZ Arrays for Atom IDT</p> <pre> XX1 = XX1 + F1 * DX DUM1 = IDT - N12 IF (DUM1.GT.0) GO TO SKIP1 ZZX(IDT) = ZZX(IDT) - F1 * DX SKIP1 YY1 = YY1 + F1 * DY DUM1 = IDT - N12 IF(DUM1.GT.0) GO TO SKIP2 ZZY(IDT) = ZZY(IDT) - F1 * DY SKIP2 ZZ1 = ZZ1 + F1 * DZ DUM1 = IDT - N12 IF(DUM1.GT.0) GO TO SKIP3 ZZZ(IDT) = ZZZ(IDT) - F1 * DZ </pre>
277 - 338	<p>-Calculate the Total Potential Energy ENGPI(ID) and Part of the Potential Energies ENGPI(IDT) and ENGPO</p> <pre> SKIP3 DUM1 = IDT - N12 IF(DUM1.GT.0) GO TO SIDE ENGPO = ENGPO + P1 ENGPI(ID) = ENGPI(ID) + P1/2 ENGPI(IDT) = ENGPI(IDT) + P1/2 GO TO SWTH SIDE ENGPO = ENGPO + P1/2 ENGPI(ID) = ENGPI(ID) + P1/2 SWTH DUM2 = KMARK + 1 DUM3 = INIB - DUM2 IF(DUM3.EQ.0) RETURN KMARK = DUM2 IF(IDN(KMARK).EQ.0) RETURN IDT = IDN(KMARK) GO TO COMPT1 BIAS DATA / 17254000000000000000 8 END </pre>

4.4.2 Packing and Unpacking Real Numbers

4.4.2.1 Real Number Representation

A number composed of an integral and fractional part may be written as

$$k_b = d_{n-1} b^{n-1} + \dots d_0 b^0 + \dots d_{-m} b^{-m}, \quad (4.59)$$

and abbreviated as $K \times b^N$. The digit d_0 , often called the radix point, separates the integral part (containing n digits) from the fractional part (containing m digits), k is the number, K is the coefficient of the number, N is the exponent of the number, and b is the base of the number system. In the decimal system $b = 10$ and 10 digits (0,1 ...9) are used to represent the number, in the binary system $b = 2$ and 2 digits (0 and 1) are used to represent the number, and in the octal system $b = 8$ and 8 digits (0, ... 6,7) are used to represent the number. For example, consider the decimal number $23 \frac{1}{4}$. When written in the form of Equation (4.59) the number appears in the decimal system as

$$23.25_{10} = 2 \times 10^1 + 3 \times 10^0 + 2 \times 10^{-1} + 5 \times 10^{-2} \quad (4.60)$$

The equivalent number expressed in the binary system becomes

$$\begin{aligned} 23.25_{10} &= 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 \\ &\quad + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = 10111.01_2 \end{aligned} \quad (4.61)$$

Similarly, in the octal system the number becomes

$$23.25_{10} = 2 \times 8^1 + 7 \times 8^0 + 2 \times 8^{-1} = 27.2_8 \quad (4.62)$$

Note that every third power of 2 is a power of 8; hence, binary digits may be grouped in three's, starting at the radix point, and expressed as octal digits. The binary digit groups equivalent to the octal digits 0 through 7 are 000, 001, 010, 011, 100, 101, 110, and 111, respectively.

Since a number is simply a polynomial in the base b , multiplication by an integral power of b (i) results in a shift of the number relative to a fixed decimal point and (ii) increases or decreases the magnitude of the number. Hence, an integral power of b may serve as a scale factor. If enough is known about the range of operands, the scaling at each step of a computation can be determined in advance. Arithmetic processing can then be regarded as involving only fractions, with the final result multiplied by the scale factor that was previously determined. Incorporating the scale factor into a number creates what is called a floating point number, the arithmetic processing of which requires built-in floating-point operations. However, when only a fixed number of digits are available to represent a number, as is the case in a digital computer, the range of numbers that can be represented is quite limited unless a scale factor is used.

Each computer system has its own particular data storage and manipulation characteristics. The storage bank of a CDC computer is divided into 60-bit registers, or words, where each bit (either "on" or "off") represents a binary digit (0 or 1). Consider the storage of a binary number in such a word. The highest-order bit (the 59th) contains the algebraic sign of the number. The next eleven higher-order bits (58th through 48th) contain the exponent of the number. The remaining 48 bits (47th through 0th) contain the coefficient, where the 0th bit is the radix point; i.e., the binary point is understood to be below the lowest order bit. For convenience, and due to the length of numbers in the binary system, the decimal or octal systems are used when programming number manipulations.

As an example, consider the storage of some real number, say +40.25. The process is as follows: (i) the number is converted from base 10 to base 2, i.e., $+40.25_{10} = 101\ 000.010_2$. (ii) The binary number is converted to the form $K \times b^N$, i.e., $101\ 000.010_2 = 101\ 000\ 010\ 000\ \dots\ 000 \times 2^{-42}$ (note: there are only 48 bits available for the coefficient). (iii) The exponent is converted to binary form, i.e., $-42_{10} = -101\ 010_2$, biased by the addition of the largest binary number that can be stored in 10 bits,

i.e., $1\ 111\ 111\ 111_2 + (-101\ 010_2) = 01\ 111\ 010\ 101_2$, and stored with the coefficient, i.e.,

4.4.2.2 PAC3 (A,X,Y,Z) and UNPAC3 (A,X,Y,Z)

Crystal lattices (see Section 2.1.3) simulated with COMENT are stationary and of fixed dimensions; i.e., all atom coordinates in COMENT unit distances (see Section 3.2.2) have values between +32.000... and +63.999... ud. When expressed in the binary system, in the form $K \times b^N$, and stored in the computer, the 15 highest-order bits for each of these numbers are

+32.000 ...	001 111 010 101	1	00
+63.999 ...	001 111 010 101	1	11

Therefore, the 13 highest order bits for any decimal number in this interval are always the same. Since this part of the number never changes, it is deleted and the remaining most significant bits are stored to represent the number. The deleted portion of the number is later restored before the number is used in a calculation.

PAC3 (A,X,Y,Z) is designed to pack the x-, y-, and z-coordinates of an atom into one word, A. Twenty bits are used for each coordinate. In each case, the first thirteen bits of the coordinate are deleted and the next twenty most significant bits are stored. The abbreviated x-coordinate is stored in bits 59 through 40, the abbreviated y-coordinate in bits 39 through 20, and the abbreviated z-coordinate in bits 19 through 0. The numbers packed in the word can be manipulated; however, no calculation can be performed with them without first unpacking and restoring the resulting numbers to their original form.

UNPAC3 (A,X,Y,Z) performs the inverse operation of PAC3. When a set of packed coordinates is needed in a calculation, UNPAC3 unpacks the contents of A, restores the resulting three numbers to their original form and stores them in the separate words X, Y, and Z. That is, the number in the packed word is separated into three twenty-bit numbers, the first digit of each is placed in the 46th bit of its new word and the binary number 001 111 010 101 100 000 000_2 (in octal form 1725400000000000000₈) is added to each, restoring them to their pre-packed form.

The loss of significance resulting from the packing and unpacking operations is discussed in Section 2.1.4.

4.4.2.3 PACF (A,P,H) and UNPACF (A,P,H)

When the equations of motion (see Section 1.3.2.1) for interface atoms are solved to determine their positions, pressure and Hookeian forces must be applied in order to account for missing neighbors and obtain realistic force balances. In COMENT units (see Section 3.2.2) the pressure forces are in the range $\pm 10 u_f$ and the Hookeian force constants are on the order of $+10^2 u_f/u_d$. As explained in Section 4.4.2.1, when expressed in the binary system, in the form $K \times b^N$, and stored in a computer word, the 12 highest-order bits contain sign and exponent information and all lower order bits contain coefficient information. The number of significant decimal figures associated with a binary coefficient of this length is far more than is required to adequately represent the pressure forces and Hookeian force constants. Therefore, these numbers are masked, shifted, and combined to retain in a single word only digits from the 33 highest-order bits of the word containing the force, and digits from the 27 highest-order bits of the word containing the corresponding force constant. Hence, the binary coefficient for the packed force consists of 21 digits, and that for the packed force constant consists of 15 digits.

PACF (A,P,H) is designed to pack either the x-, y-, or z-component of a pressure force together with the corresponding x-, y-, or z-Hookeian force

constant for atoms of a given boundary type into one word, A. The component of the force P is stored in bits 59 through 27, and the component of the force constant H in bits 26 through 0. The three words containing, respectively, the x-, y-, and z-components of the force and force constant for a boundary type can be manipulated; however, no calculation can be performed with such numbers without first unpacking them.

UNPACF (A,P,H) performs the inverse operation of PACF. When a packed force component and corresponding Hookeian force constant are needed in a calculation, UNPACF unpacks the number A and stores the two resulting numbers in separate words, P and H.

The loss of significance resulting from the packing and unpacking operation is discussed in 2.1.4.

4.4.3 Packing and Unpacking Integer Numbers

4.4.3.1 Integer Number Representation

Representation of integers in a computer is much simpler than representation of real numbers. Since each word of a CDC computer consists of 60 bits, each bit capable of representing one binary character, any integer I in the interval $-2^{60} < I < +2^{60}$ can be represented with one word. The integer number is converted to binary form (see Section 4.4.2.1) and stored with the least significant character in the 0th bit.

As an example, consider the storage of the integer number 436. The decimal number is first converted to binary form, i.e., $436_{10} = 110\ 110\ 100_2$, then the binary number is stored with the least significant character in the 0th bit, i.e.,

The representation of integers is exact. Thus, the number of bits required to represent an integer is a function of the size of the integer only. Since the identification integers generated by COMENT are small (less than 32,000), when appropriate these numbers are grouped for storage in a single computer word by dividing the word into segments, each of which is capable of representing the largest number in the group. An integer is then stored in each segment. When it is desired to retrieve the individual integers for calculation, the word is divided into the same segments that were used in the packing operation and each segment is stored in a separate word.

4.4.3.2 INBIN (II,J1,ID), FILBIN (II,K,L,M,J), and MTBIN (II,K,L,M,J)

The wide variety of calculations performed by COMENT in the course of an analysis requires efficient organization and identification of data. This is achieved by rapid storage, retrieval, and manipulation of integers from arrays generated to identify atoms relative to specific features of the simulated lattice. The integers are used to select various calculational procedures and direct overall flow for a particular analysis.

There are two packed integer arrays in COMENT. One array, IBOX(J), stores information used to identify atoms that are immediate neighbors of any given atom. The other, IDENT(I), stores information concerning the chemical identity of atoms (see Section 2.1.3.2), their boundary type (see Section 2.1.3), and their approximate locations in the lattice (see Section 2.1.2). Three subroutines are used to pack and unpack integer arrays:

- (1) INBIN (II,J1,ID) is used to pack two numbers from registers J1 and ID into one word, II. Forty-five bits are used for the first number and fifteen bits for the second. The first number consists of one integer, or two or three integers combined; the second number is a single integer. The number from J1 is stored in a word, left-shifted by fifteen bits, then the number from ID is added to the word.

-

- (ii)

- (iii)

- (iv)

(2) FILBIN(II,K,L,M,J) is used to pack four integer numbers from registers K, L, M, and J into one word, II, in a single operation. The steps involved are essentially those described for INBIN, but repeated four times. In the packed word the numbers from K, L, M, and J are stored with their least significant digits in bits 45, 30, 15, and 0, respectively. As mentioned above, packed integers must be unpacked for the computer to recognize the individual integers.

(3) MTBIN(II,K,L,M,J) performs the inverse operation of FILBIN. When a set of packed integers is needed in a calculation, MTBIN unpacks the number in II and stores the four resulting numbers in separate words K, L, M, and J. MTBIN is used to unpack numbers from both IBOX(J) and IDENT(I) arrays.

5. COMENT LISTING

*** SMALL LATTICE ***

*COMDECK COM1

```
COMMON RMIN, DT, N, NL, VX(4420), TB(4420), TEMP, DEFECT,
+ DTN, NS, NLS, NA, UVC, UHC, UF
COMMON /ALPHA/ X(4420), ZZ(4420), ZZY(4420), ZZZ(4420),
+ ENGPI(4420), IDENT(4420), IBOX(10700), EA(1001), EEA(1001),
+ E3(1001), EEB(1001), EAB(1001), EEAB(1001)
COMMON /ELMT/ ENGPO, ID, E1, E2, J3, DIV, RMAX2, INIB, N12, ML,
1 IELE, L, LT, IPT
LEVEL 2, X, ZZ(4420), ZZY, ZZZ, ENGPI, IDENT, IBOX, EA, EEA, EB, EE3,
+ EAB, EEAB
```

*COMDECK COM2
USELCM /ALPHA/

```
X      BSS      4420
ZZX    BSS      4420
ZZY    BSS      4420
ZZZ    BSS      4420
ENGPI  BSS      4420
IDENT  BSS      4420
IBOX   BSS      10700
```

*COMDECK COM3
DATA ZH/40.4/, ZH/39.3/, IDC/2367/, N1/4264/, NB/10700/

*COMDECK COM4
DATA ZST/41.58/

*** MEDIUM LATTICE ***

*COMDECK COM1

```
COMMON RMIN, DT, N, NL, VX(6460), TB(6460), TEMP, DEFECT,
+ DTN, NS, NLS, NA, UVC, UHC, UF
COMMON /ALPHA/ X(6460), ZZ(6460), ZZY(6460), ZZZ(6460),
+ ENGPI(6460), IDENT(6460), IBOX(14600), EA(1001), EEA(1001),
+ E3(1001), EEB(1001), EAB(1001), EEAB(1001)
COMMON /ELMT/ ENGPO, ID, E1, E2, J3, DIV, RMAX2, INIB, N12, ML,
1 IELE, L, LT, IPT
LEVEL 2, X, ZZ(6460), ZZY, ZZZ, ENGPI, IDENT, IBOX, EA, EEA, EB, EE3,
+ EAB, EEAB
```

*COMDECK COM2
USELCM /ALPHA/

```
X      BSS      6460
ZZX    BSS      6460
ZZY    BSS      6460
ZZZ    BSS      6460
ENGPI  BSS      6460
IDENT  BSS      6460
IBOX   BSS      14600
```

*COMDECK COM3
DATA ZH/44.5/, ZH/43.4/, IDC/4335/, N1/6232/, NB/14600/

*COMDECK COM4
DATA ZST/45.6629/

*** LARGE LATTICE ***

*COMDECK COM1

COMMON RMIN, DT, N, NL, VX(12580), TB(12580), TEMP, DEFECT,
+ DTN, NS, NLS, NA, UVC, UHC, UF
COMMON /ALPHA/ X(12580), ZZX(12580), ZZY(12580), ZZZ(12580),
+ ENGPI(12580), IDENT(12580), IBOX(26100), EA(1001), EEA(1001),
+ E3(1001), EEB(1001), EAB(1001), FEAB(1001)
COMMON /ELMT/ ENGPO, ID, E1, E2, J3, DIV, RMAX2, INIB, NI2, ML,
1 IELE, L, LT, IDT
LEVEL 2, X, ZZX, ZZY, ZZZ, ENGPI, IDENT, IBOX, EA, EEA, EB, EEB,
+ EAB, EEAB

*COMDECK COM2

USELCM /ALPHA/

X	BSS	12580
ZZX	BSS	12580
ZZY	BSS	12580
ZZZ	BSS	12580
ENGPI	BSS	12580
IDENT	BSS	12580
IBOX	BSS	26100

*COMDECK COM3

DATA ZBH/56.7/, ZH/55.6/, IDC/10239/, N1/12136/, N8/26100/

*COMDECK COM4

DATA ZST/57.9104/

DATA ECUT/5.0/, ECU1/20./, S/1.0/, SL/4.0/, ERLI/6.0/
 DATA LPG/20*1/, LPI/20*1/
 60 C COMMON FUNCTIONS
 C *****
 C
 XTRAN(A,B,C) = .424264*A+.346411*B+.244943*C
 YTRAN(A,B,C) = -.424264*A+.346411*B+.244943*C
 65 ZTRAN(A,B) = -.346411*A+.489886*B
 ROTST(A,B,C) = (A**2+B**2+C**2)**.5
 TXANX(A,B) = 1.17851*A - 1.1/851*B
 TXANY(A,B,C) = .962253*A + .962253*B - .962253*C
 70 TXANZ(A,B,C) = .680397*A + .680397*B + 1.36079*C
 BUXXN(A,B,C) = IFIX(A-31.)*.51*IFIX(B-32.)*.961*IFIX(C-32.)
 C
 N2 = N1
 C
 75 C *****
 C READ IN CALCULATION DEFINING DATA
 C *****
 C
 READ (5,DATA)
 80 WRITE (6,DATA)
 IF(.NOT.FTAPE) GO TO 20
 READ (5,710) IDTFL
 WRITE (6,720) IDTFL
 IF(CONT) GO TO 20
 85 WRITE (4) IDTFL
 WRITE(4) DEFECT
 IF(.NOT.RSTAPE) LPUT = .TRUE.
 IF(MEA.EQ.0) MEA = 1P
 20 N = N1
 90 NL = N2
 NI = N+2275
 N12 = NL+1
 ITS = 1TSS
 NS = N
 95 NLS = NL
 N12S = N12
 NAT = 0
 PEO = 0.
 ZPE = 0.
 100 IPS = IP
 MPUTA = MPUTA
 MPUTB = MPUTB
 MPOTAB = MPOTAB
 MAXPKA = 0.0
 105 DO 30 I=1,NE
 MAXPKA = AMAX1(MAXPKA,KEPKA(I))
 30 IF(MAXPKA.EQ.KEPKA(1)) INE = I
 C
 C
 110 C *****
 C GENERATE FORCE AND ENERGY TABLES FOR ALL TYPES OF ATOMIC
 C INTERACTIONS
 C *****
 C

COMMENT 51
 COMMENT 52
 COMMENT 53
 COMMENT 54
 COMMENT 55
 COMMENT 56
 COMMENT 57
 COMMENT 58
 COMMENT 59
 COMMENT 60
 COMMENT 61
 COMMENT 62
 COMMENT 63
 COMMENT 64
 COMMENT 65
 COMMENT 66
 COMMENT 67
 COMMENT 68
 COMMENT 69
 COMMENT 70
 COMMENT 71
 COMMENT 72
 COMMENT 73
 COMMENT 74
 COMMENT 75
 COMMENT 76
 COMMENT 77
 COMMENT 78
 COMMENT 79
 COMMENT 80
 COMMENT 81
 COMMENT 82
 COMMENT 83
 COMMENT 84
 COMMENT 85
 COMMENT 86
 COMMENT 87
 COMMENT 88
 COMMENT 89
 COMMENT 90
 COMMENT 91
 COMMENT 92
 COMMENT 93
 COMMENT 94
 COMMENT 95
 COMMENT 96
 COMMENT 97
 COMMENT 98
 COMMENT 99
 COMMENT 100
 COMMENT 101
 COMMENT 102
 COMMENT 103
 COMMENT 104
 COMMENT 105
 COMMENT 106
 COMMENT 107

5-5

```

115      CALL IEAF(NPOTA,NPOTB,NPOTAB,LPKNT2,ALLOY)
      C
      RCUT=NB(01)=1./(1.41421*.6)
      RMAX2=RMAX**2
120      C
      C      CHANGE FROM ANGSTROMS TO COMENI UNIT DISTANCES
      C      *****
      C
      CM = CN/UDCC
      CN = CN/UDCC
125      CL = CL/UDCC
      CO = CO/UDCC
      CP = CP/UDCC
      CQ = CO/UDCC
      VECT = VECT/UDCC
130      ERLI = ENLI
      ERLI = ERLI/UDCC
      DO 40 I=1,10
40      DERL(I) = DERL(I)/UDCC
      IF(.NOT.DEFECT) GO TO 60
135      DU 50 I=1,NE
      XG(I) = XG(I)*.6/UDCC  $ XI(I) = XI(I)*.6/UDCC
      YG(I) = YG(I)*.6/UDCC  $ YI(I) = YI(I)*.6/UDCC
      ZG(I) = ZG(I)*.6/UDCC  $ ZI(I) = ZI(I)*.6/UDCC
50      CONTINUE
140      60 IF(DEFECT,DR,CONT) GO TO 70
      C
      C
      C      CALCULATE TIME STEP LENGTH
      C      *****
145      C
      IF (DTN,EW,0.0) DTN = SQRT(VECT**2/(2.*MXPKA))
      DTD=DTN
      C
      70 FMA=1.0  $ FMB=1.0
150      IF (MB,NE,0) FMB = MA/MB
      C
      C      CALCULATE UNIT CONVERSION FACTORS
      C      *****
155      C
      UD = UDCC*1.E-10
      UE = 1.6021E-19
      UM = MA*1.6604E-27
      UV = SQRT(UE/UM)
      UT = UD/UV
160      UF = UE/UD
      UMC = UF/UD
      UVC = SQRT(UMC*UM)
      WRITE (6,680) UD,UE,UM,UV,UT,UF,UMC,UVC
      C
      C      DEFINE INCREMENTS USED IN THE BOX SEARCH
      C      *****
165      C
      CALL INCRUX
      IF(RSTAPE) GO TO 100
170      C
      C      *****

```

COMMENT 108
COMMENT 109
COMMENT 110
COMMENT 111
COMMENT 112
COMMENT 113
COMMENT 114
COMMENT 115
COMMENT 116
COMMENT 117
COMMENT 118
COMMENT 119
COMMENT 120
COMMENT 121
COMMENT 122
COMMENT 123
COMMENT 124
COMMENT 125
COMMENT 126
COMMENT 127
COMMENT 128
COMMENT 129
COMMENT 130
COMMENT 131
COMMENT 132
COMMENT 133
COMMENT 134
COMMENT 135
COMMENT 136
COMMENT 137
COMMENT 138
COMMENT 139
COMMENT 140
COMMENT 141
COMMENT 142
COMMENT 143
COMMENT 144
COMMENT 145
COMMENT 146
COMMENT 147
COMMENT 148
COMMENT 149
COMMENT 150
COMMENT 151
COMMENT 152
COMMENT 153
COMMENT 154
COMMENT 155
COMMENT 156
COMMENT 157
COMMENT 158
COMMENT 159
COMMENT 160
COMMENT 161
COMMENT 162
COMMENT 163
COMMENT 164

```

C      GENERATE AN IDEAL FCC LATTICE OF N ATOMS WITH X-, Y-, Z-AXES IN COMMENT 165
C      THE <1-10>, <11-1>, AND <111> DIRECTIONS, RESPECTIVELY. COMMENT 166
C      ***** COMMENT 167
175 C      CALL CRYSTL COMMENT 168
C      SET UP ATOM IDENTIFICATION AND COORDINATE ARRAYS COMMENT 169
C      ***** COMMENT 170
180 C      J = 0 COMMENT 171
C      DO 90 I=1,N COMMENT 172
C      M = IDENT(I) COMMENT 173
C      IF (M.EQ.1) NA = NA+1 COMMENT 174
185 C      L = 1 COMMENT 175
C      DO 80 K=1,IMP COMMENT 176
C      IF ((ELE..OT.1.AND..NOT.ALLOY.AND.I.EQ.IMPUR(K)) L = 2 COMMENT 177
80 C      CONTINUE COMMENT 178
C      CALL FILBIN (II,I,L,M,J) COMMENT 179
190 C      IDENT(I) = II COMMENT 180
C      VX(I) = X(I) COMMENT 181
C      TB(I) = X(I) COMMENT 182
90 C      CONTINUE COMMENT 183
C      IF (.NOT.ALLOY) GO TO 99 COMMENT 184
195 C      ***** COMMENT 185
C      GENERATE AN ORDERED OR DISORDERED A3-B BINARY ALLOY COMMENT 186
C      WITH L1-SUB 2 SUPER LATTICE STRUCTURE COMMENT 187
C      ***** COMMENT 188
200 C      FA=1.,FB COMMENT 189
C      IF (FB.NE.0..AND.FA.NE.0.) GO TO 91 COMMENT 190
C      WRITE(6,1060) COMMENT 191
C      GO TO 670 COMMENT 192
91 C      SO=S $ IF (SU.NE.1,0) GO TO 95 COMMENT 193
C      IF (.25-FB) 92,93,94 COMMENT 194
205 C      92 FAA=4.*FA/3. $ FBB=1,0 $ GO TO 96 COMMENT 195
C      93 FAA=1,0 $ FBB=1,0 $ GO TO 96 COMMENT 196
C      94 FAA=1,0 $ FBB=4.*FB $ GO TO 96 COMMENT 197
C      95 FAA=4.*FA*FB*80/3.*FA $ FBB=3.*FAA+4.*FB-3. COMMENT 198
210 C      96 S=.75*((FAA-FA)/FU)+.25*((FBB-FB)/FA) COMMENT 199
C      WRITE(6,1070) FA,FAA,FB,FBB,S COMMENT 200
C      IF (.NOT.CUNT) INSERT = .TRUE. COMMENT 201
C      CALL ALUYURD (FA,FAA,FB,FBB,S,DU,INSERT,NPOTAB) COMMENT 202
C      INSERT = .FALSE. COMMENT 203
215 C      99 CALL LABE (S,FA,FB,NE) COMMENT 204
C      ***** COMMENT 205
C      IF (XIPS.NE.0..AND.YIPS.NE.0..AND.ZIPS.NE.0.) GO TO 130 COMMENT 206
C      A = TB(IP) COMMENT 207
220 C      CALL UNPAC3 (A,XIPS,YIPS,ZIPS) COMMENT 208
C      GO TO 130 COMMENT 209
C      ***** COMMENT 210
C      ***** COMMENT 211
225 C      ***** COMMENT 212
C      ***** COMMENT 213
C      ***** COMMENT 214
C      ***** COMMENT 215
C      ***** COMMENT 216
C      ***** COMMENT 217
C      ***** COMMENT 218
C      ***** COMMENT 219
100 C      READ (2) ITIME,N,NL,DIO,ITS,TIME,MEE,DTP,STEMP,ERLII,NESET1,IC1,IP COMMENT 220
C      +1,ELUB,XEA,YEA,ZEA,NA,MPOTA,MPUTB,MPOTAB,IELE,FMA,FMB,UT,FA,FAA,FB COMMENT 221

```

230 +,F0B,S,HF,TEU1,PEU1,PEPA,ZPE COMMENT 222
 IF (NPOTA.EQ.HPOTA.AND.NPOTB.EQ.HPOTB.AND.HPOTAB.EQ.HPOTAB) LPOT = COMMENT 223
 +,TRUE. COMMENT 224
 READ (2) (TB(I),I=1,N) COMMENT 225
 READ (2) (X(I),I=1,N) COMMENT 226
 READ (2) (VX(I),I=1,N) COMMENT 227
 235 READ (2) (IDEN(I),I=1,N) COMMENT 228
 READ (2) SFN,MXC,MYC,MZC,VXC,VYC,VZC,PXC,PYC,PZC COMMENT 229
 READ (2) AX,AY,AZ COMMENT 230
 IF(N+20.GE.N1) GO TO 115 COMMENT 231
 CALL EXPAND (N1) COMMENT 232
 240 C ***** COMMENT 233
 NL = N2 COMMENT 234
 115 LFINST = ,TRUE. COMMENT 235
 IF(CONT.AND.IC.EQ.100) IC = IC1 COMMENT 236
 IF(CONT.AND.ERLI1.EQ.100,) ERLI = ERLI1 COMMENT 237
 245 IF(CONT.AND.RESET.EQ.100,) RESET = RESET1 COMMENT 238
 IF(CONT.AND.IP.EQ.100) IP = IP1 COMMENT 239
 IF(CONT.AND.MEA.EQ.100) MEA = MEE COMMENT 240
 IF (TEMP.NE.STEMP.AND.,NOT.TEMSET) WRITE (6,960) STEMP COMMENT 241
 IF (TEMP.NE.STEMP.AND.,NOT.TEMSET) GO TO 670 COMMENT 242
 250 ERLI = ERLI*UOCC COMMENT 243
 IPS = IP COMMENT 244
 N12 = NL+1 COMMENT 245
 ITS = ITIME+1 COMMENT 246
 IF(DEFECT.AND.ITSS.EQ.0) ITSS = ITS COMMENT 247
 255 IF(CONT.AND.DTN.LI.0.0) DTN = UTO COMMENT 248
 IF(CONT.AND.DTN.EQ.0.0) DTN = UTP COMMENT 249
 CALL LABLE (8,FA,FB,NE) COMMENT 250
 C ***** COMMENT 251
 IF (XIPS.NE.0.0.AND.YIPS.NE.0.0.AND.ZIPS.NE.0.0) GO TO 110 COMMENT 252
 260 A = TB(IP) COMMENT 253
 CALL UNPAC3 (A,XIPS,YIPS,ZIPS) COMMENT 254
 IF (DTN.EQ.UTO) GO TO 130 COMMENT 255
 110 C COMMENT 256
 C ***** COMMENT 257
 265 C MAKE 'PASI' COORDINATE ADJUSTMENTS TO CONSERVE MOMENTUM FOLLOWING COMMENT 258
 C A CHANGE IN TIME STEP LENGTH COMMENT 259
 C ***** COMMENT 260
 C COMMENT 261
 DO 120 I=1,NL COMMENT 262
 270 A = X(I) COMMENT 263
 CALL UNPAC3 (A,X11,Y11,Z11) COMMENT 264
 A = VX(I) COMMENT 265
 CALL UNPAC3 (A,VXU,VYU,VZU) COMMENT 266
 VAX = X11-(X11-VXU)*DTN/DTO COMMENT 267
 275 VYX = Y11-(Y11-VYU)*DTN/DTO COMMENT 268
 VZX = Z11-(Z11-VZU)*DTN/DTO COMMENT 269
 IF (VAX.NE.VXU) VAX = VAX+ADDE COMMENT 270
 IF (VYX.NE.VYU) VYX = VYX+ADDE COMMENT 271
 IF (VZX.NE.VZU) VZX = VZX+ADDE COMMENT 272
 280 CALL PAC3 (A,VAX,VYX,VZX) COMMENT 273
 VX(I) = A COMMENT 274
 120 C CONTINUE COMMENT 275
 130 IF (DTN.LE.0.0) WRITE(6,1030) COMMENT 276
 IF (DTN.LE.0.) GO TO 670 COMMENT 277
 285 C COMMENT 278

5-9

		DT = DTN	COMMENT	279
		DT0 = DTN	COMMENT	280
		MEAL = 1P	COMMENT	281
		DDT = DT**2	COMMENT	282
290		DEL = 1./(2.*DDT)	COMMENT	283
		IF (ITIME.GT.0.AND.LPUT) GO TO 150	COMMENT	284
	C		COMMENT	285
	C	*****	COMMENT	286
	C	CALCULATE PRESSURE FORCES AND HOOKIAN FORCE CONSTANTS FOR ATOMS AT	COMMENT	287
295	C	THE LATTICE CONTINUUM INTERFACE	COMMENT	288
	C	*****	COMMENT	289
	C		COMMENT	290
	C	IF (.NOT.IAFIX) CALL SPEFORS (UDCC)	COMMENT	291
	C	*****	COMMENT	292
300		MPOTA = NPOTASMPDIB = NPOTASMPUTAD = NPOTAR	COMMENT	293
		LPOT = .TRUE.	COMMENT	294
		ML = 1	COMMENT	295
		IF (CONT.OK..NOT.TEMSET) GO TO 150	COMMENT	296
	C		COMMENT	297
305	C	*****	COMMENT	298
	C	SUPERIMPOSE A MAXWELLIAN VELOCITY DISTRIBUTION ON THE LATTICE ATOM	COMMENT	299
	C	*****	COMMENT	300
	C		COMMENT	301
		GENVS = .TRUE.	COMMENT	302
310		CALL THERMAL (GENVS)	COMMENT	303
		TEMSET = .FALSE.	COMMENT	304
		IF (.NOT.IAFIX) GO TO 150	COMMENT	305
	C		COMMENT	306
315		DO 140 I=1,N	COMMENT	307
		II = IDENI(I)	COMMENT	308
		CALL MTBIN (II,K,L,M,J)	COMMENT	309
		IF (M.LT.10) GO TO 140	COMMENT	310
		X(I) = VX(I)	COMMENT	311
320	140	CONTINUE	COMMENT	312
	150	CCX = DT*VXC*FV	COMMENT	313
		CCY = DT*VCY*FV	COMMENT	314
		CCZ = DT*VCZ*FV	COMMENT	315
	160	ITIME = ITIME+1	COMMENT	316
		IF (ITIME.GT.ITSS+10) LPRNT = .FALSE.	COMMENT	317
325		IF (RESET.LT.3..OR.MINATOR(XDM).LE.120) LPRNT = .TRUE.	COMMENT	318
		IF (ITIME.NE.ITSS) GO TO 230	COMMENT	319
		IF (ITS.NE.ITSS) GO TO 200	COMMENT	320
		IF (DEFECT) GO TO 180	COMMENT	321
	C		COMMENT	322
330	C		COMMENT	323
	C	INITIATE ANALYSIS OF ONE OR MORE CASCADES	COMMENT	324
	C	*****	COMMENT	325
	C		COMMENT	326
		IF (FTAPE) WRITE(4) NE,TEMP	COMMENT	327
335		DO 170 I=1,NE	COMMENT	328
		II=IDENI(IPI(I))	COMMENT	329
		CALL MTBIN (II,K,L,M,J)	COMMENT	330
		DT1 = DT	COMMENT	331
		IF (L.EQ.2) DT1=DT*SDRT(FMB)	COMMENT	332
340		CALL DASPKA(PULI(I),A2II(I),XG(I),YG(I),ZG(I),IPI(I),KEPKA(I),DT1,	COMMENT	333
		+DSP1,DSP2,DSP3,LPMNT1)	COMMENT	334
		IQ = IPI(I)	COMMENT	335

5-10

```

      A = X(IQ)
      CALL UNPAC3 (A, XPKA, YPKA, ZPKA)
345 XIP = XPKA-DSP1
      YIP = YPKA-DSP2
      ZIP = ZPKA-DSP3
      CALL PAC3 (A, XIP, YIP, ZIP)
      VX(IQ) = A
350 IF (FTAPE) WRITE (4) 1, IQ, XEPKA(I), PULI(I), AZII(I), XG(I), YG(I), ZG(
      +I)
170 CONTINUE
      GO TO 200
C
C INITIATE ANALYSIS OF ISOLATED OR INTERACTING DEFECTS
C *****
C
180 IF (LINDEX.AND..NOT.COUNT) CALL DSLOIN
      IF (LINDEX.OR.COUNT) GO TO 200
360 IF (.NOT.LPDEF.OR.DEFCLS.OR.DEFINT) GO TO 200
      WRITE (6,690)
      DO 190 I=1, NF
      CALL DFCT (IPI(I), XG(I), YG(I), ZG(I), LPI(I), XI(I), YI(I), ZI(I), LPI(I
      +I), IPV(I), N12S, IPS)
365 190 CONTINUE
      200 IF (IC.GT.10) GO TO 210
      IF (SL.EQ.5.) GO TO 210
C
C GENERATE A SPHERICAL OR CYLINDRICAL MINI-LATTICE OF RADIUS ERLI
C *****
C
      CALL MINILAT
      LPRNT = .TRUE.
      XEAL = XEASYFA1 = YEASZEA1 = ZEA
375 ITO = 0
C
C 210 IF (LPRNT.AND.SL.EQ.5.) WRITE (6,740) N, NA
      SPE = PEPA*(N-NAT)
      NAT=N
380 C
C *****
C WRITE THE COORDINATES AND IDENTIFYING INFORMATION FOR ATOMS IN THE
C ENTIRE LATTICE OR NEWLY FORMED MINI-LATTICE
C *****
385 C
      WRITE(6,750) IC, ITS, IPS, ERLI
      IF (.NOT.LPRNT1.AND.SL.EQ.5.) GO TO 230
      IF (LPRNT1) WRITE (6,760)
      DO 220 I=1, N
      II = IDENT(I)
      CALL MTBIN (II, K, L, M, J)
      IF (SL.EQ.5..AND.DEFECT.AND.ADAMP.AND.M.LT.10) M = 91
      CALL FILBIN (II, K, L, M, J)
      IDENT(I) = II
395 A = X(I)
      CALL UNPAC3 (A, X11, Y11, Z11)
      A = TB(I)
      CALL UNPAC3 (A, TBX1, TBY1, TBZ1)
      DX = X11-XIPS

```

COMMENT 336
 COMMENT 337
 COMMENT 338
 COMMENT 339
 COMMENT 340
 COMMENT 341
 COMMENT 342
 COMMENT 343
 COMMENT 344
 COMMENT 345
 COMMENT 346
 COMMENT 347
 COMMENT 348
 COMMENT 349
 COMMENT 350
 COMMENT 351
 COMMENT 352
 COMMENT 353
 COMMENT 354
 COMMENT 355
 COMMENT 356
 COMMENT 357
 COMMENT 358
 COMMENT 359
 COMMENT 360
 COMMENT 361
 COMMENT 362
 COMMENT 363
 COMMENT 364
 COMMENT 365
 COMMENT 366
 COMMENT 367
 COMMENT 368
 COMMENT 369
 COMMENT 370
 COMMENT 371
 COMMENT 372
 COMMENT 373
 COMMENT 374
 COMMENT 375
 COMMENT 376
 COMMENT 377
 COMMENT 378
 COMMENT 379
 COMMENT 380
 COMMENT 381
 COMMENT 382
 COMMENT 383
 COMMENT 384
 COMMENT 385
 COMMENT 386
 COMMENT 387
 COMMENT 388
 COMMENT 389
 COMMENT 390
 COMMENT 391
 COMMENT 392

```

400      DY = Y11-YIPS      COMMENT 393
      DZ = Z11-ZIPS      COMMENT 394
      DX2 = XTRAN(DX,DY,DZ) COMMENT 395
      DY2 = YTRAN(DX,DY,DZ) COMMENT 396
      DZ2 = ZTRAN(DY,DZ)  COMMENT 397
405      DR2 = RDIST(DX2,DY2,DZ2) COMMENT 398
      DA = TBA1-XIPS      COMMENT 399
      DY = TBY1-YIPS      COMMENT 400
      DZ = TBZ1-ZIPS      COMMENT 401
      DX1 = XTRAN(DX,DY,DZ) COMMENT 402
      DY1 = YTRAN(DX,DY,DZ) COMMENT 403
      DZ1 = ZTRAN(DY,DZ)  COMMENT 404
      DR1 = RDIST(DX1,DY1,DZ1) COMMENT 405
      IF (LPRINT1) WRITE (6,770) I,K,L,M,J,DX1,DY1,DZ1,DR1,DX2,DY2,DZ2,DR
+2,X11,Y11,Z11,ENGPI(I) COMMENT 406
415      CONTINUE          COMMENT 407
      C                   COMMENT 408
      C                   COMMENT 409
      C   PREPARE FOR THE (ITIME=TH STEP IN THE) INTEGRATION PROCEDURE
      C   ***** COMMENT 410
      C                   COMMENT 411
420      230 IF (LPRINT) WRITE (6,730) ITIME COMMENT 412
      ENGKA = 0.           COMMENT 413
      ENGSP = 0.           COMMENT 414
      ENGPI = 0.           COMMENT 415
      C                   COMMENT 416
425      C                   COMMENT 417
      C   CONVERT TIME UNITS FROM COMENT UNIT TIME TO LABORATORY SECONDS
      C   ***** COMMENT 418
      C                   COMMENT 419
      C                   COMMENT 420
      C   IF (ITIME.GT,ITSS) TIME = TIME*UT*UT COMMENT 421
430      C                   COMMENT 422
      DO 240 I=1,N8        COMMENT 423
      IBOX(I) = 0          COMMENT 424
      240 CONTINUE         COMMENT 425
      C                   COMMENT 426
435      DO 250 I1=1,NL    COMMENT 427
      ZZ(I1) = 0.          COMMENT 428
      ZY(I1) = 0.          COMMENT 429
      ZZ(I1) = 0.          COMMENT 430
      ENGPI(I1) = 0.       COMMENT 431
      250 CONTINUE        COMMENT 432
440      C                   COMMENT 433
      J = 0                COMMENT 434
      DO 260 I1=1,N        COMMENT 435
      IO = I1              COMMENT 436
      II = IDENT(IO)       COMMENT 437
445      CALL MTBIN (II,K,L,M,J) COMMENT 438
      A = X(IO)            COMMENT 439
      CALL UNPAC3 (A,X11,Y11,Z11) COMMENT 440
      J = BOXND(X11,Y11,Z11) COMMENT 441
      CALL FILBIN (II,K,L,M,J) COMMENT 442
      IDENT(IO) = II       COMMENT 443
450      JJ = IBOX(J)      COMMENT 444
      CALL INBIN (I2,J1,IO) COMMENT 445
      IBOX(J) = I2        COMMENT 446
      260 CONTINUE        COMMENT 447
455      IF (LINDEF) RELAX = .TRUE. COMMENT 448
      ENG21 = 0.0         COMMENT 449

```

```

450 COMMENT
451 COMMENT
452 *****
453 INTEGRATE THE EQUATIONS OF MOTION FOR ATOMS IN THE ENTIRE LATTICE
454 ON NEWLY FORMED MINILATTICE
455 *****
456 DU 570 I3=1,NL
457 IO = 13
458 A = X(ID)
459 CALL UNPACK (A,X11,Y11,Z11)
460 A = Y1(ID)
461 CALL UNPACK (A,Y11,VY11,VZ11)
462 A = Z1(ID)
463 CALL UNPACK (A,Z11,TV11,TB11)
464 M1 = IDENT(ID)
465 CALL ATBIN (M1,K,L,M,J)
466 FM = FMA
467 IF (L.EQ.2) FM = FMB
468 XX1 = 0.
469 YY1 = 0.
470 ZZ1 = 0.
471 COMMENT
472 *****
473 CALCULATE THE MAJOR AND MINOR INDICES (M2 - M1) FOR THE INTERFACE
474 FORCE AND FORCE CONSTANT ARRAY AND THE INDEX (MX) FOR THE
475 SEARCH ARRAY
476 *****
477 IF (M.GT.100.AND.M.LT.3000) GO TO 270
478 COMMENT
479 M2 = 25
480 M1 = M/10
481 MX = 1
482 IF (M.GT.50.AND.M.LT.70) MX = (M-20)/10+13
483 GO TO 280
484 M2 = M/100
485 M1 = M-M2*100
486 IF (M.LE.1299) M3 = M
487 IF (M.GT.1299) M3 = M-1200
488 MX = M3/100+1
489 M4 = LDD(MX)
490 COMMENT
491 CALCULATE THE CONSTITUTIONAL ENERGY AND RESTORING FORCE FOR ATOM
492 IO
493 *****
494 DU 290 J1=1,M4
495 J2 = J+IK(MX,J1)
496 IF (J2.LE.0.AND.J2.GT.2*J1.M3) GO TO 290
497 J3 = IRDX(J2)
498 CALL NEAF
499 CONTINUE
500 DX = X11-IRX1
501 UY = Y11-IRY1
502 UZ = Z11-IRZ1
503 DX = MDIS(UX,OY,0Z)
504 IF (M.LE.U3.DX.M4.GT.3000) GO TO 300
505 COMMENT
506 COMMENT

```

5-12


```

515      CALL DAMP (M,M2,M1,ENGM,DX1,DY1,DZ1,FM,CCX,CCY,CCZ,IAFIX)
      XX1 = XX1+DX1
      YY1 = YY1+DY1
      ZZ1 = ZZ1+DZ1
300      ZZ(X(ID)) = ZZ(X(ID))+XX1
      ZZ(Y(ID)) = ZZ(Y(ID))+YY1
520      ZZ(Z(ID)) = ZZ(Z(ID))+ZZ1
      C
      C      CALCULATE THE NEW POSITION OF ATOM ID
      C      *****
      C
525      IF (M,EW,93,UN,4,GT,3000) GO TO 570
      IF (IAFIX.AND.M,GT,9.AND.M,NE,91.AND.M,NE,95.AND.M,NE,97) GO TO 57
      +0
      XTS = 2.*X11/CXD-VX11*CXN/CXD+UDT*ZZX(ID)*FM/CXD
      DELX = ABS(XTS-X11)
530      IF (DELX.LE,TEST) GO TO 310
      X11 = XTS+AMOE
      YTS = 2.*Y11/CYD-VY11*CYN/CYD+UDT*ZZY(ID)*FM/CYD
310      DELY = ABS(YTS-Y11)
      IF (DELY.LE,TEST) GO TO 320
535      Y11 = YTS+AMOE
      ZTS = 2.*Z11/CZD-VZ11*CZN/CZD+UDT*ZZZ(ID)*FM/CZD
320      DELZ = ABS(ZTS-Z11)
      IF (DELZ.LE,TEST) GO TO 330
      Z11 = ZTS+AMOE
540      330      CONTINUE
      XB = X11-47.469
      YB = Y11-47.053
      M1 = ROIST(XB,YB,0.0)
      IF (RESET,LI,3,DM,M,GE,10) GO TO 340
545      IF (R11,GI,RCUT,UM,Z11,LT,ZL,UM,Z11,GT,ZH) M = 10
      CALL FILHIN (I1,X,L,M,J)
      IDENT(ID) = I1
340      IF (M,GT,1.AND,TEMP,NE,0.0.AND,UR,GT,ORLIM) GO TO 345
      VX(ID) = X(ID)
550      CALL PAC3 (A,X11,Y11,Z11)
      X(ID) = A
345      IF(LINDEF) GO TO 570
      C
      C      *****
555      C      ANALYZE THE ENERGY AND POSITION OF ATOMS AT THE END OF EACH
      C      TIME STEP
      C      *****
      C
      DX = (XTS-VX11)/2.
560      DY = (YTS-VY11)/2.
      DZ = (ZTS-VZ11)/2.
      ENG2 = (DX**2+DY**2+DZ**2)*DEL/FM
      ENGKA = ENGKA+ENG2
      ET2 = ENG2+ENGPI(ID)
565      ENG3 = ((CXD-1.)*UX**2+(CYD-1.)*UY**2+(CZD-1.)*UZ**2)*4.*DEL/FM
      ENGSP = ENGSP+ENG3
      ELNS = ELNS+ENG3
      IF (DM,LE,0.0) GO TO 570
      IF (ITIME,LT,ITSS) GO TO 570
570      IF(DEFECT) GO TO 340

```

```

COMMENT 507
COMMENT 508
COMMENT 509
COMMENT 510
COMMENT 511
COMMENT 512
COMMENT 513
COMMENT 514
COMMENT 515
COMMENT 516
COMMENT 517
COMMENT 518
COMMENT 519
COMMENT 520
COMMENT 521
COMMENT 522
COMMENT 523
COMMENT 524
COMMENT 525
COMMENT 526
COMMENT 527
COMMENT 528
COMMENT 529
COMMENT 530
COMMENT 531
COMMENT 532
COMMENT 533
COMMENT 534
COMMENT 535
COMMENT 536
COMMENT 537
COMMENT 538
COMMENT 539
COMMENT 540
COMMENT 541
COMMENT 542
COMMENT 543
COMMENT 544
COMMENT 545
COMMENT 546
COMMENT 547
COMMENT 548
COMMENT 549
COMMENT 550
COMMENT 551
COMMENT 552
COMMENT 553
COMMENT 554
COMMENT 555
COMMENT 556
COMMENT 557
COMMENT 558
COMMENT 559
COMMENT 560
COMMENT 561
COMMENT 562
COMMENT 563

```

5-14

```

      IF (M,GE,10) GO TO 350
      IF (X11,GT,NCUT) GO TO 350
      IF (Z11,LT,ZBL,OR,Z11,GT,ZBM) GO TO 350
      GO TO 390
575  C
      C *****
      C DETERMINE WHETHER OR NOT ATOM ID IS NEAR AN INTERFACE WITH TOO
      C MUCH ENERGY
      C *****
580  C
      C 350 IF (ET2,LT,ES1) GO TO 390
      IF (ET2,LT,ES2) GO TO 380
      WRITE (6,780) K,ENG2,ENGPI(10)
      WRITE (6,790) J,X11,Y11,Z11
      ISTOP = 1
      IF (ID,EW,IP) GO TO 370
      IF (ET2,GT,ECUT,AND,M,LT,10) GO TO 370
      DO 360 IS=1,5
      IF (K,EQ,IPI(19)) GO TO 370
590  360 CONTINUE
      GO TO 390
      370 CALL DIRECT (DX1,DY1,DZ1), RETURNS(570)
      C *****
      RESET=1.0
      WRITE(6,800) K,ENG2,POL,AZI
      ENG2=0.0
      ISTOP = 0
      CALL PAC3 (A,52.1,52.6,32.6)
      X(ID) = A
      VX(ID) = A
      GO TO 390
      380 WRITE (6,810) K,ET2
      WRITE (6,790) J,X11,Y11,Z11
      ENG2M = AMAX1(ENG2M,ENG2)
600  390 IF (ENG2,LT,ENG2M) MEA = 10
      IF (DEFECT,AND,TEMP,NE,0.0) GO TO 400
      IF (K,EW,IPI(INE),OR,ID,EQ,MEAL) GO TO 400
      IF (ENG2,GT,ECUT) GO TO 400
      IF (.NOT,DEFECT,AND,UR,LT,CL) GO TO 570
610  C
      C *****
      C CHARACTERIZE ATOMS SIGNIFICANTLY DISPLACED FROM THEIR LATTICE SITE
      C *****
      C *****
615  C
      C 400 IF (FPAE) WRITE (4) TIME,K,L,Dx1,DY1,DZ1,ENG2,POL,AZI
      IF (ID,EW,MEAL,OR,K,LT,IPI(INE)) GO TO 450
      IF (ENG2,GT,ECUT) GO TO 450
      IF (DEFECT) GO TO 450
      IF (.NOT,REQUE) GO TO 445
620  DO 440 IN=1,20
      IF (K,EQ,IPI(19)) GO TO 450
      440 CONTINUE
      445 IF (DX,LT,CN) GO TO 570
      C
      C WRITE DATA FOR THE MOST SIGNIFICANT ATOMS AT TIME T
      C *****
      C *****

```

```

      COMMENT 564
      COMMENT 565
      COMMENT 566
      COMMENT 567
      COMMENT 568
      COMMENT 569
      COMMENT 570
      COMMENT 571
      COMMENT 572
      COMMENT 573
      COMMENT 574
      COMMENT 575
      COMMENT 576
      COMMENT 577
      COMMENT 578
      COMMENT 579
      COMMENT 580
      COMMENT 581
      COMMENT 582
      COMMENT 583
      COMMENT 584
      COMMENT 585
      COMMENT 586
      COMMENT 587
      COMMENT 588
      COMMENT 589
      COMMENT 590
      COMMENT 591
      COMMENT 592
      COMMENT 593
      COMMENT 594
      COMMENT 595
      COMMENT 596
      COMMENT 597
      COMMENT 598
      COMMENT 599
      COMMENT 600
      COMMENT 601
      COMMENT 602
      COMMENT 603
      COMMENT 604
      COMMENT 605
      COMMENT 606
      COMMENT 607
      COMMENT 608
      COMMENT 609
      COMMENT 610
      COMMENT 611
      COMMENT 612
      COMMENT 613
      COMMENT 614
      COMMENT 615
      COMMENT 616
      COMMENT 617
      COMMENT 618
      COMMENT 619
      COMMENT 620

```

5-15

```

450 IF (DEFECT.AND.MINATUR(XDM),41.120.) GO TO 570
CALL DIRECT (DX,X,DY,Y,DZ,Z), RETURNS(570)
630 C *****
      NP = NPOTA
      IF (ALLOY) NP=NPOTAB
      DX2 = DX*CLP(NP)
      DY2 = DY*CLP(NP)
635 DZ2 = DZ*CLP(NP)
      DX2 = DX1+UDCC/CLP(NP)
      DY = TBX1-XIPS
      DZ = TBZ1-ZIPS
640 DX3 = XTRAN(DX,DY,DZ)
      DY3 = YTRAN(DX,DY,DZ)
      DZ3 = ZTRAN(DY,DZ)
      DB = DB*UDCC
      IF (LPRNT.AND.ID.EW.MEA1) WRITE(6,820)
645 IF (LPRNT) WRITE(6,870)
      IF (LPRNT) WRITE(6,830) K,L,DX1,DY1,DZ1,DX2,DY2,DZ2,DB,ENG2,
      1 ENGPI(ID)
      IF (LPRNT) WRITE(6,840) DX3,DY3,DZ3
      IF (LPRNT) WRITE(6,870)
650 IF (.NOT.DEFECT) CALL IDDEF (DB,ENG2,ECUT,IPI(INE),UDCC,NB,LPRNT,
      1 LPRNT1,LPRNT2)
      C *****
      570 CONTINUE
      C
655 IF (LINDEF) CALL USLOAN (RELAX)
      IF (LINDEF) RELAX = .FALSE.
      IF (ITO.EW.1) GO TO 580
      CALL SECUND (T2)
      TT1 = T2-11
660 WRITE (6,1040) TT1
      ITO = 1
      C
      C *****
      C PREPARE FOR THE NEXT TIME STEP
      C *****
665 C *****
      C
      580 IF (ITIME.NE.1) MEA1 = MEA
      II = IDENT(MEA)
      CALL MTBIN (II,K,L,M,J)
670 A = X(MEA)
      KEA = K
      CALL UNPACS (A,XEA,YEA,ZEA)
      IF (LPRNT) WRITE (6,950) MEA,K,L,M,J,XEA,YEA,ZEA,ENG2M,ENGPI(MFA)
      IF (ISTOP.GT.0) GO TO 600
675 IF (.NOT.DEFECT.AND.ITIME.GT.ITSS.AND.ENG2M.LT..001) GO TO 600
      C
      IF (ITIME.NE.ITSS) GO TO 585
      IF (CONT.AND.ITSS.EW.ITSS) PEU = PEU1
      IF (ITIME.EQ.ITSS) PED=0.
680 SPE=SPE+PED
      PEU=ENGPI1
      ZPE=ZPE+SPE+PED
      505 ENGSP=ENGSP+ZPE
      PED=ENGPI1

```

```

COMMENT 621
COMMENT 622
COMMENT 623
COMMENT 624
COMMENT 625
COMMENT 626
COMMENT 627
COMMENT 628
COMMENT 629
COMMENT 630
COMMENT 631
COMMENT 632
COMMENT 633
COMMENT 634
COMMENT 635
COMMENT 636
COMMENT 637
COMMENT 638
COMMENT 639
COMMENT 640
COMMENT 641
COMMENT 642
COMMENT 643
COMMENT 644
COMMENT 645
COMMENT 646
COMMENT 647
COMMENT 648
COMMENT 649
COMMENT 650
COMMENT 651
COMMENT 652
COMMENT 653
COMMENT 654
COMMENT 655
COMMENT 656
COMMENT 657
COMMENT 658
COMMENT 659
COMMENT 660
COMMENT 661
COMMENT 662
COMMENT 663
COMMENT 664
COMMENT 665
COMMENT 666
COMMENT 667
COMMENT 668
COMMENT 669
COMMENT 670
COMMENT 671
COMMENT 672
COMMENT 673
COMMENT 674
COMMENT 675
COMMENT 676
COMMENT 677

```

5-16

685	TE = ENGPU+ENGPP+ENGKA+ELOS	COMMENT	678
	IF (IIME.EQ.ITS) TED = TE	COMMENT	679
	IF (CONT.AND.ITS.EQ.ITS) TEU=TEU1	COMMENT	680
	TEDIF = TE-TEU	COMMENT	681
	IF (LPRNT) WRITE (6,940) TIME,ENGPD,ENGSP,ENGKA,ELOS,TE,TEDIF,RESE	COMMENT	682
690	*T	COMMENT	683
	IF (MINATUR(XDM).LE.60) GO TO 600	COMMENT	684
	LPRNT = .FALSE.	COMMENT	685
	IF (DEFECT.AND.TEMP.EQ.0.0.AND.4IIME.EQ.ITS+50) DTN = DTN*5.	COMMENT	686
	IF (DEFECT.AND.TEMP.EQ.0.0.AND.UTN.NE.DT0) GO TO 110	COMMENT	687
695	IF (DEFECT) GO TO 160	COMMENT	688
	IF (IIME.GT.ITS.AND.ENG2M.LT.ESET.AND.RESET.EQ.0.) RESET = 2.0	COMMENT	689
	IF (RESET.GT.0.AND.RESET.LT.3.) GO TO 590	COMMENT	690
	GO TO 160	COMMENT	691
590	IF (IC.GT.9) GO TO 600	COMMENT	692
700	C	COMMENT	693
	C DETERMINE THE RADIUS OF THE NEAR MINI-LATTICE	COMMENT	694
	C *****	COMMENT	695
	C	COMMENT	696
	ERLI = ERLI+DEML(IC+1)	COMMENT	697
705	ERLI = ERLI+UDCC	COMMENT	698
	IC = IC+1	COMMENT	699
	ITS = IIME+1	COMMENT	700
	IF (RESET.EQ.2.) RESET = 3.	COMMENT	701
	IF (RESET.EQ.1.) RESET = 0.	COMMENT	702
710	600 IF (DEFECT) GO TO 605	COMMENT	703
	DTN = SQRT(VECT**2/20.)	COMMENT	704
	IF (ENG2M.GE.10.) DTN = SQRT(VECT**2/(2.*ENG2M))	COMMENT	705
	IF (ENG2M.LE.2.) DTN = SQRT(VECT**2/4.)	COMMENT	706
	LPRNT = .FALSE.	COMMENT	707
715	C	COMMENT	708
	C RE=ESTABLISH THE INITIAL ORDER OF ATOM IDENTIFICATION AND	COMMENT	709
	C COORDINATE ARRAYS	COMMENT	710
	C *****	COMMENT	711
	C	COMMENT	712
720	IF (MINATUR(XDM).GT.60.AND.SL.EQ.5.) GO TO 110	COMMENT	713
605	REWIND 3	COMMENT	714
	DO 610 I=1,NS	COMMENT	715
	WRITE (3) X(I),VX(I),TB(I),IDENT(I),ENGPI(I)	COMMENT	716
725	610 CONTINUE	COMMENT	717
	REWIND 3	COMMENT	718
	DO 620 I=1,NS	COMMENT	719
	READ (3) U1,D2,U3,IOL,D4	COMMENT	720
	II = IOL	COMMENT	721
730	CALL MTHIN (II,K,L,M,J)	COMMENT	722
	X(K) = D1	COMMENT	723
	VX(K) = D2	COMMENT	724
	TB(K) = D3	COMMENT	725
	ENGPI(K) = D4	COMMENT	726
	IF (M.GT.3000) M=M-3000	COMMENT	727
735	IF (M.GT.40.AND.M.LT.100) M = 1	COMMENT	728
	CALL FILBIN (III,K,L,M,J)	COMMENT	729
	IDENT(K) = III	COMMENT	730
620	CONTINUE	COMMENT	731
	IP = IPS	COMMENT	732
740	N = NS	COMMENT	733
	NL = NLS	COMMENT	734

N12 = N125
 YEA = K25
 IF (ISTOP.GI.0) GO TO 630
 IF (MINATOR(XUM).LE.60) GO TO 630
 IF (IC.LT.11) GO TO 110
 C 630 IF (IELE.GI.1) CALL ALUDORD (FA,FAA,FB,FB8,S,SU,INSERT,NPOTAB)
 C
 C WRITE THE FINAL STATE OF THE LATTICE
 C *****
 C
 IF (FTAPE) ENDFILE 4
 WRITE (6,1050)
 DU 660 I=1,N
 A = XLI)
 CALL UNPAC3 (A,X1,Y1,Z1)
 A = Td(I)
 CALL UNPAC3 (A,IX1,IY1,IZ1)
 I1 = IDENT(I)
 CALL MTBIN (I1,K,L,M,J)
 DX = X1-XIPS
 DY = Y1-YIPS
 DZ = Z1-ZIPS
 DX1 = X1-IX1
 DY1 = Y1-IY1
 DZ1 = Z1-IZ1
 DR1 = RDIAT(DX,DY,DZ1)
 DR2 = DR1*UDCC/CLP(INPUTA)
 DX2 = XIRAN(DX,DY,DZ)
 DY2 = YIRAN(DX,DY,DZ)
 DZ2 = ZIRAN(DY,DZ)
 IF (I.LE.NL) GO TO 640
 WRITE (6,730) I,K,L,M,J,X1,Y1,Z1,DX2,DY2,DZ2,DR2
 GO TO 630
 640 WRITE (6,730) I,K,L,M,J,X1,Y1,Z1,DX2,DY2,DZ2,DR2,ENGPI(I)
 650 IF (.NOT.FIAPE) GO TO 660
 IF (DM1.LI.CL) GO TO 660
 DX = IX1-XIPS
 DY = IY1-YIPS
 DZ = IZ1-ZIPS
 DX2 = XIRAN(DX,DY,DZ)
 DY2 = YIRAN(DX,DY,DZ)
 DZ2 = ZIRAN(DY,DZ)
 WRITE (4) TIME,I,DX2,DY2,DZ2
 660 CONTINUE
 C
 C ANALYZE THE FINAL THERMAL STATE OF THE LATTICE
 C *****
 C
 GENVS = .FALSE.
 IF (TEMP.NE.0.) CALL THRMAL (GENVS)
 IF (DEFECT) KEA = IP
 IF (XEA1.EQ.0.0) XEA1 = XIPS
 IF (YEA1.EQ.0.0) YEA1 = YIPS
 IF (ZEA1.EQ.0.0) ZEA1 = ZIPS
 C
 C ANALYZE FINAL DISLOCATION CONFIGURATION

COMMENT 735
 COMMENT 736
 COMMENT 737
 COMMENT 738
 COMMENT 739
 COMMENT 740
 COMMENT 741
 COMMENT 742
 COMMENT 743
 COMMENT 744
 COMMENT 745
 COMMENT 746
 COMMENT 747
 COMMENT 748
 COMMENT 749
 COMMENT 750
 COMMENT 751
 COMMENT 752
 COMMENT 753
 COMMENT 754
 COMMENT 755
 COMMENT 756
 COMMENT 757
 COMMENT 758
 COMMENT 759
 COMMENT 760
 COMMENT 761
 COMMENT 762
 COMMENT 763
 COMMENT 764
 COMMENT 765
 COMMENT 766
 COMMENT 767
 COMMENT 768
 COMMENT 769
 COMMENT 770
 COMMENT 771
 COMMENT 772
 COMMENT 773
 COMMENT 774
 COMMENT 775
 COMMENT 776
 COMMENT 777
 COMMENT 778
 COMMENT 779
 COMMENT 780
 COMMENT 781
 COMMENT 782
 COMMENT 783
 COMMENT 784
 COMMENT 785
 COMMENT 786
 COMMENT 787
 COMMENT 788
 COMMENT 789
 COMMENT 790
 COMMENT 791

5-18

```

      C      IF (LINDEX) CALL USLOAN (RELAX)                                COMMENT 792
800      C      SAVE THE FINAL STATE OF THE LATTICE FOR LATER USE          COMMENT 793
      C      *****                                                    COMMENT 794
      C      *****                                                    COMMENT 795
      C      *****                                                    COMMENT 796
      C      *****                                                    COMMENT 797
805      C      WRITE (1) ITIME,N,NL,UT,ITS,TIME,KEA,DYN,TEMP,ENLI,RESET,IC,JP,ELD COMMENT 798
      C      +S,XEA1,YEA1,ZEA1,NA,NPOTA,NPOTD,NPOTAB,IELE,FMA,FMB,UT,FA,FAA,FB,F COMMENT 799
      C      +BB,S,NE,TEO,PEO,PEPA,LPE                                     COMMENT 800
      C      WRITE (1) (TB(I),I=1,N)                                       COMMENT 801
      C      WRITE (1) (X(I),I=1,N)                                       COMMENT 802
810      C      WRITE (1) (VX(I),I=1,N)                                     COMMENT 803
      C      WRITE (1) (IDEN(I),I=1,N)                                     COMMENT 804
      C      WRITE (1) SFM,MXC,MYC,HZC,VGX,VGY,VCZ,PXC,PYC,PZC          COMMENT 805
      C      WRITE (1) AX,AY,AZ                                           COMMENT 806
      C      ENOFIL 1                                                       COMMENT 807
815      C      IF (FTAPE) ENOFIL 4                                         COMMENT 808
      C      *****                                                    COMMENT 809
      C      *****                                                    COMMENT 810
820      C      STOP                                                         COMMENT 811
      C      *****                                                    COMMENT 812
      C      *****                                                    COMMENT 813
      C      *****                                                    COMMENT 814
      C      *****                                                    COMMENT 815
      C      *****                                                    COMMENT 816
825      C      FORMAT ('11,10X,'CONVERSION UNITS'/, ' UO(M), UNIT DISTANCE..... COMMENT 817
      C      +.....',E12.5/, ' UE(JOULES), UNIT ENERGY.....',E12.5/, ' U COMMENT 818
      C      +M(KGM), UNIT MASS.....',E12.5/, ' UV(M/S), UNIT VFLOCIT COMMENT 819
      C      +Y.....',E12.5/, ' UT(S), UNIT TIME.....',E12 COMMENT 820
      C      +.5/, ' UF(N), UNIT FORCE.....',E12.5/, ' UMC(N/M), UNIT COMMENT 821
      C      +HUKIAN CONST.....',E12.5/, ' UVC(N/(M/S)), UNIT VISCOUS CONST... COMMENT 822
830      C      +.5/,E12.5/)                                               COMMENT 823
      C      FORMAT ('01ME FOLLOWING DEFECTS ARE INSERTED:1/)             COMMENT 824
835      C      710 FORMAT (8A10)                                           COMMENT 825
      C      720 FORMAT ('01,8A10)                                         COMMENT 826
      C      730 FORMAT ('01TIME =',15,2X,30('1')//T9,'1D',T29,'1X',T41,'1Y',T53,'1Z', COMMENT 827
      C      +T64,'1DX',T76,'1DY',T88,'1DZ',T101,'1H',T112,'1KE',T124,'1PE',T23,'1L'//) COMMENT 828
      C      740 FORMAT ('01ND MINI-LATTICE WAS GENERATED'/, ' NUMBER OF ATOMS IN THE COMMENT 829
      C      +LATTICE IS',I6/, ' NUMBER OF ATOMS WITH BOUNDARY TYPE LESS THAN 10', COMMENT 830
      C      +IS)                                                         COMMENT 831
      C      750 FORMAT ('////', ' NUMBER OF TIMES A MINI-LATTICE HAS BEEN GENERATED IS COMMENT 832
840      C      +',T80,I6/, ' TIME STEP AT WHICH THIS MINI-LATTICE WAS GENERATED IS', COMMENT 833
      C      +T80,I6/, ' ID OF MINI-LATTICE SITE CORRESPONDING TO INITIAL CENTER O COMMENT 834
      C      +F COORDINATES IS',T80,I6/, ' RADIUS OF THIS MINI-LATTICE IS',T76,F10 COMMENT 835
      C      +.5////)                                                    COMMENT 836
845      C      760 FORMAT (T7,'1I',T14,'1K',T21,'1L',T28,'1M',T35,'1J',T44,'1YI',T49,'1ZI',T COMMENT 837
      C      +55,'1RI',T64,'1X',T74,'1Y',T84,'1Z',T90,'1R',T99,'1XC',T109,'1YC',T119,'1Z COMMENT 838
      C      +C',T129,'1PE',T39,'1X1'//)                                  COMMENT 839
      C      770 FORMAT (5I7,3F5.1,F6.2,3F10.3,F6.2,4F10.4)              COMMENT 840
850      C      780 FORMAT ('01, T40, 'ATOM ID =',I5, ' IS AT THE CRYSTAL BOUNDARY WITH T COMMENT 841
      C      +00 MUCH ENERGY',E12.4, ' KINETIC AND',E12.4, ' POTENTIAL = MUST TE COMMENT 842
      C      +RMINATE THE CALCULATION,')                                  COMMENT 843
      C      790 FORMAT (' THIS ATOM IS IN BOX NUMBER',I10, ' AND HAS COMMENT COORDIN COMMENT 844
      C      +ATES, X=',F5.2, ' , Y=',F5.2, ' , AND Z=',F5.2, ' .') COMMENT 845
855      C      800 FORMAT ('01, T40, 'ATOM',I10, ' IS ALLOWED TO LEAVE THE CRYSTAL LATI COMMENT 846
      C      +CE REGION UNDER ANALYSIS',T40, ' AND IS ASSUMED TO FORM AN INTERSTIT COMMENT 847
      C      +AL ELSEWHERE IN THE CRYSTAL.',T40, ' ITS KINETIC ENERGY IS',E12.4, ' COMMENT 848

```

```

      + AND ITS POLAR AND AZIMUTHAL /140,1 ANGLES ARE /OPF12.4,1 AND /F12. COMMENT 0649
      +4,1, RESPECTIVELY.1) COMMENT 0650
      810 FORMAT ('0',T40,'CAUTION -- ATOM ID',I5,1 IS AT THE CRYSTAL BOUNDARY COMMENT 0651
      +RY WITH A TOTAL ENERGY OF',1PE12.4) COMMENT 0652
      820 FORMAT ('0',T40,'MOST ENERGETIC ATOM AT T=DT --',1) COMMENT 0653
      830 FORMAT ('I10,1 (PRESENT)',I3,F4.4,OPF12.4) COMMENT 0654
      840 FORMAT ('0X,1 (INITIAL)',3F12.4) COMMENT 0655
      870 FORMAT (/) COMMENT 0656
      930 FORMAT (5I7,8F11.4) COMMENT 0657
      865 940 FORMAT ('0X,1 TIME',10X,'ENGPO',10X,'ENG9P',10X,'ENGKA',12X,'ELOS',1 COMMENT 0658
      +2X,'TE',7X,'TEDIF',6X,'RESET'/L1P7E15.6,OPF7.3)) COMMENT 0659
      950 FORMAT ('I (MEA), K, L, M, J, XC, YC, ZC, KE, PE =',5I7,5F4.3) COMMENT 0660
      960 FORMAT ('0',T40,'THE INPUT TEMPERATURE FOR THIS CALCULATION DOES NOT COIN COMMENT 0661
      +CIDE WITH THE TEMPERATURE OF THE STUNED LATTICE. ('/F7.2,1)1/1 MUST COMMENT 0662
      870 + ABORT THE CALCULATION.1) COMMENT 0663
      1030 FORMAT ('0',T40,'CAUTION -- DTN WAS NEITHER INPUT NOR CALCULATED - MUST A COMMENT 0664
      +BORT THIS CALCULATION.1) COMMENT 0665
      1040 FORMAT ('0',T40,'TIME FOR FIRST INTEGRATION I=1,F10.5,1 SEC.1) COMMENT 0666
      1050 FORMAT ('0',T7,'I',T14,'K',T21,'L',T28,'M',T35,'J',T43,'X',T54,'Y COMMENT 0667
      +C',T65,'Z',T76,'A',T87,'Y',T98,'Z',T110,'DX',T120,'PE'/) COMMENT 0668
      1060 FORMAT('0',T40,'BINARY ALLOY CONSTITUENT FRACTIONS FA AND FB ARE ZERO AND COMMENT 0669
      1 HENCE IMPROPERLY SPECIFIED.1) COMMENT 0670
      1070 FORMAT('0',T40,'IN THIS BINARY ALLOY THE FRACTION OF A ATOMS IS FA=F6.3 COMMENT 0671
      1,1 OF WHICH FAa=F6.3/1 ARE ON A-SITES, AND THE FRACTION OF B ATOM COMMENT 0672
      880 1MS IS FB=F6.3,1 OF WHICH FBa=F6.3,1 ARE ON B-SITES.1/1 THE RES COMMENT 0673
      1ULTING LONG RANGE ORDER PARAMETER FOR THE L1-SUB 2 SUPER LATTICE I COMMENT 0674
      1S S=F6.3) COMMENT 0675
      END COMMENT 0676

```

```

1      SUBROUTINE ALUYORD (FA,FAA,FB,FBB,S,SO,INSERT,NPOT)
      C
      C--- THIS ROUTINE GENERATES AN ALLOY CRYSTAL AND ANALYZES THE NEAREST
      C      NEIGHBOR DISTRIBUTION FOR THE ATOMS WITH M=1
      C
      C--- FA IS THE FRACTION OF ATOMS OF TYPE A
      C      FAA IS THE FRACTION OF A-ATOMS AT A-SITES
      C      FB IS THE FRACTION OF ATOMS OF TYPE B
      C      FBB IS THE FRACTION OF B-ATOMS AT B-SITES
      C      S IS THE LONG RANGE ORDER PARAMETER
      C      SO IS A SWITCH VARIABLE WHICH HAS THE FOLLOWING VALUES
      C          .EW. 1 IF ONLY CHARGES IN STOICHIOMETRY EFFECT CHANGES
      C              IN S, WHICH IS TO BE CALCULATED.
      C          .NE. 1 OTHERWISE
      C      INSERT IS A LOGICAL VARIABLE WHICH HAS THE VALUES
      C          .TRUE. IF THE ATOM SITES ARE TO BE FILLED AND ANALYZED.
      C          .FALSE. IF ANALYSIS ONLY IS TO BE PERFORMED.
      C      NPOT IS THE POTENTIAL FUNCTION INDEX
      C
      C
      C      COMMON NMIN, DT, N, NL, VX(6460), YB(6460), TEMP, DEFECT,
      C      + DTN, NS, NLS, NA, UVC, UHC, UF
      C      COMMON /ALPHA/ X(6460), ZZX(6460), ZZY(6460), ZZZ(6460),
      C      + ENGP1(6460), IDENT(6460), IBUX(14600), EA(1001), EEA(1001),
      C      + EB(1001), EEB(1001), EAB(1001), EAB(1001)
      C      COMMON /ELMT/ ENGPU, ID, E1, E4, J3, DIV, RMAX2, INIB, N12, ML,
      C      + IELE, L, LT, IJT
      C      LEVEL 2, X, ZZX, ZZY, ZZZ, ENGP1, IDENT, IBUX, EA, EEA, EB, FEB,
      C      + EAB, EEB
      C      COMMON /SPFRS/ NI, IUC, ID1, ID2, ID3, ID4, ID5, ID6, NSF, ST, HXC, HYC, HZC,
      C      + VCX, VCY, VCZ, LPRNT2, NB
      C      COMMON /PUTO/ RM, ME, CLP(12), RMN(12), MMX(12), SH, AP, RP, UDCC, ZA, ZB,
      C      + RMAX
      C      COMMON /BUX/ IK(17,125)
      C
      C      LOGICAL LKEY, INSERT, SONE1, SHIF1, FULMIX, INTMIX, LPRNT2
      C      INTEGER AA, BB, NDATUM(2), IDS(4)
      C      INTEGER AABIN(14), ABBIN(14), BABIN(14), BBBIN(14)
      C      REAL FAABIN(14), FABIN(14), FBABIN(14), FBBBIN(14)
      C
      C--- NOASLB IS THE NUMBER OF ATOMS PER (X,Y)-PLANE IN THE CRYSTAL
      C      NUSLPB IS THE NUMBER OF (X,Y)-PLANES PER PERIODIC BLOCK IN
      C      THE CRYSTAL
      C      NUENPB IS THE NUMBER OF END PLANES FOR THE MODEL CRYSTAL
      C      DATA NUASLB, NUSLPB, NUENPB/164,5,8/
      C
      C--- FACTR IS THE MULTIPLIER FOR CLP(NPOT) IN COMPUTING THE DISTANCE
      C      TO NEIGHBORING ATOMS.
      C      DATA FACTR/.8/
      C
      C--- FULMIX = .TRUE. IF ALL ATOMS IN THE CRYSTAL ARE USED IN
      C      COMPUTING FA AND FB.
      C          .FALSE. IF ONLY INTERIOR ATOMS ARE USED IN COMPUTING
      C      FA AND FB.
      C
      C      DATA FULMIX/.FALSE./
      C
      C      DATA AA,BB/1HA,1HB/

```


5-21

```

C      NSLB = N/NOASLB                      ALUYORD 51
60      NMSB = NSLB/NOASLB-1                  ALUYORD 52
      IF (N.NE.NOASLB*(NMSB*NOSLPP+NUENUP)) WRITE (6,600) ALUYORD 53
      JPSET = NUENUP/2-2                      ALUYORD 54
      JPTEST = NSLB-JPSET-1                  ALUYORD 55
C      C--- SETUP IDBX(,) ARRAY              ALUYORD 56
65      DU 20 I=1,NB                          ALUYORD 57
      IDBX(I) = 0                             ALUYORD 58
      DU 30 I=1,M                             ALUYORD 59
      ID = I                                  ALUYORD 60
70      II = IDENT(I)                         ALUYORD 61
      CALL MTBIN (II,K,L,M,J)                 ALUYORD 62
      A = X(I)                               ALUYORD 63
      CALL UNPACS (A,X1,Y1,Z1)                ALUYORD 64
      J = IFIX(X1-31.)*31+IFIX(Y1-32.)*461+IFIX(Z1-32.) ALUYORD 65
75      CALL FILBIN (II,K,L,M,J)              ALUYORD 66
      IDENT(II) = II                         ALUYORD 67
      J1 = IDBX(J)                           ALUYORD 68
      CALL INBIN (I2,J1,I0)                  ALUYORD 69
      IDBX(J) = I2                           ALUYORD 70
80      C                                     ALUYORD 71
      DIST2 = (FACIT*CLP(INPUT)/UDCC)*2      ALUYORD 72
      INTMIX = .NOT.FULMIX                   ALUYORD 73
C      C--- MA = NO. OF A-ATOMS              ALUYORD 74
85      MAS = NO. OF A-ATOMS                 ALUYORD 75
      C   MAA = NO. OF A-ATOM SITES OCCUPIED BY A-ATOMS ALUYORD 76
      MB = NO. OF B-ATOMS                   ALUYORD 77
      C   MBS = NO. OF B-ATOM SITES OCCUPIED BY B-ATOMS ALUYORD 78
      MBB = NO. OF B-ATOM SITES OCCUPIED BY B-ATOMS ALUYORD 79
90      C                                     ALUYORD 80
      ISAVE = 0                             ALUYORD 81
      MA = 0                                ALUYORD 82
      MAA = 0                               ALUYORD 83
      MBS = 0                               ALUYORD 84
95      MB = 0                               ALUYORD 85
      MBB = 0                               ALUYORD 86
      MAS = 0                               ALUYORD 87
C      C                                     ALUYORD 88
      IF (.NOT.INSEM) GO TO 490              ALUYORD 89
100      C                                     ALUYORD 90
      C--- FIRST FUNCTION = - - - GENERATE CRYSTAL TO SPECIFICATIONS ALUYORD 91
C      C--- FIAA IS THE FRACTION OF IDEAL A-SITES OCCUPIED BY A-ATOMS ALUYORD 92
      PRIOR TO INTRODUCING DISORDER          ALUYORD 93
C      FIBB IS THE FRACTION OF IDEAL B-SITES OCCUPIED BY B-ATOMS ALUYORD 94
      PRIOR TO INTRODUCING DISORDER          ALUYORD 95
105      FIAA = FA/.75                       ALUYORD 96
      FIBB = FB/.25                         ALUYORD 97
      SONE1 = SU.NE.1.                      ALUYORD 98
      TESTS = AMAX1(JOUI,S)                 ALUYORD 99
110      SNEW = 2.                           ALUYORD 100
      FLTNM1 = N-1                          ALUYORD 101
      ALUYORD 102
      ALUYORD 103
      ALUYORD 104
      ALUYORD 105
      ALUYORD 106
      ALUYORD 107

```

5-22

115	NPMPH = 344	ALUYORD 108
	C	ALUYORD 109
	C--- FIRST STAGE = - - GENERATE LATTICE WITH APPROXIMATE ATOM MIX	ALUYORD 110
	C AND DISORDER	ALUYORD 111
	JPLANE = 0	ALUYORD 112
120	I = 0	ALUYORD 113
	SHIFT = .TRUE.	ALUYORD 114
	DU 140 IPLANE=1,N3LD	ALUYORD 115
	IATOM = 0	ALUYORD 116
	IF (IPLANE.EQ.JPTEST) JPLANE = JPSET	ALUYORD 117
125	JPLANE = JPLANE+1	ALUYORD 118
	C	ALUYORD 119
	C--- NOTE.....IF NUSLPD IS CHANGED THE FOLLOWING COMPUTED GO TO	ALUYORD 120
	C MUST ALSO BE CHANGED	ALUYORD 121
	GU TO (50,60,70,60,50,60,40), JPLANE	ALUYORD 122
130	C	ALUYORD 123
	40 JPLANE = 1	ALUYORD 124
	SHIFT = .NOT.SHIFT	ALUYORD 125
	50 IBATOM = 1	ALUYORD 126
	IF (SHIFT) IBATOM = 2	ALUYORD 127
135	GU TO 80	ALUYORD 128
	60 IBATOM = 165	ALUYORD 129
	GU TO 80	ALUYORD 130
	70 IBATOM = 2	ALUYORD 131
	IF (SHIFT) IBATOM = 1	ALUYORD 132
140	C--- IBATOM IS THE INDEX OF THE FIRST B-ATOM ON THE PLANE JPLANE	ALUYORD 133
	80 LKEY = .FALSE.	ALUYORD 134
	90 I = I+1	ALUYORD 135
	IATOM = IATOM+1	ALUYORD 136
	ID = IDENT(I)	ALUYORD 137
145	CALL MTBIN (ID,K,L,M,J)	ALUYORD 138
	L = 1	ALUYORD 139
	IF (FULMIX) MA = MA+1	ALUYORD 140
	IF (IATOM.LT.IBATOM) GO TO 110	ALUYORD 141
	IF (LKEY) GO TO 100	ALUYORD 142
150	L = 2	ALUYORD 143
	IF (INTMIX) GU TO 100	ALUYORD 144
	MA = MA+1	ALUYORD 145
	MB = MB+1	ALUYORD 146
	100 LKEY = .NOT.LKEY	ALUYORD 147
155	110 IF (M.GE.10) GO TO 130	ALUYORD 148
	XIRN = KANF(NUUM)	ALUYORD 149
	IF (L.EQ.2) GO TO 120	ALUYORD 150
	C	ALUYORD 151
	C--- A-ATOM SITE	ALUYORD 152
160	K = 1	ALUYORD 153
	IF (INTMIX) MA = MA+1	ALUYORD 154
	MAA = MAA+1	ALUYORD 155
	MAS = MAS+1	ALUYORD 156
	IF (XIRN.LE.FIAA) GO TO 130	ALUYORD 157
165	C--- SET FOR B-ATOM AT THIS IDEAL-LATTICE A-SITE	ALUYORD 158
	L = 2	ALUYORD 159
	MB = MB+1	ALUYORD 160
	MA = MA+1	ALUYORD 161
	MAA = MAA+1	ALUYORD 162
170	GU TO 140	ALUYORD 163
	C	ALUYORD 164

5-23

	C--- B-ATOM SITE	ALUYORD 165
	120 K = 2	ALUYORD 166
175	IF (INTNIA) MD = MD+1	ALUYORD 167
	MBB = MBB+1	ALUYORD 168
	MBS = MBS+1	ALUYORD 169
	IF (XINN,LE,FIRS) GO TO 130	ALUYORD 170
	C--- SET FOR A-ATOM AT THIS IDEAL-LATTICE B-SITE	ALUYORD 171
	L = 1	ALUYORD 172
180	MA = MA+1	ALUYORD 173
	MD = MD+1	ALUYORD 174
	MBB = MBB+1	ALUYORD 175
185	130 CALL FILMIN (ID,K,L,M,J)	ALUYORD 176
	IDENT(I) = ID	ALUYORD 177
	IF (IATOM,LT,NUASLB) GO TO 90	ALUYORD 178
	140 CONTINUE	ALUYORD 179
	C	ALUYORD 180
	C--- CALCULATE GENERATED CRYSTAL PARAMETERS	ALUYORD 181
	NORNDS = 0	ALUYORD 182
190	ITEST = 0	ALUYORD 183
	FABNEW = MA+MB	ALUYORD 184
	150 FANEW = FLOAT(MA)/FABNEW	ALUYORD 185
	FBNW = 1.-FANEW	ALUYORD 186
	C	ALUYORD 187
195	160 FAANEW = 0.	ALUYORD 188
	IF (MAS,GT,0) FAANEW = FLOAT(MAA)/FLOAT(MAS)	ALUYORD 189
	FBBNEW = 0.	ALUYORD 190
	IF (MBS,GT,0) FBBNEW = FLOAT(MBB)/FLOAT(MBS)	ALUYORD 191
	OLDS = SNEW	ALUYORD 192
200	SNEW = 0.	ALUYORD 193
	IF (FBNW,GT,0.) SNEW = .75*(FAANEW-FANEW)/FBNW	ALUYORD 194
	IF (FANEW,GT,0.) SNEW = .25*(FBBNEW-FBNEW)/FANEW+SNEW	ALUYORD 195
	C	ALUYORD 196
	C--- TEST GENERATED CRYSTAL PARAMETERS WITH SPECIFICATIONS	ALUYORD 197
205	IF (ITEST=1) 180,190,170	ALUYORD 198
	170 IF (ITEST=3) 220,240,280	ALUYORD 199
	C--- TEST ON MA	ALUYORD 200
	180 ITEST = 1	ALUYORD 201
	NATEST = PA*FANEW+.5	ALUYORD 202
210	MABDIK = NATEST-MA	ALUYORD 203
	190 IF (MA,NE,NATEST) GO TO 290	ALUYORD 204
	C	ALUYORD 205
	200 IF (SONE1) GO TO 230	ALUYORD 206
	I = FAA*FLOAT(MAS)+.5	ALUYORD 207
215	J = I-MAA	ALUYORD 208
	K = FBB*FLOAT(MBS)+.5	ALUYORD 209
	M = K-MBB	ALUYORD 210
	IF (IABS(J)-IABS(M)) 210,210,230	ALUYORD 211
	C--- TEST ON MAA	ALUYORD 212
220	210 MABTST = 1	ALUYORD 213
	MABDIK = ISIGN(1,J)	ALUYORD 214
	ITEST = 2	ALUYORD 215
	IF (MAA-MABTST) 240,480,290	ALUYORD 216
	C--- TEST ON MBB	ALUYORD 217
225	230 MABTST = K	ALUYORD 218
	MABDIK = ISIGN(1,M)	ALUYORD 219
	ITEST = 3	ALUYORD 220
	240 IF (MBB-MABTST) 240,480,290	ALUYORD 221

	C--- TEST ON SNEW	ALUYORD	222
230	250 ITEST = 4	ALUYORD	223
	STEST = TESTS-SNEW	ALUYORD	224
	MABDIK = SIGN(1.25,STEST)	ALUYORD	225
	IF (STEST) 260,460,270	ALUYORD	226
	260 IF (MAA.LE.0) GO TO 480	ALUYORD	227
235	IF (MAB.LE.0) GO TO 480	ALUYORD	228
	GO TO 290	ALUYORD	229
	270 IF (MAA.GE.MAS) GO TO 480	ALUYORD	230
	IF (MAB.GE.MBS) GO TO 460	ALUYORD	231
	GO TO 290	ALUYORD	232
240	280 IF (STEST*(TESTS-SNEW).LE.0.) GO TO 480	ALUYORD	233
	C--- ESTIMATE NEW VALUE FOR S	ALUYORD	234
	PHOJS = 2.*SNEW-OLD	ALUYORD	235
	IF (STEST*(TESTS-PHOJS).GT.0.) GO TO 290	ALUYORD	236
	IF (ABS(TESTS-PHOJS)-ABS(TESTS-SNEW)) 290,290,480	ALUYORD	237
245	C	ALUYORD	238
	C--- REFINE ATOM COUNTS AND THE LATTICE DISORDER	ALUYORD	239
	C	ALUYORD	240
	290 IF (NURND5.GE.NPNPN) GO TO 460	ALUYORD	241
	C	ALUYORD	242
250	C--- RANDOMLY SELECT A CANDIDATE ATOM	ALUYORD	243
	XIRN = RAND(NDUM)	ALUYORD	244
	NURND5 = NURND5+1	ALUYORD	245
	IF (NURND5.GE.NPNPN) GO TO 450	ALUYORD	246
	I = XIRN*FLTHM1+1.5	ALUYORD	247
255	IF (I.LE.0) GO TO 290	ALUYORD	248
	IF (I.GT.N) GO TO 290	ALUYORD	249
	300 IO = IDENT(I)	ALUYORD	250
	CALL MTBIN (IO,K,L,M,J)	ALUYORD	251
	IF (M.GE.10) GO TO 290	ALUYORD	252
260	IF (ITEST.GT.1) GO TO 350	ALUYORD	253
	C	ALUYORD	254
	C--- SECOND STAGE - - - REFINE MA AND MB COUNTS FOR SPECIFIED	ALUYORD	255
	STOICHIOMETRY	ALUYORD	256
	C	ALUYORD	257
	IF (MABDIK.GT.0) GO TO 310	ALUYORD	258
265	IF (L=1) 290,330,320	ALUYORD	259
	310 IF (L.LE.1) GO TO 290	ALUYORD	260
	320 IF (K.NE.2) GO TO 290	ALUYORD	261
	IF (MB.LE.0) GO TO 200	ALUYORD	262
	C--- REPLACE B-ATOM WITH AN A-ATOM	ALUYORD	263
270	L = 1	ALUYORD	264
	MA = MA+1	ALUYORD	265
	MB = MB-1	ALUYORD	266
	MAB = MAB-1	ALUYORD	267
	GO TO 340	ALUYORD	268
275	330 IF (K.NE.1) GO TO 290	ALUYORD	269
	IF (MA.LE.0) GO TO 200	ALUYORD	270
	C--- REPLACE A-ATOM WITH A B-ATOM	ALUYORD	271
	L = 2	ALUYORD	272
	MA = MA-1	ALUYORD	273
280	MB = MB+1	ALUYORD	274
	MAA = MAA+1	ALUYORD	275
	340 CALL FILBIN (II,K,L,M,J)	ALUYORD	276
	IDENT(II) = II	ALUYORD	277
285	GO TO 150	ALUYORD	278
	C		

C--- THIRD STAGE - - - UNDER-DISORDER OPERATION
 C CHOOSE ATOM TO BE EXCHANGED FOR A DIFFERENT TYPE
 350 IF (K.LE.0) GO TO 290
 IF (K.GT.2) GO TO 290
 290 C--- ALTER MAA AND/OR MBB
 IF (I(EST-3) .GT. 380,380,360)
 360 IF (MABDIK) 400,400,390
 370 IF (MABDIK.LE.0) GO TO 400
 IF (L-2) 290,390,290
 295 380 IF (MABDIK.LE.0) GO TO 400
 IF (L.NE.1) GO TO 290
 C--- TEST FOR INCREASING MAA AND MBB
 390 IF (L.EQ.K) GO TO 290
 IF (MAA.GE.MBB) GO TO 480
 300 IF (MBB.GE.MBB) GO TO 480
 GO TO 410
 C--- TEST FOR DECREASING MAA AND MBB
 400 IF (L.NE.K) GO TO 290
 IF (MAA.LE.0) GO TO 480
 305 IF (MBB.LE.0) GO TO 480
 C
 C--- NOW SELECT HANDFULLY A NEIGHBORING ATOM FOR EXCHANGE
 410 A = X(I)
 CALL UNPACK (A,XD,YD,ZD)
 310 J6 = 0
 DO 430 J1=1,125
 J2 = J+IK(1,J1)
 IF (J2.LE.0) GO TO 430
 IF (J2.GT.NB) GO TO 430
 315 J3 = IBUX(J2)
 IF (J3.EQ.0) GO TO 430
 CALL MTBIN (J3,ID8(1),ID8(2),ID8(3),ID8(4))
 DO 420 J5=1,4
 J7 = ID8(J5)
 320 IF (J7.LE.0) GO TO 420
 IF (J7.EQ.I) GO TO 420
 ID = IDENT(J7)
 CALL MTBIN (ID,KB,LB,MB,J8)
 IF (MB.GE.ID) GO TO 420
 325 IF (LB.EQ.L) GO TO 420
 IF (KB.EQ.K) GO TO 420
 IF (LB.LE.0) GO TO 420
 IF (LB.GT.2) GO TO 420
 A = X(J7)
 330 CALL UNPACK (A,X1,Y1,Z1)
 DIST = (XD-X1)**2+(YD-Y1)**2+(ZD-Z1)**2
 IF (DIST.GT.DIST2) GO TO 420
 IF (J6.GT.14) GO TO 420
 J6 = J6+1
 335 AABIN(J6) = J7
 420 CONTINUE
 430 CONTINUE
 IF (J6.EQ.0) GO TO 290
 J7 = J6
 340 FAANEH = J6-1
 440 XIRN = MANF(NUUM)
 J6 = XIRN*FAANEH+1.5

ALUYORD 279
 ALUYORD 280
 ALUYORD 281
 ALUYORD 282
 ALUYORD 283
 ALUYORD 284
 ALUYORD 285
 ALUYORD 286
 ALUYORD 287
 ALUYORD 288
 ALUYORD 289
 ALUYORD 290
 ALUYORD 291
 ALUYORD 292
 ALUYORD 293
 ALUYORD 294
 ALUYORD 295
 ALUYORD 296
 ALUYORD 297
 ALUYORD 298
 ALUYORD 299
 ALUYORD 300
 ALUYORD 301
 ALUYORD 302
 ALUYORD 303
 ALUYORD 304
 ALUYORD 305
 ALUYORD 306
 ALUYORD 307
 ALUYORD 308
 ALUYORD 309
 ALUYORD 310
 ALUYORD 311
 ALUYORD 312
 ALUYORD 313
 ALUYORD 314
 ALUYORD 315
 ALUYORD 316
 ALUYORD 317
 ALUYORD 318
 ALUYORD 319
 ALUYORD 320
 ALUYORD 321
 ALUYORD 322
 ALUYORD 323
 ALUYORD 324
 ALUYORD 325
 ALUYORD 326
 ALUYORD 327
 ALUYORD 328
 ALUYORD 329
 ALUYORD 330
 ALUYORD 331
 ALUYORD 332
 ALUYORD 333
 ALUYORD 334
 ALUYORD 335

	IF (J6.LF.0) GU TJ 440	ALUYORD	336
	IF (J6.GT.J7) GU TU 440	ALUYORD	337
345	J7 = AABIN(J6)	ALUYORD	338
	ID = IDENT(J7)	ALUYORD	339
	CALL MTBIN (ID,K,L,M,J8)	ALUYORD	340
	J6 = L8	ALUYORD	341
	L8 = L	ALUYORD	342
350	L = J6	ALUYORD	343
	CALL FILBIN (I,K,L,M,J)	ALUYORD	344
	IDENT(I) = I	ALUYORD	345
	CALL FILBIN (I,K,L,M,J8)	ALUYORD	346
	IDENT(J7) = I	ALUYORD	347
355	MAA = MAA+MABUIN	ALUYORD	348
	MBA = MBA+MABUIN	ALUYORD	349
	GU TO 160	ALUYORD	350
	C	ALUYORD	351
	C--- RANDOM SEARCH HAS FAILED (AFTER 3*N PROBES) TO ACHIEVE CRYSTAL	ALUYORD	352
360	C SPECIFICATIONS. TRY A SEQUENTIAL SEARCH.	ALUYORD	353
	450 WRITE (6,820)	ALUYORD	354
	460 IF (ISAVE.GE.N) GU TO 470	ALUYORD	355
	ISAVE = ISAVE+1	ALUYORD	356
	I = ISAVE	ALUYORD	357
365	GU TO 500	ALUYORD	358
	470 WRITE (6,810)	ALUYORD	359
	C	ALUYORD	360
	480 WRITE (6,880) FA,FANEW,FAA,FAANEW,FB,FBNEW,FBB,FBBNEW,S,SNEW	ALUYORD	361
	C	ALUYORD	362
370	FA = FANEW	ALUYORD	363
	FB = FBNEW	ALUYORD	364
	FAA = FAANEW	ALUYORD	365
	FBB = FBBNEW	ALUYORD	366
	S = SNEW	ALUYORD	367
375	C	ALUYORD	368
	MA = 0	ALUYORD	369
	MB = 0	ALUYORD	370
	C	ALUYORD	371
	C	ALUYORD	372
380	C--- SECOND FUNCTION = - - - NEAREST NEIGHBOR COUNT FOR EACH ATOM	ALUYORD	373
	C	ALUYORD	374
	C	ALUYORD	375
	490 DO 500 I=1,14	ALUYORD	376
	AABIN(I) = 0	ALUYORD	377
385	ABBIN(I) = 0	ALUYORD	378
	BABIN(I) = 0	ALUYORD	379
	500 BBHIN(I) = 0	ALUYORD	380
	DO 550 I=1,N	ALUYORD	381
	ID = IDENT(I)	ALUYORD	382
390	CALL MTBIN (ID,K,L,M,J)	ALUYORD	383
	IF (M.GE.10) GU TJ 540	ALUYORD	384
	A = X(I)	ALUYORD	385
	CALL UNPAC3 (A,XO,YO,ZO)	ALUYORD	386
	NDATOM(1) = 0	ALUYORD	387
395	NDATOM(2) = 0	ALUYORD	388
	DO 520 J1=1,125	ALUYORD	389
	J2 = J+1K(1,J1)	ALUYORD	390
	IF (J2.LE.0) GU TU 520	ALUYORD	391
	IF (J2.GT.NB) GU TU 520	ALUYORD	392

400 J3 = IDUX(J2)
 IF (J3.EQ.0) GO TO 520
 CALL MTBIN (J3, IDJ(1), IDS(2), IDJ(3), IDS(4))
 DO 510 J6=1,4
 J7 = IDS(J6)
 405 IF (J7.LE.0) GO TO 510
 IF (J7.EQ.1) GO TO 510
 ID = IDENT(J7)
 CALL MTBIN (ID, X8, L8, M8, J8)
 IF (L8.LE.0) GO TO 510
 410 IF (L8.GT.2) GO TO 510
 A = X(J7)
 CALL UNPAC3 (A, X1, Y1, Z1)
 DIST = (X0-X1)**2+(Y0-Y1)**2+(Z0-Z1)**2
 IF (DIST.GT.DIST2) GO TO 510
 415 C--- COUNT A-NEIGHBORS (L8=1) AND B-NEIGHBORS (L8=2)
 NUATOM(L9) = NUATOM(L8)+1
 510 CONTINUE
 520 CONTINUE
 IF (L.EQ.2) GO TO 530
 420 MA = MA+1
 J6 = NUATOM(1)+1
 IF (J6.GT.14) J6 = 14
 AABIN(J6) = AABIN(J6)+1
 J6 = NUATOM(2)+1
 425 IF (J6.GT.14) J6 = 14
 ABBIN(J6) = ABBIN(J6)+1
 GO TO 540
 530 MB = MB+1
 J6 = NUATOM(1)+1
 430 IF (J6.GT.14) J6 = 14
 BABIN(J6) = BABIN(J6)+1
 J6 = NUATOM(2)+1
 IF (J6.GT.14) J6 = 14
 BBBIN(J6) = BBBIN(J6)+1
 435 540 CALL FILMIN (I1, I, L, M, J)
 IDENT(I) = I1
 550 CONTINUE
 C
 C--- NOW COMPUTE FRACTIONS AND PRINTOUT
 440 MAA = 0
 MAB = 0
 MBA = 0
 MBB = 0
 DO 560 I=1,14
 445 MAA = MAA+AABIN(I)
 MAB = MAB+ABBIN(I)
 MBA = MBA+BABIN(I)
 560 MBB = MBB+BBBIN(I)
 FMAA = MAA
 450 FMAB = MAB
 FMBA = MBA
 FMBB = MBB
 DO 570 I=1,14
 FAABIN(I) = 0.
 455 IF (MAA.GT.0) FAABIN(I) = FLOA(AABIN(I))/FMAA
 FARBIN(I) = 0.

ALDYORD 393
 ALDYORD 394
 ALDYORD 395
 ALDYORD 396
 ALDYORD 397
 ALDYORD 398
 ALDYORD 399
 ALDYORD 400
 ALDYORD 401
 ALDYORD 402
 ALDYORD 403
 ALDYORD 404
 ALDYORD 405
 ALDYORD 406
 ALDYORD 407
 ALDYORD 408
 ALDYORD 409
 ALDYORD 410
 ALDYORD 411
 ALDYORD 412
 ALDYORD 413
 ALDYORD 414
 ALDYORD 415
 ALDYORD 416
 ALDYORD 417
 ALDYORD 418
 ALDYORD 419
 ALDYORD 420
 ALDYORD 421
 ALDYORD 422
 ALDYORD 423
 ALDYORD 424
 ALDYORD 425
 ALDYORD 426
 ALDYORD 427
 ALDYORD 428
 ALDYORD 429
 ALDYORD 430
 ALDYORD 431
 ALDYORD 432
 ALDYORD 433
 ALDYORD 434
 ALDYORD 435
 ALDYORD 436
 ALDYORD 437
 ALDYORD 438
 ALDYORD 439
 ALDYORD 440
 ALDYORD 441
 ALDYORD 442
 ALDYORD 443
 ALDYORD 444
 ALDYORD 445
 ALDYORD 446
 ALDYORD 447
 ALDYORD 448
 ALDYORD 449

```

      IF (MAB.GT.0) FABBIN(1) = FLOA!(ABBIN(1))/FMAA
      FABBIN(1) = 0.
      IF (MBA.GT.0) FBABIN(1) = FLOA!(BABIN(1))/FMBB
      FBABIN(1) = 0.
      IF (MBB.GT.0) FMBBIN(1) = FLOA!(BBBIN(1))/FMBB
      CONTINUE
      MAPB = MA+MB
      WRITE (6,590) AA,AA,MAPB,AA,MA,AA,FA,AA,AA,FAA,AABIN,FAABIN
      WRITE (6,590) AA,BB,MAPB,AA,MA,AA,FA,AA,AA,FAA,ABBIN,FABBIN
      WRITE (6,590) BB,AA,MAPB,BB,MB,BB,FB,BB,BB,FBB,BABIN,FBABIN
      WRITE (6,590) BB,BB,MAPB,BB,MB,BB,FB,BB,BB,FBB,BBBIN,FBUBIN
      IF (ISAVE.GE.N) STOP 'ALOYORD'
      RETURN
C
C
C
580  FORMAT (1M127MCRYSTAL-LATTICE DESCRIPTION/15X5HINPUT3X6HOUTPUT/
+ 9H FA,.....2F9.3/9H FAA,....2F9.3/9H FB,.....2F9.3/
+ 9H FBH,.....2F9.3/9H S,.....2F9.3)
590  FORMAT (1M015XA1,9H = ATOMS3XA1,12H = NEIGHBORS/26H NO. OF ATOM
+S EXAMINED 115/9H NO. OF ,A1,16H=ATOMS EXAMINED 115/2H FA1,21XF7.
+3/2H F2A1,20XF7.3/1X42H SPATIAL DISTRIBUTION OF NEIGHBORING ATOMS/
+25X1H06X1M16X1M26X1M36X1M46X1M56X1M66X1M76X1M86X1M95X2H105X2H115X2
+M124X3H>12/16H NUMBER OBSERVED3X1417/19H FRACTION OBSERVED 14F7.4/
+//)
600  FORMAT (113H03$$$ CAUTION $$$ SUBROUTINE ALOYORD FINDS THE N
+UMBER OF ATOMS INCONSISTENT WITH ASSUMPTIONS USED THERE-IN,/)
610  FORMAT (100H0SUBROUTINE ALOYORD UNABLE TO GENERATE THE REQUESTED L
+EVEL OF DISORDER IN THE LATTICE. RUN ABORTED.)
620  FORMAT (108H1SUBROUTINE ALOYORD FAILED TO MEET CRYSTAL SPECIFICATI
+ONS USING RANDOM SEARCH. SEQUENTIAL SEARCH ATTEMPTED.)
      END

```

```

ALOYORD 450
ALOYORD 451
ALOYORD 452
ALOYORD 453
ALOYORD 454
ALOYORD 455
ALOYORD 456
ALOYORD 457
ALOYORD 458
ALOYORD 459
ALOYORD 460
ALOYORD 461
ALOYORD 462
ALOYORD 463
ALOYORD 464
ALOYORD 465
ALOYORD 466
ALOYORD 467
ALOYORD 468
ALOYORD 469
ALOYORD 470
ALOYORD 471
ALOYORD 472
ALOYORD 473
ALOYORD 474
ALOYORD 475
ALOYORD 476
ALOYORD 477
ALOYORD 478
ALOYORD 479
ALOYORD 480
ALOYORD 481

```



```

1      SUBROUTINE CRYSTL
C
C      THIS SUBROUTINE GENERATES AN IDEAL COPPER LATTICE WITH X-, Y-, Z-
C      COORDINATE AXES IN THE <1-10>, <11-1>, AND <112>, RESPECTIVELY,
5      AND ALSO GENERATES A UNIQUE IDENTIFICATION NUMBER (I.E., BOUNDARY
C      TYPE FOR EACH ATOM OF THE LATTICE - CONTINUUM INTERFACE WHICH
C      EXPERIENCES IDENTICAL PRESSURE, HOOKIAN, AND VISCOUS DRAG (IF
C      NECESSARY) FORCES (I.E., FOR EACH GROUP OF ATOMS WHICH HAVE THE
C      SAME SURROUNDING ATOM CONFIGURATION).
10     C
C      COMMON NMIN, DT, N, NL, VX(6460), TH(6460), TEMP, DEFECT,
C      + DTN, NS, NLS, NA, UVC, UHC, UF
C      COMMON /ALPHA/ X(6460), ZX(6460), ZY(6460), ZZ(6460),
15     + ENGPI(6460), IDENT(6460), IBUX(14600), EA(1001), EEA(1001),
C      + EB(1001), EEB(1001), EAB(1001), EAB(1001)
C      COMMON /ELMT/ ENGPI, IO, EI, EC, J3, DJV, RMAX2, INIB, N12, ML,
C      + IELE, L, LT, IDT
C      LEVEL 2, X, ZX, ZY, ZZ, ENGPI, IDENT, IBUX, EA, EEA, EB, EEB,
C      + EAB, EAB
20     DIMENSION IM(12)
C
C      DATA ZSI/45.6629/
C      DATA IM/2,4*1,2,2*1,4*3/
25     C
C      DO 20 I=1,N
20     IDENT(I) = 1
C      KN = 164*3
C      IO = 1
C      IO1 = KN+1
30     C
C      GENERATE IDENTIFICATION NUMBERS FOR PROTOTYPE PLANES 1 AND 4
C
C      DO 170 J=1,39
C      GO TO 140,50,30,70,40,50,100,110,90,50,70,40,100,120,90,70,40,100,
35     +130,90,70,30,140,60,30,140,60,30,140,60,80,130,150,80,120,150,80,1
C      +10,150), J
C      IDENT(IO) = 101*IDENT(IO1) = 104
C      GO TO 160
40     IDENT(IO) = 201*IDENT(IO1) = 204
C      GO TO 160
40     IDENT(IO) = 401*IDENT(IO1) = 402
C      IO = IO+1*IO1 = IO1+1
C      IDENT(IO) = 401*IDENT(IO1) = 402
C      GO TO 160
45     IDENT(IO) = 601*IDENT(IO1) = 604
C      GO TO 160
C      IDENT(IO) = 801*IDENT(IO1) = 804
C      GO TO 160
50     IDENT(IO) = 1001*IDENT(IO1) = 1004
C      GO TO 160
C      IDENT(IO) = 1102*IDENT(IO1) = 1105
C      IO = IO+1*IO1 = IO1+1
C      IDENT(IO) = 1101*IDENT(IO1) = 1104
C      GO TO 160
55     IDENT(IO) = 901*IDENT(IO1) = 904
C      IO = IO+1*IO1 = IO1+1
C      IDENT(IO) = 902*IDENT(IO1) = 905

```

5-30

		GO TO 160	CRYSTL	51
60	110	IDENT(ID) = 10>IDENT(ID1) = 10	CRYSTL	52
		ID = ID+3>ID1 = ID1+3	CRYSTL	53
		IDENT(ID) = 10>IDENT(ID1) = 10	CRYSTL	54
		GO TO 160	CRYSTL	55
	120	IDENT(ID) = 10>IDENT(ID1) = 10	CRYSTL	56
		ID = ID+11>ID1 = ID1+11	CRYSTL	57
65		IDENT(ID) = 10>IDENT(ID1) = 10	CRYSTL	58
		GO TO 160	CRYSTL	59
	130	IDENT(ID) = 10>IDENT(ID1) = 10	CRYSTL	60
		ID = ID+13>ID1 = ID1+13	CRYSTL	61
		IDENT(ID) = 10>IDENT(ID1) = 10	CRYSTL	62
70		GO TO 160	CRYSTL	63
	140	IDENT(ID) = 10>IDENT(ID1) = 10	CRYSTL	64
		ID = ID+17>ID1 = ID1+17	CRYSTL	65
		IDENT(ID) = 10>IDENT(ID1) = 10	CRYSTL	66
		GO TO 160	CRYSTL	67
75	150	IDENT(ID) = 120>IDENT(ID1) = 1204	CRYSTL	68
	160	CONTINUE	CRYSTL	69
		ID = ID+1>ID1 = ID1+1	CRYSTL	70
	170	CONTINUE	CRYSTL	71
	C		CRYSTL	72
80	C		CRYSTL	73
	C	GENERATE IDENTIFICATION NUMBERS FOR PHOTOTYPE PLANES 2 AND 5	CRYSTL	74
	C		CRYSTL	75
	C		CRYSTL	76
		DU 290 J=1,30	CRYSTL	77
85		GO TO (190,240,230,190,250,230,190,260,230,180,270,210,180,270,210	CRYSTL	78
		*,180,270,210,180,270,210,200,280,220,200,250,220,200,240,220), J	CRYSTL	79
	180	IDENT(ID) = 10>IDENT(ID1) = 105	CRYSTL	80
		GO TO 280	CRYSTL	81
	190	IDENT(ID) = 20>IDENT(ID1) = 204	CRYSTL	82
90		GO TO 280	CRYSTL	83
	200	IDENT(ID) = 30>IDENT(ID1) = 303	CRYSTL	84
		GO TO 280	CRYSTL	85
	210	IDENT(ID) = 60>IDENT(ID1) = 605	CRYSTL	86
		GO TO 280	CRYSTL	87
95	220	IDENT(ID) = 70>IDENT(ID1) = 703	CRYSTL	88
		GO TO 280	CRYSTL	89
	230	IDENT(ID) = 80>IDENT(ID1) = 804	CRYSTL	90
		GO TO 280	CRYSTL	91
	240	IDENT(ID) = 10>IDENT(ID1) = 10	CRYSTL	92
100		ID = ID+3>ID1 = ID1+3	CRYSTL	93
		IDENT(ID) = 10>IDENT(ID1) = 10	CRYSTL	94
		GO TO 280	CRYSTL	95
	250	IDENT(ID) = 10>IDENT(ID1) = 10	CRYSTL	96
		ID = ID+11>ID1 = ID1+11	CRYSTL	97
105		IDENT(ID) = 10>IDENT(ID1) = 10	CRYSTL	98
		GO TO 280	CRYSTL	99
	260	IDENT(ID) = 10>IDENT(ID1) = 10	CRYSTL	100
		ID = ID+13>ID1 = ID1+13	CRYSTL	101
		IDENT(ID) = 10>IDENT(ID1) = 10	CRYSTL	102
110		GO TO 280	CRYSTL	103
	270	IDENT(ID) = 10>IDENT(ID1) = 10	CRYSTL	104
		ID = ID+17>ID1 = ID1+17	CRYSTL	105
		IDENT(ID) = 10>IDENT(ID1) = 10	CRYSTL	106
	280	CONTINUE	CRYSTL	107

```

115      ID = ID+1>ID1 = ID1+1      CRYSTL 108
      290      CONTINUE      CRYSTL 109
      C      CRYSTL 110
      C      CRYSTL 111
      C      GENERATE IDENTIFICATION NUMBERS FOR PROTOTYPE PLANES 3 AND 6      CRYSTL 112
120      C      CRYSTL 113
      DD 440 J=1,39      CRYSTL 114
      GO TO (350,390,370,350,400,370,350,410,370,300,420,350,300,420,330      CRYSTL 115
      +,300,420,330,310,360,410,380,340,310,360,400,380,340,310,320,340,3      CRYSTL 116
      +90,380,320,340,310,320,320,340), J      CRYSTL 117
125      300      IDENT(ID) = 103>IDENT(ID1) = 106      CRYSTL 118
      GO TO 430      CRYSTL 119
      310      IDENT(ID) = 302>IDENT(ID1) = 304      CRYSTL 120
      GO TO 430      CRYSTL 121
      320      IDENT(ID) = 501>IDENT(ID1) = 502      CRYSTL 122
130      ID = ID+1>ID1 = ID1+1      CRYSTL 123
      IDENT(ID) = 501>IDENT(ID1) = 502      CRYSTL 124
      GO TO 430      CRYSTL 125
      330      IDENT(ID) = 603>IDENT(ID1) = 606      CRYSTL 126
      GO TO 430      CRYSTL 127
135      340      IDENT(ID) = 702>IDENT(ID1) = 704      CRYSTL 128
      GO TO 430      CRYSTL 129
      350      IDENT(ID) = 903>IDENT(ID1) = 906      CRYSTL 130
      GO TO 430      CRYSTL 131
      360      IDENT(ID) = 1002>IDENT(ID1) = 1005      CRYSTL 132
140      ID = ID+1>ID1 = ID1+1      CRYSTL 133
      IDENT(ID) = 1003>IDENT(ID1) = 1006      CRYSTL 134
      GO TO 430      CRYSTL 135
      370      IDENT(ID) = 1103>IDENT(ID1) = 1106      CRYSTL 136
      GO TO 430      CRYSTL 137
145      380      IDENT(ID) = 1203>IDENT(ID1) = 1206      CRYSTL 138
      ID = ID+1>ID1 = ID1+1      CRYSTL 139
      IDENT(ID) = 1202>IDENT(ID1) = 1205      CRYSTL 140
      GO TO 430      CRYSTL 141
      390      IDENT(ID) = 10>IDENT(ID1) = 10      CRYSTL 142
150      ID = ID+3>ID1 = ID1+3      CRYSTL 143
      IDENT(ID) = 105>IDENT(ID1) = 10      CRYSTL 144
      GO TO 430      CRYSTL 145
      400      IDENT(ID) = 10>IDENT(ID1) = 10      CRYSTL 146
      ID = ID+1>ID1 = ID1+1      CRYSTL 147
155      IDENT(ID) = 10>IDENT(ID1) = 10      CRYSTL 148
      GO TO 430      CRYSTL 149
      410      IDENT(ID) = 10>IDENT(ID1) = 10      CRYSTL 150
      ID = ID+1>ID1 = ID1+1      CRYSTL 151
      IDENT(ID) = 10>IDENT(ID1) = 10      CRYSTL 152
160      GO TO 430      CRYSTL 153
      420      IDENT(ID) = 10>IDENT(ID1) = 10      CRYSTL 154
      ID = ID+1>ID1 = ID1+1      CRYSTL 155
      IDENT(ID) = 10>IDENT(ID1) = 10      CRYSTL 156
165      430      CONTINUE      CRYSTL 157
      ID = ID+1>ID1 = ID1+1      CRYSTL 158
      440      CONTINUE      CRYSTL 159
      C      CRYSTL 160
      C      SET IDENTIFICATION NUMBERS OF ATOMS IN REMAINING PLANES EQUAL TO      CRYSTL 161
      C      IDENTIFICATION NUMBERS OF SYMMETRICALLY EQUIVALENT ATOMS IN THE      CRYSTL 162
170      C      FIRST 6 PLANES      CRYSTL 163
      C      CRYSTL 164

```

5-32

```

175      NSLB = N/164
      NMSB = NSLB/6-1
      NK = 164*6
      ID = ID1
      IF (NMSB.EQ.1) GO TO 460
      DO 450 KK=2,NMSB
      DO 450 J=1,NK
      IDENT(ID) = IDENT(J)
      ID = ID+1
180      CONTINUE
      C
      C      SET IDENTIFICATION NUMBERS OF ATOMS IN THE 6 BOUNDARY PLANES
      C      NOTE = THE BOUNDARY PLANE EQUIVALENTS ARE 1,2,3,4 AND 3,4,5,6
185      460      IDN = 0
      NK = 164*4
      K = 1
      DO 480 KK=1,2
      DO 470 J=1,NK
190      IF (MOD(J,164).EQ.0) K = K+1
      JJ = J+IDN
      IDENT(ID) = IDENT(JJ)+1200
      IF (IDENT(JJ).EQ.1.OR.IDENT(JJ).EQ.10) IDENT(ID) = K*10+1
      IF (KK.EQ.1.OR.IDENT(JJ).LT.100) GO TO 470
195      M2 = IDENT(ID)/100-12
      IDENT(ID) = IDENT(ID)+IM(M2)
      470      ID = ID+1
      IDN = 328
      480      CONTINUE
200      C
      C      CALCULATE X-, Y-, Z-COORDINATES FOR ALL ATOMS IN THE CRYSTAL LATTI
      C
      XST = 34.6
      YST = 33.1
205      DELX = 1./((SQRT(2.))*6)
      DELY = SQRT(3.)/6
      DELZ = SQRT(3./2.)/3.6
      IY = 10
      K = 5
210      ID = 1
      DO 580 KK=1,NSLB
      Z11 = ZST+FLOAT(K-1)*DELZ
      DO 540 J=1,IY
      GO TO (490,500,510,520,520,520,520,510,500,490), J
215      490      XREF = XST+8.*DELX*IX = 6
      GO TO 530
      500      XREF = XST+4.*DELX*IX = 14
      GO TO 530
      510      XREF = XST+2.*DELX*IX = 18
      GO TO 530
220      520      XREF = XST*IX = 24
      530      DO 540 I=1,IX
      X11 = XREF+FLOAT(I-1)*DELX
      Y11 = YST+FLOAT(J-1)*DELY
225      CALL PAC3 (A,X11,Y11,Z11)
      X(ID) = A
      ID = ID+1
      540      CONTINUE

```

```

CRYSTL 165
CRYSTL 166
CRYSTL 167
CRYSTL 168
CRYSTL 169
CRYSTL 170
CRYSTL 171
CRYSTL 172
CRYSTL 173
CRYSTL 174
CRYSTL 175
CRYSTL 176
CRYSTL 177
CRYSTL 178
CRYSTL 179
CRYSTL 180
CRYSTL 181
CRYSTL 182
CRYSTL 183
CRYSTL 184
CRYSTL 185
CRYSTL 186
CRYSTL 187
CRYSTL 188
CRYSTL 189
CRYSTL 190
CRYSTL 191
CRYSTL 192
CRYSTL 193
CRYSTL 194
CRYSTL 195
CRYSTL 196
CRYSTL 197
CRYSTL 198
CRYSTL 199
CRYSTL 200
CRYSTL 201
CRYSTL 202
CRYSTL 203
CRYSTL 204
CRYSTL 205
CRYSTL 206
CRYSTL 207
CRYSTL 208
CRYSTL 209
CRYSTL 210
CRYSTL 211
CRYSTL 212
CRYSTL 213
CRYSTL 214
CRYSTL 215
CRYSTL 216
CRYSTL 217
CRYSTL 218
CRYSTL 219
CRYSTL 220
CRYSTL 221

```

```

230      IF (MOD(KK,2).NE.0) XST = XST+DELX/2.
      IF (MOD(KK,2).EQ.0) XST = XST-DELX/2.
      N1 = MOD(KK,6)
      IF (KK.GT.(NSLB-4)) N1 = N1-2
      IF (N1.LE.0) N1 = N1+6
      GO TO (550,550,560,550,550,560), N1
235      550  YST = YST+DELY/3.
      GO TO 570
      560  YST = YST-DELY/3.*2.
      570  K = K+1
240      IF (KK.NE.(NSLB-4)) GO TO 500
      K = 1
      YST = YST+DELY/3.
      580  CONTINUE
      RETURN
      END

```

```

CRYSTL 222
CRYSTL 223
CRYSTL 224
CRYSTL 225
CRYSTL 226
CRYSTL 227
CRYSTL 228
CRYSTL 229
CRYSTL 230
CRYSTL 231
CRYSTL 232
CRYSTL 233
CRYSTL 234
CRYSTL 235
CRYSTL 236
CRYSTL 237

```

```

1      SUBROUTINE DAMP (M2,M1,ENBH,DX1,DY1,DZ1,FM,CCX,CCY,CCZ,IAFIX)      DAMP      2
      C      THIS SUBROUTINE CALCULATES THE RESTORING FORCES AND VISCOUS      DAMP      3
      C      DAMPING FORCES ACTING ON EACH ATOM, EACH TIME STEP.          DAMP      4
5      C      COMMON /FORCE/ AX(25,9),AY(25,9),AZ(25,9),SFH              DAMP      5
      C      COMMON /DMP/ CXN,CYN,CZN,CXD,CYD,CZD,FX,DX,DY,DZ,NWDL,NWDH,NFRZ DAMP      6
      C      LOGICAL IAFIX,NFRZ                                           DAMP      7
      C      CALCULATE DAMPING FACTOR                                     DAMP      8
10     C      IF (M.EQ.91) GO TO 40                                       DAMP      9
      C      IF (M.LT.10.OR.NFRZ) GO TO 30                                DAMP     10
      C      IF ((M.GE.10.AND.N.LT.NWDL).OR.(M.GT.NWDH.AND.M.LT.90).OR.M.FG.97) DAMP     11
      C      + GO TO 20                                                    DAMP     12
      C      USE FULL DAMPING                                              DAMP     13
15     C      CXN = 1.-.5*CCX*FM                                          DAMP     14
      C      CYN = 1.-.5*CCY*FM                                          DAMP     15
      C      CZN = 1.-.5*CCZ*FM                                          DAMP     16
      C      CXD = 1.-.5*CCX*FM                                          DAMP     17
      C      CYD = 1.-.5*CCY*FM                                          DAMP     18
      C      CZD = 1.-.5*CCZ*FM                                          DAMP     19
20     C      GO TO 50                                                    DAMP     20
      C      USE WEAK DAMPING                                              DAMP     21
25     C      CXN = 1.-.1*CCX*FM                                          DAMP     22
      C      CYN = 1.-.1*CCY*FM                                          DAMP     23
      C      CZN = 1.-.1*CCZ*FM                                          DAMP     24
      C      CXD = 1.-.1*CCX*FM                                          DAMP     25
      C      CYD = 1.-.1*CCY*FM                                          DAMP     26
      C      CZD = 1.-.1*CCZ*FM                                          DAMP     27
      C      GO TO 50                                                    DAMP     28
30     C      USE NO DAMPING                                              DAMP     29
35     C      CXN = CXD = 1.0                                             DAMP     30
      C      CYN = CYD = 1.0                                             DAMP     31
      C      CZN = CZD = 1.0                                             DAMP     32
      C      GO TO 50                                                    DAMP     33
40     C      ZERO KINETIC ENERGY DAMPING                               DAMP     34
      C      CXN = .5*CXD = 1.5                                          DAMP     35
      C      CYN = .5*CYD = 1.5                                          DAMP     36
      C      CZN = .5*CZD = 1.5                                          DAMP     37
      C      CALCULATE INTERFACE FORCES                                    DAMP     38
45     C      IF (M.LT.10.OR.(M.GT.90.AND.M.LT.100)) GO TO 60           DAMP     39
      C      IF (M.EQ.10) GO TO 70                                         DAMP     40
      C      USE FORCES CALCULATED BY SPEFMS                               DAMP     41
      C      IF (IAFIX) GO TO 60                                           DAMP     42
      C      A = AX(M2,M1) $ CALL UNPACF (A,A1,B1)                       DAMP     43
      C      A = AY(M2,M1) $ CALL UNPACF (A,A2,B2)                       DAMP     44
      C      A = AZ(M2,M1) $ CALL UNPACF (A,A3,B3)                       DAMP     45
      C      GO TO 80                                                      DAMP     46
      C      USE NO INTERFACE FORCES                                       DAMP     47
50     C      A1 = A2 = A3 = 0.0                                           DAMP     48
      C      B1 = B2 = B3 = 0.0                                           DAMP     49
      C      GO TO 80                                                      DAMP     50
      C      USE SPECIFIED WEAK INTERFACE FORCES                           DAMP     51
55     C      A1 = A2 = A3 = 0.0                                           DAMP     52
      C      B1 = B2 = B3 = SFH*FM                                         DAMP     53
      C      DX1 = A1-B1*FM*DX                                             DAMP     54
      C      DY1 = A2-B2*FM*DY                                             DAMP     55
      C      DZ1 = A3-B3*FM*DZ                                             DAMP     56
      C      DZ1 = A3-B3*FM*DZ                                             DAMP     57
      C      DZ1 = A3-B3*FM*DZ                                             DAMP     58

```

SUBROUTINE DAMP

76/16

UP1=0 TRACE

FTN 4.5+414

11/13/79 14.17.08

PAGE

2

ENGH = 0.5*(B1*UX**2+B2*DY**2+B3*UZ**2)
RETURN
END

DAMP 59
DAMP 60
DAMP 61

60

5-36

```

1      SUBROUTINE DASPKA (POL,AZI,XG,YG,ZG,IP,KEPKA,UT,DX11,DY11,DZ11,LPR DASPKA      2
      +NT1) DASPKA      3
      C DASPKA      4
      C THIS SUBROUTINE INITIATES A PKA (IDENTIFICATION NUMBER IP) WITH DASPKA      5
      C ENERGY KEPKA, IN A DIRECTION DEFINED BY EITHER THE POLAR AND DASPKA      6
      C AZIMUTHAL ANGLES POL AND AZI, RESPECTIVELY, OR THE NORMALIZED DASPKA      7
      C DIRECTIONAL INDICES XG, YG, ZG. ALSO, THE SUBROUTINE WRITES DASPKA      8
      C A - THE NORMALIZED DIRECTIONAL INDICES AND DIRECTIONAL COSINES DASPKA      9
      C RELATIVE TO (1) A COORDINATE SYSTEM WITH AXES DIRECTED DASPKA     10
      C ALONG <100>, <010>, AND <001> (XG,YG,ZG,ALPHAG,BETAG,GAMMAG DASPKA     11
      C RESPECTIVELY) AND (2) A COORDINATE SYSTEM WITH AXES DASPKA     12
      C DIRECTED ALONG THE <110>, <111>, AND <112> (XC,YC,ZC, DASPKA     13
      C ALPHAC,BETAC,GAMMAC, RESPECTIVELY). DASPKA     14
      C B - THE TIME STEP LENGTH DT, THE ENERGY KEPKA, THE ROOT MEAN DASPKA     15
      C SQUARED DISPLACEMENT REQUIRED TO GIVE THE ENERGY DXYZ (AND DASPKA     16
      C THE COMPONENTS OF THIS DISPLACEMENT, DX11, DY11, DZ11), AND DASPKA     17
      C THE POLAR AND AZIMUTHAL ANGLES POL AND AZI (RELATIVE TO DASPKA     18
      C <001> AND <100>, RESPECTIVELY). DASPKA     19
      C DASPKA     20
      C DASPKA     21
      C DASPKA     22
      C DASPKA     23
      C DASPKA     24
      C DASPKA     25
      C LC = LC+1 DASPKA     26
      C MARK = 0 DASPKA     27
      C DTNC = 57.2957795131 DASPKA     28
      C RG = SQRT(XG**2+YG**2+ZG**2) DASPKA     29
      C IF (RG.GT.0.0) GO TO 90 DASPKA     30
      C MARK = 1 DASPKA     31
      C IF (POL.NE.0.0.OR.POL.NE.180.) GO TO 20 DASPKA     32
      C XG = 0.0 DASPKA     33
      C YG = 0.0 DASPKA     34
      C IF (POL.EQ.0.0) ZG = 1. DASPKA     35
      C IF (POL.EQ.180.) ZG = -1. DASPKA     36
      C GO TO 80 DASPKA     37
      C IF (AZI.NE.90..OR.AZI.NE.270.) GO TO 50 DASPKA     38
      C XG = 0.0 DASPKA     39
      C IF (AZI.GT.180.) GO TO 30 DASPKA     40
      C YG = 1.0 DASPKA     41
      C GO TO 40 DASPKA     42
      C YG = -1.0 DASPKA     43
      C ZG = YG*COS(POL/DTNC)/(SIN(AZI/DTNC)*SQRT(1.-(COS(POL/DTNC))**2)) DASPKA     44
      C GO TO 80 DASPKA     45
      C IF (AZI.GT.90..AND.AZI.LT.270.) GO TO 60 DASPKA     46
      C XG = 1.0 DASPKA     47
      C GO TO 70 DASPKA     48
      C XG = -1.0 DASPKA     49
      C YG = XG*TAN(AZI/DTNC) DASPKA     50
      C ZG = XG*COS(POL/DTNC)/(COS(AZI/DTNC)*SQRT(1.-(COS(POL/DTNC))**2)) DASPKA     51
      C XG1 = ABS(XG) DASPKA     52
      C YG1 = ABS(YG) DASPKA     53
      C ZG1 = ABS(ZG) DASPKA     54
      C GCM = AMAX1(XG1,YG1,ZG1) DASPKA     55
      C XG = XG/GCM DASPKA     56
      C YG = YG/GCM DASPKA     57
      C ZG = ZG/GCM DASPKA     58
      C RG = SQRT(XG**2+YG**2+ZG**2) DASPKA     59

```


5-37

```

90      XC = (.70/107*XB+.707107*YG)/.6
      YC = (.577351*XB+.577351*YG+.577351*ZG)/.6
60      ZC = (.408239*XB+.408239*YG+.616477*ZG)/.6
      ALPHAG = ACOS(XG/XC)
      BETAG = ACOS(YG/YC)
      GAMMAG = ACOS(ZG/XC)
      DXYZ = SQRT(2.*KEPKA*UT**2)
      RC = SQRT(XC**2+YC**2+ZC**2)
      ALPHAC = ACOS(XC/RC)
      BETAC = ACOS(YC/RC)
      GAMMAC = ACOS(ZC/RC)
      DX11 = DXYZ*XC/RC
      DY11 = DXYZ*YC/RC
70      DZ11 = DXYZ*ZC/RC
      ALPHAG = ALPHAG*DTNC*ALPHAC = ALPHAC*DTNC
      BETAG = BETAG*DTNC*BETAC = BETAC*DTNC
      GAMMAG = GAMMAG*DTNC*GAMMAC = GAMMAC*DTNC
75      IF (MARK.GT.0) GO TO 130
      XYG = SQRT(XG**2+YG**2)
      IF (XYG.EQ.0.) GO TO 120
      IF (YG.LT.0.0) GO TO 100
      AAG = DTNC*ACOS(XG/XYG)
80      GO TO 110
      AAG = 360.-DTNC*ACOS(XG/XYG)
100     PAG = ACOS(ZG/XC)*DTNC
110     GO TO 130
      AAG = 0.0
      PAG = 0.0
85     IF (ZG.LT.0.) PUL = 180.
130     IF (.NOT.LPHNT1) GO TO 140
      WRITE (6,150) LC,IP,KEPKA,PUL,AZI,DT,DXYZ,DX11,DY11,DZ11
      WRITE (6,160) XG,YG,ZG,ALPHAG,BETAG,GAMMAG,XC,YC,ZC,ALPHAC,BETAC,G
90     +AMMAC
140     RETURN
      C
      C
95     C
150     FORMAT ('OPKA NUMBER',I5,' (IU=',I6,')', HAS A KINETIC ENERGY OF',
+FB.2,' WITH PULAR AND AZIMUTHAL ANGLES OF',FB.2,' AND',FB.2,' RESP
+ECTIVELY.'/' HENCE, FOR A TIME STEP OF',1PE11.4,' THE ROOT MEAN SQ
+UARED DISPLACEMENT REQUIRED TO INITIATE THIS PKA IS',1PE11.4,' WIT
+H X=, Y=, Z=COMPONENTS,',OPFB.4,' ',FB.4,' ',FB.4,' , RESPECTIVELY.
+ '/')
100    C
160     FORMAT ('THE DIRECTIONAL INDICES AND COSINES RELATIVE TO THE UNIT
+CELL=EDGE COORDINATE SYSTEM ARE'/' XG=',FB.2,' , YG=',FB.2,' , ZG=
+FB.2,' , ALPHAG=',FB.2,' , BETAG=',FB.2,' , AND GAMMAG=',FB.2,' , RES
+PECTIVELY.'/' WHILE THOSE FOR THE CUMENT COORDINATE SYSTEM ARE'/'
+XC=',FB.2,' , YC=',FB.2,' , ZC=',FB.2,' , ALPHAC=',FB.2,' , BETAC=',FB
+.2,' , AND GAMMAC=',FB.2,' , RESPECTIVELY.'/)
105    C
      END
      DASPKA
54
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108

```

5-38

```

1      SUBROUTINE DFCT (ID,DXD,DYD,DZD,LPD,DXE,DYE,DZE,LPE,IDV,N12S,IPS) DFCT      2
      C      THIS ROUTINE INSERTS EITHER A VACANCY OR AN INTERSTITIAL INTO THE DFCT      3
      C      CRYSTAL. DFCT      4
      C      . DFCT      5
5      C      COMMON RHIN, UT, N, NL, VX(6460), TB(6460), TEMP, DEFECT, DFCT      6
      C      + DTN, NS, NLS, NA, UVC, UHC, UF CM1      2
      C      COMMON /ALPHA/ X(6460), ZZX(6460), ZZY(6460), ZZZ(6460), CM1      3
      C      + ENGP1(6460), IDENT(6460), IBUA(14600), EA(1001), EEA(1001), CM1      4
      C      + EB(1001), EEB(1001), EAB(1001), EEAH(1001) CM1      5
10     C      COMMON /ELNT/ ENGPU, ID, E1, E2, J3, DIV, RMAX2, INIB, N12, HL, CM1      6
      C      + IELE, L, LT, IDT CM1      7
      C      LEVEL 2, X, ZZX, ZZY, ZZZ, ENGP1, IDENT, IHOX, EA, EEA, EB, EEB, CM1      8
      C      + EAB, EEAH CM1      9
15     C      COMMON /HBOX/ MEA, MEA1, NID, ENL, ENLI, XIPS, YIPS, ZIPS, SL, IP, HR, CM1      10
      C      + RESET, CONT, LFINST, XEA, YEA, ZEA, ADAMP, IC, IAFIX, FTAPE DFCT      5
      C      LOGICAL ADAMP, IAFIX, FTAPE, LFINST, CONT, DEFECT DFCT      6
      C      DFCT      7
      C      XTRAN(A,B,C) = .424264*A+.346411*B+.244943*C DFCT      8
20     C      YTRAN(A,B,C) = -.424264*A+.346411*B+.244943*C DFCT      9
      C      ZTRAN(A,B) = -.346411*A+.489888*B DFCT      10
      C      TRANX(A,B) = 1.17851*A - 1.17851*B DFCT      11
      C      TRANY(A,B,C) = .962253*A + .962253*B = .962253*C DFCT      12
      C      TRANZ(A,B,C) = .680397*A + .680397*B + 1.36079*C DFCT      13
25     C      IF (IDV.NE.0) GO TO 30 DFCT      14
      C      IF (IQ.EQ.0) RETURN DFCT      15
      C      INSERT AN INTERSTITIAL IN THE CRYSTAL DFCT      16
30     C      DFCT      17
      C      X1 = 0.5*V1 = 0.5*ZV1 = 0. DFCT      18
      C      NU = NSN = N+15*NL = NL+15*N12 = N12+1 DFCT      19
      C      NT=N $ NS=N $ NLS=NL $ N12S=N12 $ IPN=IQ+1 DFCT      20
      C      DX = TRANX(DXD,DYD) DFCT      21
35     C      DY = TRANY(DXD,DYD,DZD) DFCT      22
      C      DZ = TRANZ(DXD,DYD,DZD) DFCT      23
      C      A = X(IQ) DFCT      24
      C      CALL UNPAC3 (A,X1D,Y1D,Z1D) DFCT      25
      C      X11 = DX+X1Q DFCT      26
40     C      Y11 = DY+Y1Q DFCT      27
      C      Z11 = DZ+Z1Q DFCT      28
      C      CALL PAC3 (A,X11,Y11,Z11) DFCT      29
      C      X(IQ) = A+VX(IQ) = A DFCT      30
      C      I1 = IDENT(IQ) DFCT      31
45     C      CALL HTBIN (I1,K,L,M,J) DFCT      32
      C      L = LPD DFCT      33
      C      CALL FILBIN (I1,K,L,M,J) DFCT      34
      C      IDENT(IQ) = I1 DFCT      35
      C      DX = TRANX(DXE,DYE) DFCT      36
50     C      DY = TRANY(DXE,DYE,DZE) DFCT      37
      C      DZ = TRANZ(DXE,DYE,DZE) DFCT      38
      C      X11 = DX+X1Q DFCT      39
      C      Y11 = DY+Y1Q DFCT      40
      C      Z11 = DZ+Z1Q DFCT      41
55     C      CALL PAC3 (A,X11,Y11,Z11) DFCT      42
      C      DU 20 L1=IPN,NS DFCT      43
      C      X(NT) = X(NU) DFCT      44
      C      DFCT      45
      C      DFCT      46
      C      DFCT      47
      C      DFCT      48
      C      DFCT      49
      C      DFCT      50

```

5-39

		VX(NT) = VX(NU)	DFCT	51
		TB(NT) = TB(NU)	DFCT	52
60		II = IDENT(NU)	DFCT	53
		CALL MTBIN (II,K,L,M,J)	DFCT	54
		K = K+1	DFCT	55
		CALL FILBIN (II,K,L,M,J)	DFCT	56
		IDENT(NT) = II	DFCT	57
65		NT = NT+1	DFCT	58
		NU = NU+1	DFCT	59
	20	CONTINUE	DFCT	60
		X(IPN) = A	DFCT	61
		VX(IPN) = A	DFCT	62
70		TB(IPN) = TB(IU)	DFCT	63
		J = OAK = IPNOL = LPEOM = 1	DFCT	64
		CALL FILBIN (II,K,L,M,J)	DFCT	65
		IDENT(IPN) = II	DFCT	66
		A = TB(IU)	DFCT	67
75		CALL UNPAC3 (A,X1,Y1,Z1)	DFCT	68
		X2 = X1-X1PS	DFCT	69
		Y2 = Y1-Y1PS	DFCT	70
		Z2 = Z1-Z1PS	DFCT	71
		X11 = XTRAN(X2,Y2,Z2)	DFCT	72
80		Y11 = YTRAN(X2,Y2,Z2)	DFCT	73
		Z11 = ZTRAN(Y2,Z2)	DFCT	74
		LI = LI+1	DFCT	75
		XD = X11+OXDSYD = Y11+OYDSZD = Z11+OZD	DFCT	76
		XE = X11+OXESYE = Y11+OYESZE = Z11+OZE	DFCT	77
85		WRITE (6,60) LI,IU,X11,Y11,Z11,LPU,XD,YD,ZD,LPE,XE,YE,ZE	DFCT	78
		GO TO 40	DFCT	79
	C		DFCT	80
	C	INSERT A VACANCY IN THE CRYSTAL	DFCT	81
	30	X11 = 32.	DFCT	82
90	C		DFCT	83
		Y11 = 32.	DFCT	84
		Z11 = 32.	DFCT	85
		CALL PAC3 (A,X11,Y11,Z11)	DFCT	86
		X(IDV) = A	DFCT	87
95		VX(IDV) = A	DFCT	88
		A = TB(IDV)	DFCT	89
		CALL UNPAC3 (A,X1,Y1,Z1)	DFCT	90
		X2 = X1-X1PS	DFCT	91
		Y2 = Y1-Y1PS	DFCT	92
100		Z2 = Z1-Z1PS	DFCT	93
		XV1 = XTRAN(X2,Y2,Z2)	DFCT	94
		YV1 = YTRAN(X2,Y2,Z2)	DFCT	95
		ZV1 = ZTRAN(Y2,Z2)	DFCT	96
		LV = LV+1	DFCT	97
105		WRITE (6,90) LV,IUV,XV1,YV1,ZV1	DFCT	98
	40	IF (FLAPE) WRITE (4) TEMP,IPB,OXD,OYD,OZD,NT,OXE,OYE,OZE,IDV,XV1,Y	DFCT	99
		+V1,ZV1	DFCT	100
		RETURN	DFCT	101
	C		DFCT	102
110	C		DFCT	103
	50	FORMAT ('VACANCY NUMBER',I5,' IS LOCATED AT',I5,'(X=',F5.2,' Y='	DFCT	104
		*,F5.2,', AND Z=',F5.2,')')	DFCT	105
60		FORMAT ('INTERSTITIAL NUMBER',I5,' IS LOCATED ABOUT',I5,'(X=',F5.	DFCT	106
		+2,', Y=',F5.2,', AND Z=',F5.2,')', ' WITH ONE ATOM (LD=',I5,') AT X	DFCT	107

SUBROUTINE DFCT

76/76

OPT=0 TRACE

FTN 4.5+414

11/15/79 14.17.08

PAGE

5

115

+D='F5.2,' , YD='F5.2,' ,1/' AND ZD='F5.2,' AND ANOTHER ATOM (LD=' DFCT 108
+,15,') AT XE='F5.2,' , YE='F5.2,' , AND ZE='F5.2) DFCT 109
END DFCT 110

```

1      SUBROUTINE DIRECT (DBA,DBY,DBZ), RETURNS(A1)
      C
      C      THIS SUBROUTINE CALCULATES PARAMETERS WHICH CHARACTERIZE ATOMS
      C      SIGNIFICANTLY DISPLACED FROM THEIR LATTICE SITES.
5      C
      COMMON NMIN, DT, N, NL, VX(6460), TH(6460), TEMP, DEFECT,
      * DYN, NS, NLS, NA, UVC, UHC, UF
      COMMON /ALPHA/ X(6460), ZZX(6460), ZZY(6460), ZZZ(6460),
      * ENGP1(6460), IDENT(6460), IBUX(14600), EA(1001), EEA(1001),
10     * EB(1001), EEB(1001), EAB(1001), EEAH(1001)
      COMMON /ELMT/ ENGPU, IO, EI, EC, JS, DIV, RMAX2, INIB, N12, ML,
      * IFLE, L, LT, IOT
      LEVEL 2, A, ZZX, ZZY, ZZZ, ENGP1, IDENT, INUX, EA, EEA, EU, FER,
      * EAB, EEAH
15     COMMON /CENTR/ X11, Y11, Z11, AX1, YY1, ZZ1, VX11, VY11, VZ11,
      * TBX1, TRY1, TBZ1, DX1, DY1, DZ1, AZ1, POL
      COMMON /IBUX/ MEA, MEA1, NID, ERL, ERLI, XIPS, YIPS, ZIPS, SL, IP, NR,
      * RESET, COUNT, LFRINGT, XEA, YEA, ZEA, ADAMP, IC, IAFIX, FTAPE
      COMMON /SMTC/ TEMSET, DEFINT, DEFCLS, LINDF, PTDEF, ALLOY, RSTAPF,
20     * LPRNT
      COMMON /DMP/ CXV, CYN, CZN, CXD, CTD, CZD, FHD, DX, DY, DZ, NHUL, NHDH, NFRZ
      COMMON /CNST/ CL, CM, CN, CU, CP, COU, CO
      DIMENSION RPOL(15), NAZI(50), DIN(21)
      LOGICAL DEFECT, LPRNT
25     DATA RPOL /0.,10.435,19.471,20.565,35.204,36.694,45.,54.736,
      * 57.689,63.435,65.405,71.565,74.499,76.367,90./
      DATA NAZI /30.,90.,45.,135.,0.,90.,45.,135.,26.565,63.435,
      * 110.565,153.435,0.,90.,45.,135.,18.435,71.565,108.435,161.565,0.,
      * 290.,26.505,63.435,110.565,153.435,0.,90.,33.69,50.31,123.64,
30     * 146.31,14.036,75.464,104.036,165.964,45.,135.,0.,90.,18.435,
      * 71.565,108.435,161.565,26.565,63.435,116.565,153.435/
      DATA DIN /5H0 0 1,5H1 0 3,5H1 1 4,5H1 0 2,5H1 1 2,5H2 1 3,5H1 1 0,
      * 1 5H1 1 1,5H3 1 2,5H2 0 1,5H2 1 1,5H3 0 1,5H3 2 1,5H4 1 1,5H1 1 0,
      * 2 5H1 0 0,5H0 1 0,5H3 1 0,5H1 3 0,5H2 1 0,5H1 2 0/
35     XTRAN(A,B,C) = .424264*A+.346411*B+.244943*C
      YTRAN(A,B,C) = -.424264*A+.346411*B+.244943*C
      ZTRAN(A,B) = -.346411*A+.484800*B
      ROIST(A,B,C) = (A**2+B**2+C**2)**0.5
      C
40     RCST = 57.2957795131
      POL = 0.0
      AZI = 0.0
      PI=3.1415927
      C
      CONVERT RPOL AND NAZI ARRAYS TO RADIANS (ONLY ONCE)
45     IF(LOUP.EQ.1) GO TO 7
      DO 1 I=1,15
      1 RPOL(I)=RPOL(I)/RCST
      DO 5 I=1,50
      5 NAZI(I)=NAZI(I)/RCST
50     7 LOUP=1
      C
      DAX = X11-XIPS
      DAY = Y11-YIP3
      DAZ = Z11-ZIPS
85     DX1 = XTRAN(DAX,DAY,DAZ)
      DY1 = YTRAN(DAX,DAY,DAZ)
      DZ1 = ZTRAN(DAY,DAZ)

```

5-42

	IF (DEFECT) GO TO 10	DIRECT	51
	DBX = XTRAN(DX,UY,UZ)	DIRECT	52
60	DBY = YTRAN(DX,UY,UZ)	DIRECT	53
	DBZ = ZTRAN(DY,JZ)	DIRECT	54
	DBR = MDIST(DBX,DBY,DBZ)	DIRECT	55
	DBRO = MDIST(DBX,DBY,U,0)	DIRECT	56
		DIRECT	57
65	C DIRECTION OF MOTION OF DISPLACED ATOM AT TIME T	DIRECT	58
	C *****	DIRECT	59
	C	DIRECT	60
	IF (DBR.EQ.0.0) RETURN A1	DIRECT	61
	POL = RCST*ACOS(DBZ/DBR)	DIRECT	62
70	IF (DBRO.LE.0.0) GO TO 20	DIRECT	63
	IF (DBY.LT.0.0) AZI = 360.-RCST*ACOS(DBX/DBRO)	DIRECT	64
	IF (DBY.GE.0.0) AZI = RCST*ACOS(DBX/DBRO)	DIRECT	65
	GO TO 20	DIRECT	66
		DIRECT	67
75	C DIRECTION OF DISPLACED ATOM RELATIVE TO ITS INITIAL LATTICE SITE	DIRECT	68
	C *****	DIRECT	69
	C	DIRECT	70
	10 DAR = MDIST(DX1,DY1,DZ1)	DIRECT	71
	DARW = MDIST(DAX,DAY,DAZ)	DIRECT	72
80	IF (DARW.GT.CG.DR,DAN.EQ.0.0) RETURN A1	DIRECT	73
	IF (DARW.LE.0.0) GO TO 20	DIRECT	74
	POL = RCST*ACOS(DZ1/DAR)	DIRECT	75
	DARO = MDIST(DX1,DY1,U,0)	DIRECT	76
	AZI = 0.0	DIRECT	77
85	IF (DARO.LE.0.0) GO TO 20	DIRECT	78
	IF (DY1.LT.0.0.AND.DARO.NE.0.0) AZI = 360.-RCST*ACOS(DX1/DARO)	DIRECT	79
	IF (DY1.GE.0.0.AND.DARO.NE.0.0) AZI = RCST*ACOS(DX1/DARO)	DIRECT	80
		DIRECT	81
	C	DIRECT	82
90	20 IF (.NOT.LPRNT) RETURN	DIRECT	83
	AMAD = 45.0	DIRECT	84
	I5 = 0	DIRECT	85
	NM1 = 16 \$ NM2 = 23 \$ NM3 = 0	DIRECT	86
	IF (POL.GE.135.0) GO TO 100	DIRECT	87
95	NM1 = 22 \$ NM2 = 30 \$ NM3 = 14	DIRECT	88
	IF (POL.GE.90.0) GO TO 100	DIRECT	89
	NM1 = 7 \$ NM2 = 15 \$ NM3 = 14	DIRECT	90
	IF (POL.GT.45.0) GO TO 100	DIRECT	91
	NM1 = 1 \$ NM2 = 8 \$ NM3 = 0	DIRECT	92
100	100 I4=NM3	DIRECT	93
	DO 125 I1=NM1,NM2	DIRECT	94
	I2=I1	DIRECT	95
	IF (I1.GT.15) I2=I1-15	DIRECT	96
	IF (RPOL(I2).NE.0.0) GO TO 105	DIRECT	97
	AD=POL	DIRECT	98
105	IF (POL.GT.90.) AD=POL-180.	DIRECT	99
	I4=I4+2	DIRECT	100
	HAZ=RAZI(I4)	DIRECT	101
	AMAD=AMIN1(ABS(AD),AMAD)	DIRECT	102
	IF (AMAD.EQ.ABS(AD)) I5=I2	DIRECT	103
110	GO TO 125	DIRECT	104
	105 RPD = RPOL(I2)	DIRECT	105
	IF (POL.GT.90.) RPU = PI-RPOL(I2)	DIRECT	106
	DZR=DBR*CUS(RPU)	DIRECT	107
	IF (DEFECT) DZN=DAN*COS(RPD)	DIRECT	108

```

115      C      NM4 = 2
           IF(I2.EQ.0.OR,I2.EQ.9.OR,I2.EQ.11.OR,I2.EQ.13.OR,I2.EQ.14) NM4=4
           IF(I2.EQ.15) NM4=12
           DO 120 I3=1,NM4
120      I4 = I4 + 1
           RAZ=RAZI(I4)
           107 IF (DEFECT) GO TO 108
           DHRO=SQRT(DHR**2-UZH**2)
           DXR=DHRO*COS(MAZ)
125      DYR=DHRO*SIN(MAZ)
           AD=RCST*ACOS((DXR*DXR+DYR*DYR+UZH*UZH)/DHR**2)
           GO TO 110
      C
130      108 DHRO=SQRT(DAR**2-UZH**2)
           DXR=DHRO*COS(MAZ)
           DYR=DHRO*SIN(MAZ)
           AD=RCST*ACOS((DXR*DXR+DYR*DYR+UZH*UZH)/DAR**2)
      C
135      110 AMAD=AMINI(ABS(AD),AMAD)
           IF(AMAD.EQ.ABS(AD)) I5=I2
           IF(RAZ,GE,PI) GO TO 115
           RAZ=RAZ+PI
           GO TO 107
140      115 IF(I5.NE.15) GO TO 120
           AZP=RAZI/RCST
           IF(AZP.GT,PI) AZP=AZP-PI
           IF(AZP.GT,PI/2.) AZP=AZP-PI/2.
           DA1=ABS(AZP-RAZI(39))
           DA2=ABS(AZP-RAZI(41))
145      DA3=ABS(AZP-RAZI(42))
           DA4=ABS(AZP-RAZI(43))
           DA5=ABS(AZP-RAZI(44))
           DA6=ABS(AZP-RAZI(47))
           DA7=ABS(AZP-RAZI(48))
150      AMDA=AMINI(DA1,DA2,DA3,DA4,DA5,DA6,DA7)
           IF(AMDA.EQ,DA1) I5=15
           IF(AMDA.EQ,DA2) I5=16
           IF(AMDA.EQ,DA3) I5=17
           IF(AMDA.EQ,DA4) I5=18
155      IF(AMDA.EQ,DA5) I5=19
           IF(AMDA.EQ,DA6) I5=20
           IF(AMDA.EQ,DA7) I5=21
           120 CONTINUE
           125 CONTINUE
160      IF(I5.EQ,0) GO TO 140
           WRITE(6,155)
           WRITE(6,150) AMAD,DIR(I5),PUL,AZI
           RETURN
165      140 WRITE(6,150)
           RETURN
           150 FORMAT(T10,' THIS ATOM IS',F8.4,' DEGREE FROM A <',A5,'> DIRECTIO
           IN WITH POLAR AND AZIMUTHAL',T10,' ANGLES',F9.4,' AND',F9.4,' , RESP
           2ECTIVELY.')
```

1		SUBROUTINE DSLOAN	DSLOAN	2
	C		DSLOAN	3
	C	THIS SUBROUTINE WILL ANALYZE THE STRAIN FIELD AND ENERGY	DSLOAN	4
	C	ASSOCIATED WITH AN (A/2) <110> EDGE DISLOCATION IN AN FCC LATTICE.	DSLOAN	5
5	C	NOTE - THIS SUBROUTINE IS NOT COMPLETE AT THIS TIME.	DSLOAN	6
	C		DSLOAN	7
		RETURN	DSLOAN	8
		END	DSLOAN	9

1		SUBROUTINE DSLOIN	DSLOIN	2
	C		DSLOIN	3
	C	THIS SUBROUTINE WILL INSERT AN (A/2) <110> EDGE DISLOCATION INTO	DSLOIN	4
	C	AN FCC LATTICE.	DSLOIN	5
5	C	NOTE - THIS SUBROUTINE IS NOT COMPLETE AT THIS TIME.	DSLOIN	6
	C		DSLOIN	7
		RETURN	DSLOIN	8
		END	DSLOIN	9

1		SUBROUTINE EXPAND (N1)	EXPAND	2
	C		EXPAND	3
	C	THIS ROUTINE EXPANDS A SMALL THERMALIZED LATTICE INTO EITHER A	EXPAND	4
	C	MEDIUM OR LARGE THERMALIZED LATTICE (ALL ATOM COORDINATES	EXPAND	5
5	C	AND IDENTIFICATION NUMBERS ARE CHANGED ACCORDINGLY.)	EXPAND	6
	C		EXPAND	7
		COMMON RMIN, DT, N, NL, VX(6460), TB(6460), TEMP, DEFECT,	COM1	2
		+ DTN, NS, NLB, NA, UVC, UMC, UF	COM1	3
		COMMON /ALPHA/ X(6460), ZZX(6460), ZZY(6460), ZZZ(6460),	COM1	4
10		+ ENGPI(6460), IDENT(6460), IBUX(14600), EA(1001), EEA(1001),	COM1	5
		+ EB(1001), EEB(1001), EAB(1001), EAB(1001)	COM1	6
		COMMON /ELMT/ ENGPU, ID, E1, E2, J3, DIV, RMAX2, INIB, N12, ML,	COM1	7
		1 IELE, L, LI, IOT	COM1	8
		LEVEL, Z, X, ZZX, ZZY, ZZZ, ENGPI, IDENT, IBUX, EA, EEA, EB, EEB,	COM1	9
15		+ EAB, EEA	COM1	10
		DATA ZST/45.6629/	COM4	2
	C		EXPAND	10
		NA=0	EXPAND	11
		NK=164*6*3	EXPAND	12
20		NK1=164*4*2	EXPAND	13
		NM1=N1-NK1+1	EXPAND	14
		NM2=N-NK1+1	EXPAND	15
		J=NM2	EXPAND	16
	C		EXPAND	17
25	C	MOVE 8 PLANES OF BOUNDARY ATOMS TO END OF NEW LATTICE	EXPAND	18
	C	*****	EXPAND	19
	C		EXPAND	20
		DO 10 ID1=NM1,N1	EXPAND	21
		II=IDENT(J)	EXPAND	22
30		CALL MTBIN (II,I,L,M,J)	EXPAND	23
		I=ID1	EXPAND	24
		IF(M.EQ.1) NA=NA+1	EXPAND	25
		CALL FILBIN (II,I,L,M,0)	EXPAND	26
		IDENT(ID1)=II	EXPAND	27
35		X(ID1)=X(J)	EXPAND	28
		VX(ID1)=VX(J)	EXPAND	29
		TB(ID1)=TB(J)	EXPAND	30
		J=J+1	EXPAND	31
		10 CONTINUE	EXPAND	32
40	C		EXPAND	33
	C	EXPAND TO LARGER LATTICE IN SEIS OF 16 PLANES AT A TIME	EXPAND	34
	C	*****	EXPAND	35
	C		EXPAND	36
		ID1=0	EXPAND	37
45		DO 20 KK=1,4	EXPAND	38
		DO 20 K=1,NK	EXPAND	39
		ID1=ID1+1	EXPAND	40
		IF(ID1.GT.NM1=1) GO TO 30	EXPAND	41
		II=IDENT(K)	EXPAND	42
50		CALL MTBIN (II,I,L,M,J)	EXPAND	43
		I=ID1	EXPAND	44
		IF(M.EQ.1) NA=NA+1	EXPAND	45
		CALL FILBIN (II,I,L,M,0)	EXPAND	46
		IDENT(ID1)=II	EXPAND	47
55		X(ID1)=X(K)	EXPAND	48
		VX(ID1)=VX(K)	EXPAND	49
		TB(ID1)=TB(K)	EXPAND	50

	20	CUNTINUE	EXPAND	51
	C		EXPAND	52
60	C	EXPAND ALL Z=COORDINATES IN NEW LATTICE	EXPAND	53
	C	*****	EXPAND	54
	C		EXPAND	55
	30	K=5	EXPAND	56
		N = N1	EXPAND	57
65		NSLB=N/164	EXPAND	58
		ID1=1	EXPAND	59
		DELZ=SQRT(3./2.)/5.0	EXPAND	60
		DO 50 KK=1,NSLB	EXPAND	61
		Z11=ZST-FLOAT(K-1)*DELZ	EXPAND	62
70		DO 40 I=1,164	EXPAND	63
		A=X(ID1)	EXPAND	64
		CALL UNPAQ3 (A,X1,Y1,Z1)	EXPAND	65
		A=VX(ID1)	EXPAND	66
		CALL UNPAQ3 (A,VX1,VY1,VZ1)	EXPAND	67
75		A=TB(ID1)	EXPAND	68
		CALL UNPAQ3 (A,TX1,TY1,TZ1)	EXPAND	69
	C		EXPAND	70
		DELZ1=Z1-TZ1	EXPAND	71
		DELVZ=VZ1-TZ1	EXPAND	72
80	C		EXPAND	73
		CALL PAC3 (A,X1,Y1,Z11+DELZ1)	EXPAND	74
		X(ID1)=A	EXPAND	75
		CALL PAC3 (A,VX1,VY1,Z11+DELVZ)	EXPAND	76
		VX(ID1)=A	EXPAND	77
85		CALL PAC3 (A,TX1,TY1,Z11)	EXPAND	78
		TB(ID1)=A	EXPAND	79
	C		EXPAND	80
		ID1=ID1+1	EXPAND	81
	40	CUNTINUE	EXPAND	82
90		K=K+1	EXPAND	83
		IF(KK,NE,(NSLB-4)) GO TO 50	EXPAND	84
		K=1	EXPAND	85
	50	CUNTINUE	EXPAND	86
		RETURN	EXPAND	87
95		END	EXPAND	88

5-48

```

1      SUBROUTINE IDDEF (DN,ENG2,ECUT,IPI,UDCC,NB,LPRNT,LPRNT1,LPRNT2)  IDDEF      2
      C      IDDEF      3
      C      THIS SUBROUTINE IDENTIFIES DEFECTS AND LOCATES DISTURBANCES NEAR  IDDEF      4
      C      THE LATTICE-CONTINUUM INTERFACE. IDDEF ALSO DETERMINES WHETHER  IDDEF      5
      C      OR NOT TO INCREASE MINI-LATTICE SIZE.  IDDEF      6
      C      IDDEF      7
      COMMON RMIN, DT, N, NL, VX(6460), TB(6460), TEMP, DEFECT,  COM1      2
      + DTN, NS, NLS, NA, UVC, UHC, UP  COM1      3
      COMMON /ALPHA/ X(6460), ZZ(6460), ZZY(6460), ZZZ(6460),  COM1      4
      + ENGP1(6460), IDENT(6460), IBUX(14600), EA(1001), EEA(1001),  COM1      5
      + EB(1001), EEB(1001), EAB(1001), EEA(1001)  COM1      6
      COMMON /ELMT/ ENGPU, ID, E1, E2, J3, DIV, RMAX2, INIB, N12, ML,  COM1      7
      + IELE, L, LT, IOT  COM1      8
      LEVEL 2, X, ZZ, ZZY, ZZZ, ENGP1, IDENT, IBUX, EA, EEA, EB, EEB,  COM1      9
      + EAB, EEA  COM1     10
      COMMON /BUX/ IX(17,125)  IDDEF     11
      COMMON /CENTR/ X11, Y11, Z11, AX1, YY1, ZZ1, VX11, VY11, VZ11,  IDDEF     12
      + TBX1, TBY1, TBZ1, UX1, DY1, DX1, AZ1, POL  IDDEF     13
      COMMON /HBOX/ MEA,MEA1,NID,ERL,ERL1,XIPS,YIPS,ZIPS,SL,IP,NR,  IDDEF     14
      + RESET,CONT,LFINST,XEA,YEA,ZEA,ADAMP,IC,IAFIX,FTAPE  IDDEF     15
      COMMON /CNST/ CL, CM, CN, CU, CP, CDD, CD  IDDEF     16
      DIMENSION IDN(4)  IDDEF     17
      LOGICAL FTAPE,LPRNT,LPRNT1,LPRNT2  IDDEF     18
      C      IDDEF     19
      XTRAN(A,B,C) = .424264*A+.346411*B+.244943*C  IDDEF     20
      YTRAN(A,B,C) = -.424264*A+.346411*B+.244943*C  IDDEF     21
      ZTRAN(A,B) = -.346411*A+.489886*B  IDDEF     22
      RDIST(A,B,C) = (A**2+B**2+C**2)**0.5  IDDEF     23
      BUXNO(A,B,C) = IFIX(A-31.)*J1+IFIX(B-32.)*J2+IFIX(C-32.)*J3  IDDEF     24
      C      IDDEF     25
      II=IDENT(10)  IDDEF     26
      CALL MTBIN (II,K,L,M,J)  IDDEF     27
      NVC1=0 3 NVC2=0 3 NRC=0  IDDEF     28
      J00=J  IDDEF     29
      RAM=5.0  IDDEF     30
      JDT = 5  IDDEF     31
      IF (DR.LT.CM) GO TO 220  IDDEF     32
      DO 155 J0=1,2  IDDEF     33
      IF (J0.EQ.1) GO TO 10  IDDEF     34
      IF (DR.LT.CDD) GO TO 155  IDDEF     35
      C      IDDEF     36
      C      WHEN ATOM ID IS FAR FROM ITS LATTICE SITE  IDDEF     37
      C      *****  IDDEF     38
      C      IDDEF     39
      A=TB(ID)  IDDEF     40
      CALL UNPAC3 (A,XC,YC,ZC)  IDDEF     41
      J=BOXNO(XC,YC,ZC)  IDDEF     42
      NVC1=0 3 NVC2=0 3 NRC=0  IDDEF     43
      C      IDDEF     44
      10 JDT=4  IDDEF     45
      DO 150 J1=1,125  IDDEF     46
      J2=J+IK(1,J1)  IDDEF     47
      IF (J2.LE.U.OR.J2.GT.NB) GO TO 150  IDDEF     48
      J5=IBUX(J2)  IDDEF     49
      IF (J5.EQ.U) GO TO 150  IDDEF     50
      CALL MTBIN (J5,IDN(1),IDN(2),IDN(3),IDN(4))  IDDEF     51
      DO 100 J4=1,4  IDDEF     52
      IDI=IDN(J4)  IDDEF     53

```

```

107 IDDEF IF(IDI,EO,0) GO TO 100
106 IDDEF A=X(IDI)
105 IDDEF CALL UNPAC3 (A,X,XN,YYN,ZZN)
104 IDDEF I=IDENI(IDI)
103 IDDEF CALL MTRIN (I1,XN,LM,MN,JN)
102 IDDEF IF(IDI,EO,10) GO TO 20
101 IDDEF DXN = XXN-X11
100 IDDEF DYN = YYN-Y11
99 IDDEF DZN = ZZN-Z11
98 IDDEF KAI=RDIST(DXN,DYN,DZNN)
97 IDDEF HAM=AMINI(KAM,KAI)
96 IDDEF IF(RAM,EO,RAI) NNID=IDI
95 IDDEF DXN = XXN-X11
94 IDDEF DYN = YYN-Y11
93 IDDEF DZN = ZZN-Z11
92 IDDEF OXN = XN-X11
91 IDDEF IF(DXM,GT,CN) GO TO 40
90 IDDEF JDI=0
89 IDDEF IF(K,EO,KN) GO TO 50
88 IDDEF JDI=1
87 IDDEF IF(NVC1,NE,0) NVC2=URK*UDCC
86 IDDEF IF(NVC1,NE,0) NVC1=KN
85 IDDEF NVC1=DRK*UDCC
84 IDDEF GO TO 40
83 IDDEF NVC1=KN
82 IDDEF CALL UNPAC3 (A,IXN,IYN,IZN)
81 IDDEF DXN = IXN-X11
80 IDDEF DYN = IYN-Y11
79 IDDEF DZN = IZN-Z11
78 IDDEF OMN=HOIST(DXN,DYN,DZNN)
77 IDDEF DRU=DRK*UDCC
76 IDDEF DETERMINE WHETHER OR NOT TO INCREASE MINI-LATTICE SIZE
75 IDDEF *****
74 IDDEF IF(ORN,GT,CO,UM,K,EW,KN) GO TO 100
73 IDDEF IF(M,GT,9,OR,MN,ME,93) GO TO 50
72 IDDEF RESGT=1.0
71 IDDEF CU=CO*UDCC
70 IDDEF IF(LPMN1) WRITE(6,100) K,CU,DMU,KN
69 IDDEF IF(LPMN1) WRITE(6,100)
68 IDDEF IF(ORN,GT,CP) GO TO 100
67 IDDEF OIXN = IXN-I1XN
66 IDDEF OIYN = IYN-I1YN
65 IDDEF OIZN = IZN-I1ZN
64 IDDEF OIRN=RDIST(OIXN,OIYN,OIZN)*UDCC
63 IDDEF IF (LPMNTD) WRITE (6,170)
62 IDDEF IF (LPMNTD) WRITE (6,190) K,DMU,KN,DIRN
61 IDDEF CONTINUE
60 CONTINUE
59 J = J00
58 CONTINUE
57 150 CONTINUE
56 155 CONTINUE
55 A=VX(NNID)

```

67-5

```

115      CALL UNPAC3 (A,VXIN,VYIN,VZIN)          IDDEF 108
        DVXN = VXIN-VX11                        IDDEF 109
        DYYN = VYIN-VY11                        IDDEF 110
        DVZN = VZIN-VZ11                        IDDEF 111
        RA2=ROIST(DVXN,DVYN,DVZN)               IDDEF 112
120      RNM = AMIN1(RAM,RA2)                     IDDEF 113
        IF(RNM.EQ.0) GO TO 210                   IDDEF 114
        II=IDENT(NNID)                           IDDEF 115
        CALL MTBIN (II,1NN,LNN,MNN,JNN)         IDDEF 116
        A=TB(NNID)                               IDDEF 117
125      CALL UNPAC3 (A,TXN,1YN,TZN)             IDDEF 118
        DTX = TXN-XIPS                           IDDEF 119
        DTY = TYN-YIPS                           IDDEF 120
        DTZ = TZN-ZIPS                           IDDEF 121
        DX4=X1TRAN(DTX,DTY,DTZ)                 IDDEF 122
130      DY4=Y1TRAN(DTX,DTY,DTZ)                 IDDEF 123
        DZ4=Z1TRAN(DTY,DTZ)                     IDDEF 124
        IF(LPNT1.AND.ENG2.GT.ECUT.UR.10.EQ.ME1.OR.K.EQ.IPI) WRITE(6,200) IDDEF 125
        + K,KNN,DX4,DY4,DZ4                     IDDEF 126
        WRITE(6,170)                             IDDEF 127
135      210 IF(NVC1.NE.0.AND.NVC2.NE.0) JDT=3    IDDEF 128
        IF(NVC1.NE.0.AND.NVC.NE.0) JDT=2        IDDEF 129
        CN1=CN+UDCC                             IDDEF 130
        IF(.NOT.LPNT,OR.JDT.EQ.5,OR.JDT.EQ.0) GO TO 220 IDDEF 131
        IF(JDT.EQ.1) WRITE(6,300) NVC1,RVC1,K   IDDEF 132
140      IF(JDT.EQ.2) WRITE(6,310) NVC1,NVC,RVC1,RC,K IDDEF 133
        IF(JDT.EQ.3) WRITE(6,310) NVC1,NVC2,RVC1,RVC2,K IDDEF 134
        IF(JDT.EQ.4) WRITE(6,320) CN1,K         IDDEF 135
        IF(LPNT1) WRITE(6,160)                  IDDEF 136
145      220 IF(JDT.EQ.5) JDT=0                  IDDEF 137
        IF(FTAPE) WRITE(4) JDT                  IDDEF 138
        RETURN                                  IDDEF 139
        170 FORMAT(//)                          IDDEF 140
        180 FORMAT(//)                          IDDEF 141
150      180 FORMAT (T40,'ATOM ID =',I5,' IS WITHIN',1PE12.4,' OF THE CRYSTALLI IDDEF 142
        +TE SURFACE(I,E,'E12.4,' FROM ATOM',I5,')/T40,'MUST INCREASE THE IDDEF 143
        +SIZE OF THE CRYSTALLITE.')            IDDEF 144
        190 FORMAT (T40,'ATOM',I5,' IS',1PE12.4,' FROM SITE',I5,' - DISTANCE 0 IDDEF 145
        +ETWEEN RESPECTIVE LATTICE SITES IS',E12.4) IDDEF 146
        200 FORMAT (T40,'ATOM',I6,' IS APPROACHING ATOM',I6,' WHICH WAS INITIA IDDEF 147
        +LLY AT SITE',3F8.3)                    IDDEF 148
155      300 FORMAT (25X,'REPLACEMENT --',T40,'ATOM',I5,' IS',1PE12.4,' FROM SI IDDEF 149
        +TE',I5)                                IDDEF 150
        310 FORMAT (24X,'INTERSTITIAL --',I40,'ATOMS',I5,' AND',I5,' ARE',1PE1 IDDEF 151
        +2.4,' AND',E12.4,' RESPECTIVELY FROM SITE',I5) IDDEF 152
160      320 FORMAT (29X,'VACANCY --',T40,'NO ATOMS ARE WITHIN',1PE12.4,' OF SI IDDEF 153
        +TE',I5)                                IDDEF 154
        END                                      IDDEF 155

```


5-52

```

+ -2.221407,1.50/604,-.080144,2.186182,-1.575972,0./      IFAF      51
DATA (AK(2,I),I=1,10) /-16.062,3.835,-2.861,-.27878,-2.4058,      IFAF      52
+ 1.596,-.061795,.85416,2*0./      IFAF      53
DATA (AK(3,I),I=1,10) /-18.001,-1.7761,270.71,-175.23,.4533,      IFAF      54
+ 9.6189,-32.761,.45348,2*0./      IFAF      55
DATA (AK(4,I),I=1,10) /-31.717,34.621,-21.268,5.942,-11.888,4.304,      IFAF      56
+ .14028,-.063633,2*0./      IFAF      57
DATA (AK(5,I),I=1,10) /-49.649,37.568,-36.206,-.89031,.19628,      IFAF      58
+ .15577,-.41571,3.3067,2*0./      IFAF      59
DATA (AK(6,I),I=1,10) /-22.171,20.465,-8.8735,.63054,-1.6269,      IFAF      60
+ 2.3841,.018713,3*0./      IFAF      61
DATA (AK(7,I),I=1,10) /-25.62,11.475,-99.383,2.4053,-3.066,2.931,      IFAF      62
+ .29947,-.2967,2*0./      IFAF      63
DATA (BK(1,I),I=1,10) /1081.8838,79.9655,6.3981,1.045285,.650164,      IFAF      64
+ -1.219882,-.017443,-.162063,.307171,0./      IFAF      65
DATA (BK(2,I),I=1,10) /10.536,.89861,3.1996,1.0453,.55321,-1.2512,      IFAF      66
+ -.017476,-.12729,2*0./      IFAF      67
DATA (BK(3,I),I=1,10) /13.537,3.4425,1.2498,24.883,-1.4008,.10866,      IFAF      68
+ 4.1486,-.76555,2*0./      IFAF      69
DATA (BK(4,I),I=1,10) /22.648,-5.8965,14.876,-1.3155,4.8564,      IFAF      70
+ -5.843,.11486,.46962,2*0./      IFAF      71
DATA (BK(5,I),I=1,10) /29.485,-.30454,22.236,1.4028,-.22162,      IFAF      72
+ -.074416,.0068497,-.71163,2*0./      IFAF      73
DATA (BK(6,I),I=1,10) /11.308,-1.4943,10.283,.60831,1.2448,-1.6835      IFAF      74
+ .10171,3*0./      IFAF      75
DATA (BK(7,I),I=1,10) /13.077,-9.481,33.204,-3.2901,.15628,      IFAF      76
+ -2.6031,-.25364,.46662,2*0./      IFAF      77
DATA (CK(1,I),I=1,10) /-628.5649,-47.6408,-4.4591,-.357854,-.507164      IFAF      78
+ .347295,.016333,-.08962,-0.054405,0./      IFAF      79
DATA (CK(2,I),I=1,10) /-4.5298,-2.243,-1.4233,-.35785,.51973,      IFAF      80
+ .34524,.016354,-.067396,2*0./      IFAF      81
DATA (CK(3,I),I=1,10) /-5.9307,-2.8654,-.86936,-.10889,1.0652,      IFAF      82
+ -.3691,.22691,.34606,2*0./      IFAF      83
DATA (CK(4,I),I=1,10) /-9.4173,-4.3917,-2.5958,.84537,2.0713,      IFAF      84
+ 1.7753,-.8195,-.32634,2*0./      IFAF      85
DATA (CK(5,I),I=1,10) /-15.138,-4.302,-4.9157,-.38175,.33667,      IFAF      86
+ .26266,.25091,-.15512,2*0./      IFAF      87
DATA (CK(6,I),I=1,10) /-7.6778,-5.815,-4.1569,-.19722,.42637,      IFAF      88
+ .16318,-.23164,3*0./      IFAF      89
DATA (CK(7,I),I=1,10) /-6.6588,-5.73,-1.0853,2.5762,1.0795,      IFAF      90
+ .34541,-.41741,-.24716,2*0./      IFAF      91
DATA (DK(1,I),I=1,10) /138.11,10.005,.8453,-.21093,-.155699,      IFAF      92
+ -.011272,.023168,.010455,.001445,0./      IFAF      93
DATA (DK(2,I),I=1,10) /-.98659,.37356,-.008403,-.20932,-.14016,      IFAF      94
+ -.013247,.014944,.01184,2*0./      IFAF      95
DATA (DK(3,I),I=1,10) /1.3115,.57406,-.14569,-.16326,-.1284,      IFAF      96
+ -.05203,-.075134,-.057562,2*0./      IFAF      97
DATA (DK(4,I),I=1,10) /1.9819,.33872,-.49852,-.54685,-.16523,      IFAF      98
+ .57226,.58449,.059664,2*0./      IFAF      99
DATA (UK(5,I),I=1,10) /3.9597,1.7143,.14227,-.23802,-.15158,      IFAF      100
+ -.078197,-.033452,.033383,2*0./      IFAF      101
DATA (UK(6,I),I=1,10) /2.6783,1.4178,.3387,-.23956,-.21302,.13954,      IFAF      102
+ .11246,3*0./      IFAF      103
DATA (DK(7,I),I=1,10) /1.7269,.21445,-.75498,-.57261,.16933,      IFAF      104
+ .42445,.38681,.043059,2*0./      IFAF      105
DATA CLP/3.6147,2*3.608,3*3.6147,3.60766,4.06,3.753,3.524,4.0496,      IFAF      106
+ 3.5672/      IFAF      107

```


Line	Column	Code	Text	Line	Column	Code	Text
115			DATA XN/U.5,1.0,2*U.5,1.,7*0.5/ DATA XMX/4.005,2*4.418461,4.005,2*3.2,4.41846,2*4.97246,4.316, + 2*4.95973/ DATA Z/2*29.,2*79.,29.,79.,2*20.,2*13.,26.,13./ DATA SR1/.0730,.0/35,.071,1.091,.1255,.0901/ 120 C C C C C C NOTE == WHEN IELE .GT. 1, THE MAXIMUM RANGE OF THE 3 POTENTIALS MUST BE USED SO THAT THEY HAVE A COMMON RMIN, RMAX, E1, AND E2. HENCE, FOR POTENTIALS WITH LESSEN RANGE, THEIR FORCE AND ENERGY MUST BE SET TO ZERO. 125 C UDCC = .6*CLP(NPUTA) IF (ALLOY) UDCC=.6*CLP(NPUTAB) RMIN = RMIN(NPUTA)/UDCC RMAX = RMAX(NPUTA)/UDCC 130 IF (IELE.GT.1) XMIN = AMIN1(RMIN(NPOTA),RMIN(NPOTB),RMIN(NPOTAB))/UDCC +C IF (IELE.GT.1) XMAX = AMAX1(RMAX(NPOTA),RMAX(NPOTB),RMAX(NPOTAB))/UDCC +C RM = RM1/UDCC RE = RE1/UDCC 135 DU 20 K=1,7 DO 20 I=1,10 RK(K,I) = RK(K,I)/UDCC AK(K,I) = AK(K,I)*UDCC**3 BK(K,I) = BK(K,I)*UDCC**2 140 CK(K,I) = CK(K,I)*UDCC 20 CONTINUE IF (NPUTA.NE.3) GO TO 25 CALL PUT1 (RM,C1,C2,C3) 145 C DM = MM-RK(1,3) DE = RE-RK(1,3) D1 = DM**5-10.*DE**3*DM**2+15.*DE**4*DM-6.*DE**5 D2 = DM**4-6.*DE**2*DM**2+8.*DE**3*DM-3.*DE**4 150 D3 = DM**3-3.*DE*DM**2+3.*DE**2*DM-DE**3 D4 = C1-AK(1,3)*(3.*DE*DM**2-3.*DE**2*DM+DE**3)*BK(1,3)*DM**2-CK(1 + 3)*DM-DK(1,3) D5 = 5.*DM**4-20.*DE**3*DM+15.*DE**4 D6 = 4.*DM**3-12.*DE**2*DM+8.*DE**3 155 D7 = 3.*DM**2-6.*DE*DM+3.*DE**2 D8 = C2-AK(1,3)*(6.*DE*DM-3.*DE**2)-2.*BK(1,3)*DM-CK(1,3) D9 = 20.*DM**3-20.*DE**3 D10 = 12.*DM**2-12.*DE**2 D11 = 6.*DM-6.*DE 160 D12 = C3-6.*AK(1,3)*DE-2.*BK(1,3) D1 = ((D8-D4*D5/D1)*(D10-D2*D9/D1)-(D12-D4*D9/D1)*(D6-D2*D5/D1))/(+ (D7-D3*D5/D1)*(D10-D2*D9/D1)-(D11-D3*D9/D1)*(D6-D2*D5/D1)) D2 = ((D12-D4*D9/D1)-D1*(D11-D3*D9/D1))/(D10-D2*D9/D1) 165 D6 = (D4-DH*D2-D1*U3)/D1 DJ = -10.*DG*DE**3-6.*DH*DE**2+3.*(AK(1,3)-D1)*DE+BK(1,3) DKK = 15.*DG*DE**4+6.*DH*DE**3+3.*(AK(1,3)-D1)*DE**2+CK(1,3) DL = 6.*DG*DE**5+3.*UH*DE**4+(AK(1,3)-D1)*DE**3+DK(1,3) C 25 L2=1 170 IF (IELE.GT.1) L2 = 3 DU 210 K=1,L2				

5-54

	IF (K=2) 30,40,50	IEAF	165
30	NPOT = NPUTA	IEAF	166
	GU TO 60	IEAF	167
175	40 NPOT = NPUTB	IEAF	168
	GU TO 60	IEAF	169
50	NPOT = NPUTAB	IEAF	170
	60 IF(NPUT,LI,7) GO TO 70	IEAF	171
	SK = SH1(NPUT=6)	IEAF	172
180	ZA = Z(1,NPUT=6)	IEAF	173
	ZB = Z(2,NPUT=6)	IEAF	174
70	WW = RMIN**2	IEAF	175
	H = (RMAX**2-WW)/1000,	IEAF	176
	DO 210 I=1,1001	IEAF	177
185	W = SQRT(WW)	IEAF	178
	GU TO (80,90,100,110,120,130,140,140,140,140,140), NPOT	IEAF	179
80	CALL POT1 (W,F0,F1,WM)	IEAF	180
	GU TO 150	IEAF	181
90	CALL POT2 (W,F0,F1)	IEAF	182
190	GU TO 150	IEAF	183
100	CALL POT3 (W,F0,F1)	IEAF	184
	GU TO 150	IEAF	185
110	CALL POT4 (W,F0,F1)	IEAF	186
	GU TO 150	IEAF	187
195	120 CALL POT5 (W,F0,F1)	IEAF	188
	GU TO 150	IEAF	189
130	CALL POT6 (W,F0,F1)	IEAF	190
	GU TO 150	IEAF	191
140	CALL POT7 (W,F0,F1,NPOT)	IEAF	192
200	150 IF(W*UDCC,GE,RMN(NPUT)=.001*AND,W*UDCC,LE,RMX(NPOT)) GO TO 160	IEAF	193
	F0 = 0.05F1 = 0.0	IEAF	194
160	IF (K=2) 170,180,190	IEAF	195
170	EA(I) = -F1/W*EEA(I) = F0	IEAF	196
	GU TO 200	IEAF	197
205	180 EB(I) = -F1/W*EEB(I) = F0	IEAF	198
	GU TO 200	IEAF	199
190	EAB(I) = -F1/W*EEAB(I) = F0	IEAF	200
200	WW = WW*H	IEAF	201
210	CONTINUE	IEAF	202
210	E1 = 1./H	IEAF	203
	E2 = ((-RMIN**2)/H)+1.	IEAF	204
	IF (.NOT,LPRINT2) RETURN	IEAF	205
	WRITE (6,UPUT)	IEAF	206
215	IF (IELE,EG,1) WRITE (6,260)	IEAF	207
	IF (IELE,GT,1) WRITE (6,250)	IEAF	208
	WW = RMIN**2	IEAF	209
	DO 220 I=1,1001	IEAF	210
	W = SQRT(WW)	IEAF	211
	R = W*UDCC	IEAF	212
220	IF (IELE,EG,1) WRITE (6,230) I,R,EEA(I),-EA(I)*W/UDCC	IEAF	213
	IF (IELE,GT,1) WRITE (6,240) I,R,EEA(I),-EA(I)*W/UDCC,EEB(I),-EB(I)*W/UDCC,EEAB(I),-EAB(I)*W/UDCC	IEAF	214
	WW = WW*H	IEAF	215
220	CONTINUE	IEAF	216
225	RETURN	IEAF	217
	C	IEAF	218
	C	IEAF	219
	C	IEAF	220
	C	IEAF	221

230	FORMAT (I10,1P3E15.4)	IEAF	222
240	FORMAT (I10,1P7E15.4)	IEAF	223
250	FORMAT (////,I50,'INTERATOMIC POTENTIAL AND FORCE ARRAYS'//T10,'I'	IEAF	224
	*,T25,'R',I30,'POTENTIAL=A',T47,'FORCE=A',T60,'POTENTIAL=B',T77,'FO	IEAF	225
	*NCE=B',T89,'POTENTIAL=AB',T100,'FORCE=AB'//)	IEAF	226
260	FORMAT (////,I15,'INTERATOMIC POTENTIAL AND FORCE ARRAYS'//T10,'I'	IEAF	227
235	*,T23,'R',I31,'POTENTIAL',T40,'FORCE'//)	IEAF	228
	END	IEAF	229

5-56

```

1      SUBROUTINE INCBOX                                INCBOX    2
      C                                                INCBOX    3
      C      THIS SUBROUTINE GENERATES INCREMENTED ARRAYS FOR USE IN THE 'BOX INCBOX    4
      C      SEARCH'. THERE IS AN ARRAY FOR EACH BOUNDARY TYPE CLASS. TOGETHER INCBOX    5
      C      WITH THE BOX IDENTIFICATION NUMBER J, THEY ARE USED TO DETERMINE INCBOX    6
      C      THE IDENTIFICATION NUMBERS OF SURROUNDING BOXES (AND, HENCE, FIND INCBOX    7
      C      ATOMS IN THEM).                                INCBOX    8
      C                                                INCBOX    9
      C      COMMON /BOX/ IK(17,125)                    INCBOX   10
10     DIMENSION LD(17,6),LD1(4,4),LD2(4,4)             INCBOX   11
      C                                                INCBOX   12
      C      DATA LD/13*1,2,3,2*1,          15*5,3,4, INCBOX   13
      C      +      2*1,2,1,2,3*1,2,8*1,      3*5,3*(4,5),2*(1,3),4*5, INCBOX   14
      C      +      1,3*2,5*1,5,7*1,          6*5,3*4,2*5,3,5*5/ INCBOX   15
15     DATA LD1/4*(2,4),2*2,2*1,2*5,2*4/             INCBOX   16
      C      DATA LD2/2*(3,5),4*5,1,3,2*1,3*5,5/ INCBOX   17
      C                                                INCBOX   18
      C      DO 20 K=1,17                                INCBOX   19
      C      DO 20 J=1,125                                INCBOX   20
20     IK(K,J) = 0                                       INCBOX   21
      C                                                INCBOX   22
      C      GENERATE FIRST AREA FOR PATTERNS 1 THROUGH 17 INCBOX   23
      C                                                INCBOX   24
      C      DO 60 K=1,17                                INCBOX   25
      C      M1 = LD(K,1)SM2 = LD(K,2)SM3 = LD(K,3)SM4 = LD(K,4) INCBOX   26
25     M5 = LD(K,5)SM6 = LD(K,6)                        INCBOX   27
      C      J = 1                                       INCBOX   28
      C      DO 30 I3=M1,M2                              INCBOX   29
      C      CC = 961*(I3-3)                             INCBOX   30
30     DO 30 I1=M3,M4                                  INCBOX   31
      C      CC1 = 31*(I1-3)                             INCBOX   32
      C      DO 30 I2=M5,M6                              INCBOX   33
      C      IK(K,J) = I2-3+CC1+CC                      INCBOX   34
      C      J = J+1                                    INCBOX   35
35     30 CONTINUE                                     INCBOX   36
      C      IF (K.LE.9.OR.K.GE.14) GO TO 60             INCBOX   37
      C                                                INCBOX   38
      C      GENERATE SECOND AREA FOR PATTERNS 10 THROUGH 13 INCBOX   39
      C                                                INCBOX   40
      C      M3 = LD1(K-9,1)SM4 = LD1(K-9,2)SM5 = LD1(K-9,3)SM6 = LD1(K-9,4) INCBOX   41
40     DO 40 I3=1,5                                     INCBOX   42
      C      CC = 961*(I3-3)                             INCBOX   43
      C      DO 40 I1=M3,M4                              INCBOX   44
      C      CC1 = 31*(I1-3)                             INCBOX   45
45     DO 40 I2=M5,M6                                  INCBOX   46
      C      IK(K,J) = I2-3+CC1+CC                      INCBOX   47
      C      J = J+1                                    INCBOX   48
      C      40 CONTINUE                                INCBOX   49
      C                                                INCBOX   50
      C      GENERATE THIRD AREA FOR PATTERNS 10 THROUGH 13 INCBOX   51
      C                                                INCBOX   52
      C      M3=LD2(K-9,1)SM4=LD2(K-9,2)SM5=LD2(K-9,3)SM6=LD2(K-9,4) INCBOX   53
50     DO 50 I3=1,5                                     INCBOX   54
      C      CC = 961*(I3-3)                             INCBOX   55
      C      DO 50 I1=M3,M4                              INCBOX   56
      C      CC1 = 31*(I1-3)                             INCBOX   57
55     DO 50 I2=M5,M6                                  INCBOX   58

```

SUBROUTINE INCBUX 76/76 OPT=0, TRACE

FTN 4.5+414

11/13/79 14.17.08

PAGE 2

```
60      50      IK(K,J) = I2-3+CC1+CC  
      60      J = J+1  
      50      CONTINUE  
      60      CONTINUE  
      RETURN  
      END
```

```
INCBUX 59  
INCBUX 60  
INCBUX 61  
INCBUX 62  
INCBUX 63  
INCBUX 64
```

1		SUBROUTINE LABLES (S,FA,FB,NE)	2
	C		3
	C	THIS SUBROUTINE LABLES ALL JOBS, REGARDLESS OF WHETHER THE JOB	4
	C	INITIATES OR CONTINUES THE SOLUTION OF A GIVEN PROBLEM.	5
5	C		6
		COMMON RMIN, UT, N, NL, VX(6460), TH(6460), TEMP, DEFECT,	7
		+ OTH, NS, NLS, NA, UVC, UHC, UF	8
		COMMON /ALPHA/ X(6460), ZZX(6460), ZZY(6460), ZZZ(6460),	9
10		+ ENGI(6460), IDENT(6460), IBUA(14600), FA(1001), EEA(1001),	10
		+ EP(1001), EEB(1001), EAB(1001), EEA(1001)	11
		COMMON /ELMT/ ENGPU, ID, EI, EC, J3, DIV, RMAX2, INIB, NI2, ML,	12
		1 IELE, L, LT, IUT	13
		LEVEL 2, X, ZZX, ZZY, ZZZ, ENGI, IDENT, IBUX, EA, EEA, EB, FEB,	14
		+ EAB, EEA	15
15		COMMON /BOX/ MEA, MEA1, NID, ERL, ERLI, XIPS, YIPS, ZIPS, SL, IP, NR,	16
		1 RESET, CONT, LFINST, XEA, YEA, ZEA, ADAMP, IC, IAFIX, FTAPE	17
		COMMON /SWTCH/ TEMSET, DEFINT, DEFCLS, LINDEF, PTDEF, ALLOY, NSTAPE,	18
		+ LPRNT	19
20		LOGICAL CUNT, TEMSET, FTAPE, DEFINT, DEFCLS, LINDEF, PTDEF, DEFECT	20
		LOGICAL ALLOY, NSTAPE, LPRNT	21
	C		22
		DIMENSION OPT(20), IFMT(20), IFM11(20)	23
		DATA OPT /10HA PKA ,10H ,10HA POINT DE,10HFECT ,	24
25		+ ,10H ,10HFECTS ,10HA DISLOCAT,10HION ,	25
		+ ,10HA DEFECT C,10HCLUSTER ,10HA DEFECT I,10HINTERACTION, ,	26
		+ ,10HUNDENED ,10HDISORDERED,10HBEGINS ,10HCONTINUES ,	27
		+ ,10H ,10H ,10H ,10H /	28
	C		29
30		READ(5,1) OPT(17)	30
		READ(5,1) OPT(18)	31
		WRITE(6,1010)	32
		J1=20	33
		IF(DEFECT) GU TO 20	34
		IF(NE.GT.1) GU TO 15	35
35		I1 = 1	36
		GO TO 20	37
		15 ENCODE(10,11,OPT(2)) NE	38
		I1 = 2	39
40		20 IF(.NOT.PIDEF) GO TO 25	40
		IF(NE.GT.1) GU TO 16	41
		I1 = 3 \$ J1 = 4	42
		GO TO 25	43
		16 ENCODE(10,12,OPT(5)) NE	44
		I1 = 5 \$ J1 = 6	45
45		25 IF(.NOT.LINDEF) GU TO 30	46
		I1 = 7 \$ J1 = 8	47
		30 IF(.NOT.DEFCLS) GU TO 35	48
		I1 = 9 \$ J1 = 10	49
50		35 IF(.NOT.DEFINT) GU TO 40	50
		I1 = 11 \$ J1 = 12	51
		40 I4 = 15	52
		IF(CONT) I4 = 16	53
		I5 = 13	54
		IF(S.NE.1.) I3 = 14	55
55		I2 = 17	56
		I5 = 18	57
		IF(FTAPE) GO TO 100	58

		IF (NSTAPE) GO TO 100	LABEL	51
	C	*****	LABEL	52
60	C	INITIAL CONDITIONS ARE GENERATED	LABEL	53
	C	*****	LABEL	54
	C		LABEL	55
		IF (ALLOY) GO TO 50	LABEL	56
	C		LABEL	57
65	C	IDENTIFY CALCULATION FOR A PURE METAL	LABEL	58
	C		LABEL	59
		IF (.NOT. TEMSET) ENCODE(200,1025,IFMT(1)) OPT(I4),OPT(I1),OPT(J1),	LABEL	60
		1 OPT(I2)	LABEL	61
		IF (TEMSET) ENCODE(200,1030,IFMT(1)) OPT(I2),TEMP	LABEL	62
70		CALL SQUEZ (IFMT)	LABEL	63
	C	*****	LABEL	64
		WRITE(6,1000) (IFMT(I),I=1,14)	LABEL	65
		GO TO 200	LABEL	66
	C		LABEL	67
75	C	IDENTIFY CALCULATION FOR A BINARY ALLOY	LABEL	68
	C		LABEL	69
		50 IF (.NOT. TEMSET) ENCODE(200,1035,IFMT(1)) OPT(I4),OPT(I1),OPT(J1),	LABEL	70
		1 OPT(I3),5	LABEL	71
		IF (.NOT. TEMSET) ENCODE(200,1035,IFMT(1)) OPT(I2),FA,FB	LABEL	72
80		IF (TEMSET) ENCODE(200,1040,IFMT(1)) OPT(I3),8,OPT(I2),FA,FB	LABEL	73
		IF (TEMSET) ENCODE(200,1041,IFMT(1)) TEMP	LABEL	74
		CALL SQUEZ (IFMT) 3 CALL SQUEZ (IFMT)	LABEL	75
	C	*****	LABEL	76
		WRITE(6,1000) (IFMT(I),I=1,14)	LABEL	77
85		WRITE(6,1005) (IFMT(1),I=1,12)	LABEL	78
		GO TO 200	LABEL	79
	C		LABEL	80
	C	*****	LABEL	81
	C	INITIAL LATTICE CONDITIONS ARE READ FROM TAPE	LABEL	82
90	C	*****	LABEL	83
	C		LABEL	84
		100 IF (ALLOY) GO TO 150	LABEL	85
	C		LABEL	86
95	C	CALCULATION FOR A PURE METAL	LABEL	87
	C		LABEL	88
		IF (.NOT. (CONT. AND TEMSET)) GO TO 104	LABEL	89
		ENCODE(200,1045,IFMT(1)) OPT(I4),OPT(I2),TEMP	LABEL	90
		CALL SQUEZ (IFMT)	LABEL	91
	C	*****	LABEL	92
100		WRITE(6,1000) (IFMT(I),I=1,14)	LABEL	93
		GO TO 200	LABEL	94
	C		LABEL	95
		104 IF (TEMP.NE.0.0) GO TO 105	LABEL	96
		ENCODE(200,1050,IFMT(1)) OPT(I4),OPT(I1),OPT(J1),OPT(I5),OPT(I2)	LABEL	97
105		GO TO 110	LABEL	98
		105 ENCODE(200,1055,IFMT(1)) OPT(I4),OPT(I1),OPT(J1),OPT(I5),OPT(I2)	LABEL	99
		ENCODE(200,1041,IFMT(1)) TEMP	LABEL	100
		CALL SQUEZ (IFMT)	LABEL	101
	C	*****	LABEL	102
110		110 CALL SQUEZ (IFMT)	LABEL	103
	C	*****	LABEL	104
		WRITE(6,1000) (IFMT(I),I=1,14)	LABEL	105
		IF (TEMP.NE.0.0) WRITE(6,1005) (IFMT(1),I=1,12)	LABEL	106
		GO TO 200	LABEL	107

```

115      C
      C      CALCULATION FOR A BINARY ALLOY
      C
      150 IF(.NOT.(CONT.AND.TENSET)) GO TO 154
      ENCODE(200,1060,IFMT(1)) OPT(14),OPT(I1),S,OPT(12)
120      ENCODE(200,1061,IFMT(1)) FA,FB,TEMP
      CALL SQUEZ (IFMT) 3 CALL SQUEZ (IFMT)
      C
      *****
      WRITE(6,1000) (IFMT(I),I=1,14)
      WRITE(6,1005) (IFMT(I),I=1,12)
125      C
      154 IF(TEMP.NE.0.0) GO TO 155
      ENCODE(200,1065,IFMT(1)) OPT(14),OPT(I1),OPT(J1),OPT(15),OPT(13),
      1 S
      ENCODE(200,1066,IFMT(1)) OPT(12),FA,FB
130      GO TO 160
      155 ENCODE(200,1070,IFMT(1)) OPT(14),OPT(I1),OPT(J1),OPT(15),OPT(13),
      1 S
      ENCODE(200,1071,IFMT(1)) OPT(12),FA,FB,TEMP
135      C
      160 CALL SQUEZ (IFMT) 3 CALL SQUEZ (IFMT)
      *****
      WRITE(6,1000) (IFMT(I),I=1,14)
      WRITE(6,1005) (IFMT(I),I=1,12)
      200 WRITE(6,1010)
      RETURN
140      1 FORMAT(A10)
      11 FORMAT(A2,' PKA:')
      12 FORMAT(A2,' POINT D')
      1000 FORMAT(13A10,A5)
      1005 FORMAT(10X,12A10)
      1010 FORMAT (///)
145      1025 FORMAT('THIS CALCULATION ',A10,' ANALYSIS OF ',2A10,' IN AN INITI
      1ALLY IDEAL, STATIC, ',A10,' LATTICE.')
      1030 FORMAT('THIS CALCULATION INITIATES FORMATION OF AN IDEAL ',A10,'
      1LATTICE AT A TEMPERATURE OF ',F7.2,' DEGREES KELVIN.')
150      1035 FORMAT('THIS CALCULATION ',A10,' ANALYSIS OF ',2A10,' IN AN INITI
      1ALLY ',A10,' (S=',F7.3,'), STATIC, L1S2 (I.E., FCC-A3B TYPE),')
      1036 FORMAT(A10,' LATTICE (FA=',F7.3,', FB=',F7.3,').')
      1040 FORMAT('THIS CALCULATION INITIATES FORMATION OF AN ',A10,' (S=',F
      17.3,'), L1S2 (I.E., FCC-A3B TYPE), ',A10,' LATTICE (FA=',F7.3,',FB=
155      2',F7.3,').')
      1041 FORMAT('AT A TEMPERATURE OF ',F7.2,' DEGREES KELVIN.')
      1045 FORMAT('THIS CALCULATION ',A10,' FORMATION OF AN IDEAL ',A10,'L
      1LATTICE AT A TEMPERATURE OF ',F7.2,' DEGREES KELVIN.')
160      1050 FORMAT('THIS CALCULATION ',A10,' ANALYSIS OF ',2A10,' IN AN INITI
      1ALLY ',A10,' STATIC ',A10,' LATTICE.')
      1055 FORMAT('THIS CALCULATION ',A10,' ANALYSIS OF ',2A10,' IN AN INITI
      1ALLY ',A10,' ',A10,' LATTICE')
      1060 FORMAT('THIS CALCULATION ',A10,' FORMATION OF AN ',A10,' (S=',F
      17.3,'), L1S2 (I.E., FCC-A3B TYPE), ',A10,' LATTICE')
165      1061 FORMAT('FA=',F7.3,', FB=',F7.3,') AT A TEMPERATURE OF ',F7.2,' DE
      1GREES KELVIN.')
      1065 FORMAT('THIS CALCULATION ',A10,' ANALYSIS OF ',2A10,' IN AN INITI
      1ALLY ',A10,' AND ',A10,' (S=',F7.3,'), STATIC,')
      1066 FORMAT('L1S2 (I.E., FCC-A3B TYPE), ',A10,' LATTICE (FA=',F7.3,', F
      1R=',F7.3,').')
170      1070 FORMAT('THIS CALCULATION ',A10,' ANALYSIS OF ',2A10,' IN AN INITI

```


SUBROUTINE LABE

76/76

OPT=0 TRACE

FTN 4.5*414

11/13/79 14.17.08

PAGE

4

175 IALLY 'A10,' AND 'A10,' (S='F7.3,') LISP')
1071 FORMAT('I.E., FCC-A38 TYPE), 'A10,' LATTICE (FA='F7.3,', FR='F7.3,') AT A TEMPERATURE OF 'F7.2,' DEGREES KELVIN,')
END

LABLE 165
LABLE 166
LABLE 167
LABLE 168

1		SUBROUTINE MNILAT	MNILAT	2
	C		MNILAT	3
	C	THIS SUBROUTINE GENERATES A CYLINDRICAL OR SPHERICAL MINI-LATTICE	MNILAT	4
	C	OF RADIUS ERL1, CENTERED ON A LINE OR POINT MIDWAY BETWEEN THE	MNILAT	5
5	C	ORIGIN AND THE COORDINATES OF THE MOST ENERGETIC ATOM.	MNILAT	6
	C		MNILAT	7
	C		MNILAT	8
		COMMON RMIN, DT, N, NL, VX(6460), TB(6460), TEMP, DEFECT,	COM1	2
		+ UTH, NS, NLS, NA, UVC, UHC, UP	COM1	3
10		COMMON /ALPHA/ (6460), ZZ(6460), ZZY(6460), ZZZ(6460),	COM1	4
		+ ENGP1(6460), IDENT(6460), IBUX(14600), EA(1001), EEA(1001),	COM1	5
		+ ER(1001), EER(1001), EAB(1001), EEA(1001)	COM1	6
		COMMON /ELMT/ ENGPU, ID, EI, EC, J3, DIV, RMAX2, INID, N12, ML,	COM1	7
		+ IELE, L, LT, IOT	COM1	8
15		LEVEL 2, A, ZZX, ZZY, ZZZ, ENGP1, IDENT, IBUX, EA, EEA, EB, FEB,	COM1	9
		+ EAB, EEA	COM1	10
		COMMON /IBUX/ MEA, MEA1, NID, ERL, ERL1, XIPS, YIPS, ZIPS, SL, IP, NR	MNILAT	11
		+ , RESET, COUNT, LFIRST, XEA, YEA, ZEA, ADAMP, IC, IAFIX, FTAPE	MNILAT	12
		LOGICAL DEFECT, ADAMP, IAFIX, FTAPE, LFIRST, CONT	MNILAT	13
20	C		MNILAT	14
		XTRAN(A,H,C) = .424264*A+.346411*B+.244943*C	MNILAT	15
		YTRAN(A,B,C) = -.424264*A+.346411*B+.244943*C	MNILAT	16
		ZTRAN(A,B) = -.346411*A+.484880*B	MNILAT	17
25	C		MNILAT	18
		DM1 = DM2 = DM3 = 0.0	MNILAT	19
	C		MNILAT	20
		CALL SECOND (AT)	MNILAT	21
		IF(CONT.AND.LFIRST) GO TO 20	MNILAT	22
30		A = X(MEA)	MNILAT	23
		CALL UNPAC3 (A,XEA,YEA,ZEA)	MNILAT	24
		XEA = .5*(XEA+XIPS)	MNILAT	25
		YEA = .5*(YEA+YIPS)	MNILAT	26
		ZEA = .5*(ZEA+ZIPS)	MNILAT	27
35	20	NA = 0	MNILAT	28
		LFIRST = .FALSE.	MNILAT	29
		DXE = XEA-XIPS	MNILAT	30
		DYE = YEA-YIPS	MNILAT	31
		DZE = ZEA-ZIPS	MNILAT	32
40		YEA2 = YTRAN(DXE,DYE,DZE)	MNILAT	33
		ZEA2 = ZTRAN(DYE,DZE)	MNILAT	34
		NID = 0	MNILAT	35
		ERL = ERL1	MNILAT	36
		NE = NLS	MNILAT	37
45		NR = NS	MNILAT	38
		IF(DEFECT) XEA = XIPS	MNILAT	39
		IF(DEFECT) YEA = YIPS	MNILAT	40
		IF(DEFECT) ZEA = ZIPS	MNILAT	41
		REWINO 3	MNILAT	42
50		DO 190 IS=1,4	MNILAT	43
		IF (MS.EQ.1.AND.IS.EQ.3.AND.RESET.NE.3.) GO TO 200	MNILAT	44
		IF (MS.EQ.1.AND.IS.EQ.3.AND.RESET.EQ.3.) GO TO 180	MNILAT	45
		DO 170 I=1,NR	MNILAT	46
		IF (IS.EQ.1) WRITE (3) X(I),Y(I),TB(I),IDENT(I),ENGP(I)	MNILAT	47
		IF (IS.GT.1.AND.IS.LT.4) GO TO 30	MNILAT	48
55		A = X(I)	MNILAT	49
		CALL UNPAC3 (A,X11,Y11,Z11)	MNILAT	50
		GO TO 40	MNILAT	51

5-63

```

30      READ (3) U1,D2,D3,IDL,DL
      CALL UNPAC3 (U1,X11,Y11,Z11)
60      C
      C SL=0 FOR CYLINDRICAL <100>
      C SL=1 FOR CYLINDRICAL X-AXIS <110>
      C SL=2 FOR CYLINDRICAL Y-AXIS <111>
      C SL=3 FOR CYLINDRICAL Z-AXIS <11<
      C SL=4 FOR SPHERICAL <A,Y,Z>
      C SL=5 ENTIRE CRYSTAL LATTICE IS USED
      C
40      IF (SL.EQ.4) GO TO 80
      IF (SL.EQ.3) GO TO 70
70      IF (SL.EQ.2) GO TO 60
      IF (SL.EQ.1) GO TO 50
      DX1 = X11-XIPS
      DY1 = Y11-YIPS
      DZ1 = Z11-ZIPS
75      Y12 = YTRAN(DX1,DY1,DZ1)
      Z12 = ZTRAN(DY1,DZ1)
      XYZ = (Y12-YEA2)**2+(Z12-ZEA2)**2
      XYZ = XYZ/.36
      GO TO 90
80      XYZ = (Y11-YEA)**2+(Z11-ZEA)**2
      GO TO 90
60      XYZ = (X11-XEA)**2+(Z11-ZEA)**2
      GO TO 90
70      XYZ = (X11-XEA)**2+(Y11-YEA)**2
      GO TO 90
85      XYZ = (X11-XEA)**2+(Y11-YEA)**2+(Z11-ZEA)**2
90      XYZ = SQRT(XYZ)
      IF (IS.LT.4) GO TO 110
      IF (XYZ.GT.ERLF) GO TO 170
90      II = IDENT(I)
      CALL MTBIN (II,K,L,M,J)
      IF (H.EQ.45) M = 45
      IF (XYZ.GT.ERLE) GO TO 100
      IF (H.EQ.45) M = 47
95      100 IF (H.NE.45.OH.M.NE.97) GO TO 170
      CALL FILBIN (II,K,L,M,J)
      IDENT(II) = II
      GO TO 170
100      110 IF (XYZ.GT.ERL) GO TO 160
      NID = NID+1
      IF (NID.NE.1) GO TO 120
      ERL=ERL+(SQRT(NMAX2)+.005)
      MS = I
      IF (RESET.NE.3) GO TO 120
105      ERLF = ERL
      ERLE=ERL-(SQRT(NMAX2)+.005)*.5
      ERL=ERL+(SQRT(NMAX2)+.005)
120      IF (IS.EQ.1) GO TO 150
      IF (IS.EQ.3) GO TO 130
110      IF (I.LT.MS) GO TO 170
      IF (I.GT.MS) GO TO 130
      NN = NID-1
      NID = 1
130      II = 10L

```

```

MINILAT 51
MINILAT 52
MINILAT 53
MINILAT 54
MINILAT 55
MINILAT 56
MINILAT 57
MINILAT 58
MINILAT 59
MINILAT 60
MINILAT 61
MINILAT 62
MINILAT 63
MINILAT 64
MINILAT 65
MINILAT 66
MINILAT 67
MINILAT 68
MINILAT 69
MINILAT 70
MINILAT 71
MINILAT 72
MINILAT 73
MINILAT 74
MINILAT 75
MINILAT 76
MINILAT 77
MINILAT 78
MINILAT 79
MINILAT 80
MINILAT 81
MINILAT 82
MINILAT 83
MINILAT 84
MINILAT 85
MINILAT 86
MINILAT 87
MINILAT 88
MINILAT 89
MINILAT 90
MINILAT 91
MINILAT 92
MINILAT 93
MINILAT 94
MINILAT 95
MINILAT 96
MINILAT 97
MINILAT 98
MINILAT 99
MINILAT 100
MINILAT 101
MINILAT 102
MINILAT 103
MINILAT 104
MINILAT 105
MINILAT 106
MINILAT 107

```

5-64

115		IF (XYZ.L1.ERL1) GO TO 140	MINILAT	108
		CALL MTBIN (I1,K,L,M,J)	MINILAT	109
		IF ((M.GT.9.AND.M.L1.90).OR.M.GT.100) M = M + 3000	MINILAT	110
		IF (M.LT.10) M = 95	MINILAT	111
		CALL FILBIN (I1,K,L,M,J)	MINILAT	112
120	140	IDENT(NID) = 11	MINILAT	113
		X(NID) = 01	MINILAT	114
		VX(NID) = 02	MINILAT	115
		TB(NID) = 03	MINILAT	116
		ENGPI(NID) = 04	MINILAT	117
125		IF (I.EQ.1P.AND.IS.EQ.2) IP = NID	MINILAT	118
		IF (I.EQ.10V.AND.IS.EQ.2) 10V = NID	MINILAT	119
		GO TO 170	MINILAT	120
	150	IF (XYZ.L1.ERL1.AND.I.LE.NE)>NNL = NID	MINILAT	121
		IF (XYZ.L1.ERL1) NT = I+1	MINILAT	122
130		GO TO 170	MINILAT	123
	160	IF (IS.EQ.1) GO TO 170	MINILAT	124
		IF (IS.EQ.2.AND.I.L1.M9) GO TO 170	MINILAT	125
		NT = NT+1	MINILAT	126
		IDENT(NT) = 10L	MINILAT	127
135		X(NT) = 01	MINILAT	128
		VX(NT) = 02	MINILAT	129
		TB(NT) = 03	MINILAT	130
		ENGPI(NT) = 04	MINILAT	131
	170	CONTINUE	MINILAT	132
140		REWIND 3	MINILAT	133
		IF (IS.EQ.1) NK = NT-1	MINILAT	134
		IF (IS.EQ.2) NK = M9-1	MINILAT	135
	180	IF (IS.EQ.3) NK = NN	MINILAT	136
	190	CONTINUE	MINILAT	137
145	200	N = NN	MINILAT	138
		NL =>NNL	MINILAT	139
		N12 =>NNL+1	MINILAT	140
		MEA1 = 0	MINILAT	141
		IF (IAFIX) GO TO 210	MINILAT	142
150		NL = N	MINILAT	143
		N12 = N+1	MINILAT	144
	210	DO 220 I=1,N	MINILAT	145
		II = IDENT(I)	MINILAT	146
		CALL MTBIN (II,K,L,M,J)	MINILAT	147
155		IF (M.LT.10) NA = NA+1	MINILAT	148
		IF (.NOT.(DEFECT.AND.ADAHP).OR.!EMP.NE.0.0) GO TO 220	MINILAT	149
		IF (M.LT.10) M = 91	MINILAT	150
		CALL FILBIN (II,K,L,M,J)	MINILAT	151
		IDENT(II) = II	MINILAT	152
160	220	CONTINUE	MINILAT	153
	C	CENTER OF THE MINI-LATTICE AND CENTER OF THE GENERALIZED	MINILAT	154
	C	COORDINATE SYSTEM, IN COMENI'S COORDINATE SYSTEM	MINILAT	155
		WRITE (6,230)	MINILAT	156
		WRITE (6,240) XEA,YEA,ZEA,XIPS,YIPS,ZIPS	MINILAT	157
165		DX = XEA-XIPS	MINILAT	158
		DY = YEA-YIPS	MINILAT	159
		DZ = ZEA-ZIPS	MINILAT	160
		XEA1 = XTAN(DX,DY,DZ)	MINILAT	161
		YEA1 = YTAN(DX,DY,DZ)	MINILAT	162
170		ZEA1 = ZTAN(DY,DZ)	MINILAT	163
	C	CENTER OF THE MINI-LATTICE IN THE GENERALIZED COORDINATE SYSTEM	MINILAT	164

	WRITE (6,250) XE41,YE41,ZE41,UN1,UN2,UN3	MINILAT	165
	WRITE (6,260) MEA,M,NL,NA,DIN	MINILAT	166
	CALL SECOND (S1)	MINILAT	167
175	TIME = DT=41	MINILAT	168
	WRITE (6,270) TIME	MINILAT	169
	RETURN	MINILAT	170
	C	MINILAT	171
	C	MINILAT	172
180	C	MINILAT	173
	230 FORMAT (////147,'MINI=LATTICE DATA'/T47,17(1=1)//T41,'CENTER OF',	MINILAT	174
	+T70,'CENTER OF'/T40,'MINI=LATTICE',T69,'COORDINATES'/T37,'X',T45,	MINILAT	175
	+T71,T55,'Z',T66,'Y',T74,'Y',T66,'Z'/T31,T23(1=1),T60,T23(1=1)//	MINILAT	176
	240 FORMAT (1'GENERAL COORDINATE SYSTEM',I30,3F8.3,5X,3F8.3)	MINILAT	177
185	250 FORMAT (1'GENERALIZED COORDINATE SYSTEM',I30,3F8.3,5X,3F8.3)	MINILAT	178
	260 FORMAT (1010) OF MOST ENERGETIC ATOM',T45,I10/1' NUMBER OF ATOMS IN	MINILAT	179
	+MINI=LATTICE',T45,I10/1' NUMBER OF ATOMS WHOSE EQUATIONS',5X,10F10	MINILAT	180
	+110 ARE SOLVED EACH TIME STEP',T45,I10/1' NUMBER OF ATOMS WITH BOU	MINILAT	181
	+BOUNDARY TYPE',5X,1'LESS THAN 10',I45,I10/1' LENGTH OF NEW TIME STEP',T	MINILAT	182
190	+45,F10.5////)	MINILAT	183
	270 FORMAT (10TIME TO GENERATE MINI=LATTICE IS',F10.5,' SEC.')	MINILAT	184
	END	MINILAT	185

1		SUBROUTINE PNT1 (N,P0,F1,F2)	PNT1	2
	C		PNT1	3
	C	MULTIPLY POTENTIAL	PNT1	4
	C		PNT1	5
5		COMMON /PNTU/ W,KE,CLP(12),KIN(12),KMX(12),SK,AP,MP,UOCC,ZA,ZB,	PNT1	6
		+ KMAX	PNT1	7
	C		PNT1	8
		ZZ=ZA*ZB	PNT1	9
		SC = 51/UOCC	PNT1	10
10		CC = 14.599465*ZZ/UOCC	PNT1	11
		S1 = (EXP(-.3*W/SC))	PNT1	12
		S2 = (EXP(-1.2*W/SC))	PNT1	13
		S3 = (EXP(-6.0*W/SC))	PNT1	14
		S51 = .35*S1+.55*S2+.10*S3	PNT1	15
15		S52 = .1475/SC*S1+.55/SC*S2+.50/SC*S3	PNT1	16
		S53 = .0315/SC*S1*S2+.792/SC*S2*S3+3.6/SC*S2*S3	PNT1	17
		F0 = CC/W*S51	PNT1	18
		F1 = (-CC/W**2)*S51-(CC/W)*S52	PNT1	19
		F2 = (2.*CC/W**3)*S51+(2.*CC/W**2)*S52+(CC/W)*S53	PNT1	20
20		RETURN	PNT1	21
		END	PNT1	22

1		SUBROUTINE P012 (N,P0,F1)	P012	2
	C		P012	3
	C	ENGLERT POTENTIAL	P012	4
	C		P012	5
5		COMMON /P012/ N1,LE,CLP(12),KHN(12),KHX(12),SW,AP,KP,UDCC,ZA,ZR,	P012	6
		+ KMAX	P012	7
		COMMON /P012/ AK(7,10),BK(7,10),CK(7,10),OK(7,10)	P012	8
	C		P012	9
		K = 1	P012	10
10		F1 = 0.0	P012	11
		F0 = 0.0	P012	12
		IF (N.LE.KK(1,1)) GO TO 30	P012	13
	20	IF (N.GT.KK(1,K).AND.N.LE.KK(1,K+1)) GO TO 30	P012	14
		IF (N.GT.KK(1,10)) GO TO 40	P012	15
15		IF (N.LE.7) K = K+1	P012	16
		GO TO 20	P012	17
	30	S01 = AK(1,K)*(N-KK(1,K))**3	P012	18
		S02 = BK(1,K)*(N-KK(1,K))**2	P012	19
		S03 = CK(1,K)*(N-KK(1,K))	P012	20
20		F0 = S01+S02+S03+OK(1,K)	P012	21
		F1 = 3.*AK(1,K)*(N-KK(1,K))**2+2.*BK(1,K)*(N-KK(1,K))+CK(1,K)	P012	22
	40	RETURN	P012	23
		END	P012	24

1		SUBROUTINE F013 (W,F0,F1)	P013	2
	C		P013	3
	C	MIXED HOLLER AND ENGLERT POTENTIAL WITH SPLICE FUNCTION	P013	4
	C		P013	5
5		COMMON /P013/ KA(7,10),AK(7,10),BK(7,10),CK(7,10),DK(7,10)	P013	6
		COMMON /P014/ UG,DH,DI,DJ,PK,OL	P013	7
		COMMON /P015/ K1,KE,CLP(12),RM(12),RMX(12),SK,AP,RP,UDCC,ZA,ZR,	P013	8
		+ KMAX	P013	9
	C		P013	10
10		IF (W.LE,KH/UDCC) GO TO 20	P013	11
		IF (V.GE,KE/UDCC) GO TO 30	P013	12
	C		P013	13
	C	CALCULATION FOR THE SPLICE REGION BETWEEN KM AND KE	P013	14
		KA = K-KK(1,3)	P013	15
15		FU = UG*DH**5+DI*DH**4+DI*DH**3+DI*DH**2+DKK*DH+DL	P013	16
		FJ = UG*DH**4+4*DH*DK**3+3*DI*DH**2+2*DJ*DH+DKK	P013	17
		RETURN	P013	18
	20	CALL P011 (W,F0,F1,UDM)	P013	19
		RETURN	P013	20
20	30	CALL P012 (W,F0,F1)	P013	21
		RETURN	P013	22
		END	P013	23

1		SUBROUTINE POTS (I,FC,F1)	POTS	2
	C		POTS	3
	C	JONKSON INTERACTION POTENTIAL FOR CU	POTS	4
	C	JOURNAL OF APPLIED PHYSICS, MARCH 1974.	POTS	5
5	C		POTS	6
	C	COMMON /POTS/ R0,RC,CLP(12),RNN(12),RMX(12),SK,AP,RP,UOCC,ZA,ZR,	POTS	7
		+ K04A	POTS	8
	C		POTS	9
		C1 = 011.17172002 = 1142.62204*UOCC	POTS	10
10		C5 = 054.0069*UOCC**2+04 = 314.05000*UOCC**3	POTS	11
		C5 = 59.6702*UOCC**4+06 = 4.44795*UOCC**5	POTS	12
		F1 = C1+C2**+C3**+2=C4**+3+C5**+4=C6**+5	POTS	13
		F1 = C2+2.*C3**+3.*C4**+2+4.*C5**+3=5.*C6**+4	POTS	14
		RETURN	POTS	15
15		END	POTS	16

SUBROUTINE PNT6

7077, OPTNO TRAIL

FIN 4,5+014

11/13/79 14.17.08

PAGE 1

1		SUBROUTINE PNT6 (N,FO,F1)	PNT6	2
	C		PNT6	3
	C	THIS POTENTIAL IS A MIXED SURF-WATER (WELDY 1.7 ANG.) AND	PNT6	4
	C	UNDERWATER (WELDY 1.7 ANG.) POTENTIAL FOR LO	PNT6	5
5	C		PNT6	6
	C	COMMON /PNT6/ F1, F2, CLP(12), RND(12), KHX(12), SH, AP, RP, UOCC, ZA, ZH,	PNT6	7
		* RZA	PNT6	8
	C		PNT6	9
		IF (N.LT.1.7/UOCC) CALL PNT4 (N,FO,F1)	PNT6	10
10		IF (N.GE.1.7/UOCC) CALL PNT5 (N,FO,F1)	PNT6	11
		RETURN	PNT6	12
		END	PNT6	13

1		SUBROUTINE POT7 (W,F0,F1,INPUT)	POT7	2
	C		POT7	3
	C	READS OUT THE CALCULATED POTENTIALS FOR CU, AU, CH=AU, NI, AL, NI=AL	POT7	4
	C		POT7	5
5		COMMON /POTD/ RI,RE,CLP(12),RMM(12),RMX(12),SK,AP,RP,UDCC,ZA,ZR,	POT7	6
		* N142	POT7	7
		COMMON /POTL/ KA(7,10),AK(7,10),BA(7,10),CK(7,10),DK(7,10)	POT7	8
		DIFCO(10),IT(6)	POT7	9
		DATA LMI /4*9.0,9/	POT7	10
10	C		POT7	11
		KA = INPUT*5	POT7	12
		KA = LMI*(KA+1)	POT7	13
		K = 1	POT7	14
		F0 = 0.0	POT7	15
15		F1 = 0.0	POT7	16
		IF (N.LT.KK(KK,1)) GO TO 50	POT7	17
20		IF (1.61.KK(KK,K).AND.W.LE.KK(KK,K+1)) GO TO 30	POT7	18
		IF (1.61.KK(KK,K)) GO TO 40	POT7	19
		IF (K.LE.1X-1) K = K+1	POT7	20
20		GO TO 20	POT7	21
	30	SS1 = AK(KK,K)*(W-KK(KK,K))*A5	POT7	22
		SS2 = BK(KK,K)*(W-KK(KK,K))*A2	POT7	23
		SS3 = CA(KK,K)*(W-KK(KK,K))	POT7	24
		F0 = SS1+SS2+SS3+DK(KK,K)	POT7	25
25		F1 = 5.*AK(KK,K)*(W-KK(KK,K))*A2+2.*BK(KK,K)*(W-KK(KK,K))+CK(KK,K)	POT7	26
	40	RETURN	POT7	27
	50	CALL POT1 (W,F0,F1,F2)	POT7	28
		RETURN	POT7	29
		END	POT7	30

```

1      SUBROUTINE SPEFORS (JUDCC)
2
3      THIS SUBROUTINE CALCULATES UNIQUE SPECIAL (PRESSURE) FORCES AND
4      FORCE CONSTANTS (Hookeian AND VISCOS) FOR THE POTENTIAL OFING
5      USED. THOSE FORCES ARE APPLIED AT THE LATTICE CONTINUUM INTERFACE.
6      THEY ARE INTENDED TO MAKE THE FINITE LATTICE APPEAR INFINITE FOR
7      BOTH 'STATIC' AND 'DYNAMIC' CALCULATIONS.
8
9      COMMON /RMT/ UT, N, NL, VX(6460), Y3(6460), TEMP, DEFECT,
10      * UT4, * S, * NL5, * A, * UVC, * UHC, * U
11      COMMON /ALPHA/ X(6460), ZZX(6460), ZZY(6460), ZZZ(6460),
12      * ENGPI(6460), IJX(16460), IJY(16460), IJZ(16460), EA(1001), EEA(1001),
13      * F(1001), EEL(1001), EAB(1001), EAB(1001)
14      COMMON /VEL IT/ EIGMU, ID, EI, EF, J3, DIV, RMAX2, INIB, N12, NL,
15      * IFLF, L, LI, IO)
16      LEVEL 2, A, ZZX, ZZY, ZZZ, ENGPI, IJENT, IJUX, EA, EEA, EB, FEH,
17      * EAB, FEAB
18      COMMON /BOX/ IK(17,125)
19      COMMON /FORCE/ AX(25,9), AY(25,9), AZ(25,9), SFM
20      COMMON /CENTR/ X11,Y11,Z11,AX1,YY1,ZZ1,EXTRA(11)
21      COMMON /BBOX/ MEA,MEAL,NID,EEL,EELI,XIPS,YIPS,ZIPS,SL,IP,NR,
22      * REBEL,CONT,LFIRST,XEA,YEA,ZEA,ADAMP,IC,IAFIX,FTAPE
23      COMMON /SPEFORS/ * I, IUC, ID1, IJ2, IJ3, IJ4, IJ5, IJ6, NMF, ST, HXC, MYC, HZC,
24      * VCA, VCY, VCL, LPMNT2, NB
25      EQUIVALENCE (VN,UVC), (HN,UHC), (IP,NF)
26      DIMENSION H(25)
27      LOGICAL LPMNT2,LL151,IAFIX,FTAPE,CONT,LFIRST,ADAMP
28
29      BUANU(14,1,C) = IFIX(A=31,)+51*IFIX(J=32,)+961*IFIX(C=32,.)
30
31      NL = 0
32      STEP = ST/JUDCC
33
34      DO 20 I=1,NH
35      IJUX(I) = 0
36
37      I4=PI-1
38      I5 = 0
39
40      DO 30 I=1,N
41      ZZX(I) = 0.0
42      ZZY(I) = 0.0
43      ZZZ(I) = 0.0
44      ENGPI(I) = 0.0
45      II = IJENT(I)
46      CALL MTBTH (I1,K,L,M,J)
47      A = X(I)
48      CALL JHP4C3 (A,X11,Y11,Z11)
49      J = IJUX(I)
50      CALL FIL 11 (I1,K,L,M,J)
51      IJENT(I) = II
52      I1 = IJUX(J)
53      CALL IJH 14 (I2,J1,1)
54      IJUX(J) = I2
55      CONTINUE
56
57      RETURN
58      END

```

5-74

		H2 = 0	SPEFORS	51
		SP1 = 10.0	SPEFORS	52
60		PH=10.	SPEFORS	53
	C		SPEFORS	54
		DO 170 I=14,N	SPEFORS	55
		LLIST = .FALSE.	SPEFORS	56
		I = IC	SPEFORS	57
65		IF (IE.EQ.14) I=100	SPEFORS	58
		IF (IE.EQ.100) GO TO 170	SPEFORS	59
		IF (LPH=12.AND.(I.EQ.100.OR.I.EQ.101.OR.I.EQ.102.OR.I.EQ.103.OR.I.	SPEFORS	60
		+EQ.104.OR.I.EQ.105.OR.I.EQ.106)) LLIST = .TRUE.	SPEFORS	61
	C		SPEFORS	62
70	40	I1 = IDENT(I)	SPEFORS	63
		CALL MTBIN (I1,K,L,M,J)	SPEFORS	64
	C	CALCULATE MAJOR AND MINOR INDICES FOR FORCE ARRAY (MX AND MY)	SPEFORS	65
		IF (M.LE.10) GO TO 50	SPEFORS	66
		MX = M/100	SPEFORS	67
75		MY = M-MX*100	SPEFORS	68
		IF (M.GT.100) GO TO 50	SPEFORS	69
		MX = 25	SPEFORS	70
		MY = M/10	SPEFORS	71
80	50	IF (M.LE.10.AND.I.NE.100) GO TO 170	SPEFORS	72
	C		SPEFORS	73
		IF (L.EQ.1.AND.M.EQ.42) GO TO 170	SPEFORS	74
		IF (L.EQ.1) M2 = 1	SPEFORS	75
		A = X(I)	SPEFORS	76
85		CALL UNPAC3 (A,X11,Y11,Z11)	SPEFORS	77
		IT = 1	SPEFORS	78
		I1 = 1	SPEFORS	79
		ID = 1	SPEFORS	80
90	60	XX1 = 0.05YY1 = 0.05ZZ1 = 0.0	SPEFORS	81
		ENGPI(I) = 0.0	SPEFORS	82
	C		SPEFORS	83
		DO 70 J1=1,125	SPEFORS	84
		J2 = J+1K(1,J1)	SPEFORS	85
		IF (J2.LE.0.OR.J2.GT.NB) GO TO 70	SPEFORS	86
95		J3 = 180X(J2)	SPEFORS	87
		CALL NEAF	SPEFORS	88
	70	CONTINUE	SPEFORS	89
	C		SPEFORS	90
		IF (I.NE.100) GO TO 120	SPEFORS	91
100	C		SPEFORS	92
		IF (IT.GT.1) GO TO 60	SPEFORS	93
		PXC = -XX1SPYC = -YY1SPZC = -ZZ1	SPEFORS	94
		IT = 2	SPEFORS	95
		X31 = X11+YS1 = Y11+Z31 = Z11	SPEFORS	96
105		IF (LLIST) WRITE (6,250) I,K,L,M,J,X11,Y11,Z11,PXC,PYC,PZC	SPEFORS	97
		GO TO 90	SPEFORS	98
	C		SPEFORS	99
	80	IF (I1.EQ.2) H(11) = -(XX1+PXC)/(X11-X31)	SPEFORS	100
		IF (I1.EQ.3) H(11) = -(YY1+PYC)/(Y11-Y31)	SPEFORS	101
		IF (I1.EQ.4) H(11) = -(ZZ1+PZC)/(Z11-Z31)	SPEFORS	102
110		STN = ST*11+1.E-10	SPEFORS	103
		XF = XX1*FN5YF = YY1*FN3ZF = ZZ1*FN	SPEFORS	104
		CH = H(I1)*HN	SPEFORS	105
		IF (LLIST.AND.I1.EQ.2.AND.I1.EQ.1) WRITE (6,260)	SPEFORS	106
		IF (LLIST) WRITE (6,240) IT,I1,STN,ENGPI(I),XF,YF,ZF,CH	SPEFORS	107

115		II = II+1	SPEFORS 108
	C		SPEFORS 109
		IF (II.EQ.25F) GO TO 100	SPEFORS 110
	C		SPEFORS 111
	10	IF (II.EQ.2) X11 = X11+STEP	SPEFORS 112
120		IF (II.EQ.3) Y11 = Y11+STEP	SPEFORS 113
		IF (II.EQ.4) Z11 = Z11+STEP	SPEFORS 114
		GO TO 60	SPEFORS 115
	C		SPEFORS 116
	100	X11 = X51+Y11 * Y51+Z11 = Z51	SPEFORS 117
125		II = 1	SPEFORS 118
		HT = 0.0	SPEFORS 119
	C		SPEFORS 120
		DO 110 IP=1,NSP	SPEFORS 121
	110	H1 = HT+H(12)	SPEFORS 122
130	C		SPEFORS 123
		IF (IT.EQ.2) HXC = HT/NSF	SPEFORS 124
		IF (IT.EQ.3) HYC = HT/NSF	SPEFORS 125
		IF (IT.EQ.4) HZC = HT/NSF	SPEFORS 126
		IT = IT+1	SPEFORS 127
135	C		SPEFORS 128
		IF (II.EQ.4) GO TO 90	SPEFORS 129
		F=AMAX1(ABS(PXC),ABS(PYC),ABS(PZC))	SPEFORS 130
		FPO=AMAX1(HXC,MYC,MZC)	SPEFORS 131
		VGX = SJNT(8,*HXC)*VCY = SJNT(8,*MYC)*VCZ = SURT(8,*MZC)	SPEFORS 132
140		XP = PXC*FNSYP = MYC*FNSZP = PZC*FNN	SPEFORS 133
		XH = HXC*FNSYM = MYC*FNSZH = MZC*FNN	SPEFORS 134
		XV = VGX*FNSYV = VCY*FNSZV = VLZ*FNN	SPEFORS 135
		IF (LLIST) WRITE (6,270) 1,L,N,XP,YP,ZP,XH,YH,ZH,XV,YV,ZV	SPEFORS 136
		GO TO 170	SPEFORS 137
145	C		SPEFORS 138
	120	IF (II.GT.1) GO TO 130	SPEFORS 139
		PX = -X11+FY = -Y11+PZ = -Z11	SPEFORS 140
		IT = 2	SPEFORS 141
		X51 = X11+Y51 = Y11+Z51 = Z11	SPEFORS 142
150		IF (LLIST) WRITE (6,280) 1,A,L,H,J,X11,Y11,Z11,PX,PY,PZ	SPEFORS 143
		GO TO 140	SPEFORS 144
	C		SPEFORS 145
	130	IF (IT.EQ.2) H(11) = -(X11+PX)/(X11-X51)	SPEFORS 146
		IF (IT.EQ.3) H(11) = -(Y11+PY)/(Y11-Y51)	SPEFORS 147
155		IF (IT.EQ.4) H(11) = -(Z11+PZ)/(Z11-Z51)	SPEFORS 148
		STH = S1*11*1.E-10	SPEFORS 149
		XF = X11*FNSYP = Y11*FNSZF = Z11*FNN	SPEFORS 150
		CH = H(11)*HNN	SPEFORS 151
		IF (LLIST) AND (II.EQ.2.AND. II.EQ.1) WRITE (6,280)	SPEFORS 152
160		IF (LLIST) WRITE (6,240) IT,11,STH,ENGP1(1),XF,YF,ZF,CH	SPEFORS 153
		II = II+1	SPEFORS 154
		IF (II.GT.25F) GO TO 150	SPEFORS 155
	C		SPEFORS 156
	140	IF (II.EQ.2) X11 = X11+STEP	SPEFORS 157
165		IF (II.EQ.3) Y11 = Y11+STEP	SPEFORS 158
		IF (II.EQ.4) Z11 = Z11+STEP	SPEFORS 159
		GO TO 60	SPEFORS 160
	C		SPEFORS 161
	150	X11 = X51+Y11 * Y51+Z11 = Z51	SPEFORS 162
170		II = 1	SPEFORS 163
		HT = 0.0	SPEFORS 164

	C		SPEFORS	165
		DO 160 I=1,NBF	SPEFORS	166
175	160	NI = NI+1(I2)	SPEFORS	167
	C		SPEFORS	168
		IF (IT,EO,2) HAP = HT/NSF	SPEFORS	169
		IF (IT,EO,3) HYP = HT/NSF	SPEFORS	170
		IF (IT,EO,4) HZP = HT/NSF	SPEFORS	171
		IT = IT+1	SPEFORS	172
180		IF (IT,EO,4) GO TO 140	SPEFORS	173
	C		SPEFORS	174
		XP = PX*FNSYP = PY*FNSZP = PZ*FNP	SPEFORS	175
		HX = HXC-HXP, HY = HYC-HYP, HZ = HZC-HZP	SPEFORS	176
		XH = HX*HNSY, HY = HY*HNSZ, HZ = HZ*HNP	SPEFORS	177
185		HHA = HX/HXC, HMY = HY/HYC, HNZ = HZ/HZC	SPEFORS	178
		IF (LLIST) WRITE (6,240) I,L,M,XP,YP,ZP,XH,YH,ZH,HMX,HMY,HMZ	SPEFORS	179
		IF (HA,LT,0,0) HX = -HX	SPEFORS	180
		CALL PACF (A,PX,HA)	SPEFORS	181
		AX(HX,HH) = A	SPEFORS	182
190		IF (HY,LT,0,0) HY = -HY	SPEFORS	183
		CALL PACF (A,PY,HY)	SPEFORS	184
		AY(HY,HH) = A	SPEFORS	185
		IF (HZ,LT,0,0) HZ = -HZ	SPEFORS	186
		CALL PACF (A,PZ,HZ)	SPEFORS	187
195		AZ(HZ,HH) = A	SPEFORS	188
		IF (HX,NE,0,0) SFM = AMIN1(ABS(HX),SFM)	SPEFORS	189
		IF (ABS(PX),GT,F) PH=AMIN1(PH,ABS(PX))	SPEFORS	190
	C		SPEFORS	191
	C		SPEFORS	192
200	C		SPEFORS	193
	170	CONTINUE	SPEFORS	194
	C		SPEFORS	195
		IF (.NOT.LPRINT2) GO TO 230	SPEFORS	196
		I3 = N-NA	SPEFORS	197
205		WRITE (6,300) I3	SPEFORS	198
		WRITE (6,310) VCX,VCY,VCZ	SPEFORS	199
		PM1=PM*FNP S SFM1=SFM*HN	SPEFORS	200
		WRITE(6,320) PM1,SFM1	SPEFORS	201
		TSTMN=F/FPD	SPEFORS	202
210		WRITE(6,330) TSTMN,F,FPD	SPEFORS	203
		WRITE (6,330)	SPEFORS	204
		DO 220 I=1,N	SPEFORS	205
		A = X(I)	SPEFORS	206
		CALL UNPAL3 (A,X11,Y11,Z11)	SPEFORS	207
215		PX = PY = PZ = XH = HY = HZ = 0.0	SPEFORS	208
		II = IDENT(I)	SPEFORS	209
		CALL MTRIN (II,K,L,M,J)	SPEFORS	210
		IF (M,GT,100) GO TO 160	SPEFORS	211
		MX = 25	SPEFORS	212
220		MM = M/10	SPEFORS	213
		HZ = 1	SPEFORS	214
		IF (M,GT,50,AND,M,LT,70) HZ = (M-20)/10+13	SPEFORS	215
		GO TO 140	SPEFORS	216
	180	MX = M/100	SPEFORS	217
225		MM = M-M*100	SPEFORS	218
		IF (M,LT,1299) M3 = M	SPEFORS	219
		IF (M,GT,1299) M3 = M-1200	SPEFORS	220
		HZ = M/100+1	SPEFORS	221

230 190 IF ('L1,1') GO TO 210
IF ('L1,1') GO TO 200
A = AX(10,10) & CALL UNPACK (A, PX, HX)
A = AY(10,10) & CALL UNPACK (A, PY, HY)
A = AZ(10,10) & CALL UNPACK (A, PZ, HZ)
GO TO 210
230 HX = HX * HZ * PF1
210 WRITE (6,340) I,X,L,M,J,MZ,HX,MH,X11,Y11,Z11,PX,HX,PY,HY,PZ,HZ
220 CONTINUE
230 RETURN
C
240 C
240 FORMAT(2I0,1P412.5,1P415.5)
250 FORMAT ('U',T4,'I',T12,'K',T10,'L',T24,'M',T20,'J',T33,'X11',T43,'Y11',T53,'Z11',T63,'PAC',T73,'MYC',T83,'PZC',T93,'S16,6F10.5')
260 FORMAT ('U',T4,'I',T12,'K',T10,'L',T24,'M',T20,'J',T33,'X11',T43,'Y11',T53,'Z11',T63,'PAC',T73,'MYC',T83,'PZC',T93,'S16,6F10.5')
270 FORMAT ('U',T4,'I',T12,'L',T10,'M',T24,'XP',T36,'YP',T48,'ZP',T59,'XH',T70,'YH',T81,'ZH',T94,'XH',T104,'YH',T114,'ZH',T124,'S16,6F10.5')
280 FORMAT ('U',T4,'I',T12,'K',T10,'L',T24,'M',T30,'J',T53,'X11',T43,'Y11',T53,'Z11',T63,'PAC',T76,'MYC',T86,'PZC',T96,'S16,6F10.5')
290 FORMAT ('U',T4,'I',T12,'L',T10,'M',T24,'XP',T36,'YP',T48,'ZP',T59,'XH',T70,'YH',T81,'ZH',T94,'XH',T104,'YH',T114,'ZH',T124,'S16,6F10.5')
300 FORMAT ('NUMBER OF INTERFACE ATOMS FOR THE ENTIRE LATTICE IS',T80
310 FORMAT ('X=, Y=, AND Z-VISCOUS FORCE CONSTANTS FOR ALL INTERFACE
320 FORMAT ('SMALLEST (EFFECTIVELY) NON-ZERO PRESSURE FORCE AND HOOKI
330 FORMAT ('I',T7,'I',T14,'K',T18,'L',T23,'M',T31,'J',T45,'H',T39,'M
340 FORMAT (2I7,14,16,16,3I4,3F8.2,1P610.3)
350 FORMAT('FOR THE POTENTIAL USED IN THIS STUDY, THE MINIMUM VALUE F
FOR THE VARIABLE TEST IS TWICE',F10.0,' MAXIMUM VALUE OF F IS',F10.
+6,' MAXIMUM VALUE OF H IS',F10.5/)
END

1		SUBROUTINE SQUEZ (IF)	SQUEZ	2
	C		SQUEZ	3
	C	SUBROUTINE SQUEZ IS USED BY LABLE TO SUPPRESS EXTRA BLANKS	SQUEZ	4
	C	FROM THE GENERATED ALPHANUMERIC FORMATS	SQUEZ	5
5	C		SQUEZ	6
	C	DIMENSION IF(20),IF1(200),IF2(200)	SQUEZ	7
	C		SQUEZ	8
		DO 10 I=1,200	SQUEZ	9
		IF1(I) = 1H	SQUEZ	10
10	10	IF2(I) = 1H	SQUEZ	11
		DECODE (200,20,IF1(I)) IF1	SQUEZ	12
	20	FORMAT(2(100A1))	SQUEZ	13
		J=1	SQUEZ	14
		IF2(J)=IF1(I)	SQUEZ	15
15		DO 50 I=P,200	SQUEZ	16
		IF(IF1(I).EQ.1H .AND. IF1(I-1).EQ.1H) GO TO 50	SQUEZ	17
		J=J+1	SQUEZ	18
		IF2(J)=IF1(I)	SQUEZ	19
	30	CONTINUE	SQUEZ	20
20		ENCODE(200,20,IF1(I)) IF2	SQUEZ	21
		RETURN	SQUEZ	22
		END	SQUEZ	23

5-79

```

1      SUBROUTINE THERMAL (GENVS)
C----- IF THE LOGICAL VARIABLE GENVS IS .TRUE. THEN
C      THIS SUBROUTINE PRODUCES A POSITION ARRAY VX(.) OF N TRIPLES
C      (VX(I)=(VXI,VYI,VZI)) SO THAT TO MOVE FROM VX(.) TO X(.) IN DTM
C      SECONDS REQUIRES A MAXWELL VELOCITY DISTRIBUTION.
5      C      FINALLY PERTAINING STATISTICAL DATA IS PRODUCED BY A STATISTICAL
C      ANALYSIS OF THE VELOCITY DISTRIBUTION.
C      IF GENVS IS .FALSE. THEN THE ARRAY VX(.) IS ASSUMED GIVEN AND ONLY
C      THE VELOCITY DISTRIBUTION ANALYSIS IS PERFORMED.
10     C
C----- X(.) IS THE PRESENT POSITION OF ATOM(.)
C      VX(.) IS THE PAST POSITION OF ATOM(.)
C      TB(.) IS THE INITIAL POSITION OF ATOM(.) (FIXED LATTICE SITES -
C      INVARIANT)
15     C
C      LOGICAL GENVS
C      INTEGER NMARK,STAN
C      REAL IGINV
20     C
C      COMMON NMIN, DT, M, NL, VX(6460), TB(6460), TEMP, DEFECT,
C      + DTN, NS, NLS, NA, OVC, UDC, UF
C      COMMON /ALPHA/ X(6460), ZZX(6460), ZZY(6460),
C      + ENGP1(6460), IDENT(6460), IBUX(14600), EA(1001), EEA(1001),
C      + EB(1001), EEB(1001), EAB(1001), EFAB(1001)
25     C      COMMON /ELMT/ ENGP1, ID, EI, EE, J3, DIV, RMAX2, INIB, N12, ML,
C      + JELE, L, LT, IDT
C      LEVEL 2, A, ZZY, ZZY, ZZZ, ENGP1, IDENT, IBUX, EA, EEA, EB, FER,
C      + EAB, EEAU
30     C
C      DIMENSION NUBSRV(41),VHIST(40),LINE(102),THEORY(41),MESSAGE(3,2,2),
C      + CHISQS(40),CHISQ1(40),ICOMP(4),ITLET(2,3)
C
C      DATA PI,STAR,BLANK,INITIL,IVEE,IARR,IALPHA,ISIGMA,IPERCN/
C      + 3.141592654,14*,10H ,10H INITIAL ,1HV,1HH,SHALPHA,
35     C      + 7H3.D. ,7HPC4CENT/
C      DATA (ICOMP(J),J=1,4)/1HX,1MY,1HZ,1H /
C      DATA (LINE(I),I=1,102)/102*1H*/
C      DATA (((MESSAGE(I,J,K),I=1,3),J=1,2),K=1,2)/
C      + 10HDISPLACEME,10HNT, COMPN,10HENT ,
40     C      + 10HVELOCITY, ,10HCOMPONENT ,10H ,
C      + 10HABSOLUTE D,10HISPLACEME,10H1 ,
C      + 10HDEPL ,10H ,10H /
C      DATA ((ITLET(I,J),I=1,2),J=1,3)/
C      + 5HMEAN ,5HCV > ,5HVAR ,5HKE > ,5HMODE ,5HVP /
45     C
C----- SGLVLS,SGLVLI ARE SIGNIFICANCE LEVELS AND CHISQS(.),CHISQ1(.) ARE
C      THE CURRENT/HOING CHISQ2 CRITICAL VALUES USED WITH THE HISTOGRAM
C      DATA SGLVLS,SGLVLI/.05,.01/
C      DATA (CHISQS(I),I=1,40)/
50     C      + 5.34, 5.99, 7.01, 7.49,11.07,12.59,14.07,15.51,16.92,18.31,
C      + 19.66,21.03,22.50,23.68,25.00,26.50,27.59,28.67,30.14,31.41,
C      + 32.67,33.92,35.17,36.42,37.67,38.89,40.11,41.34,42.56,43.77,
C      + 44.97,46.17,47.37,48.57,49.77,50.96,52.16,53.36,54.56,55.76/
55     C      DATA (CHISQ1(I),I=1,40)/
C      + 6.63, 7.21,11.34,13.28,15.09,16.81,18.48,20.09,21.67,23.21,
C      + 24.72,26.22,27.69,29.14,30.58,32.00,33.41,34.81,36.19,37.57,
C      + 38.95,40.29,41.64,42.98,44.31,45.64,46.96,48.28,49.59,50.89,

```

	* 52.17,53.45,54.75,56.01,57.27,58.57,59.85,60.13,62.41,63.69/	THRMAL	51
C		THRMAL	52
60	C--- BLTZIN IS THE GULIZMANN CONSTANT (EV/DEG K)	THRMAL	53
	DATA BLTZIN/.001755E-4/	THRMAL	54
	DATA A/1.5/	THRMAL	55
C		THRMAL	56
C		THRMAL	57
65	TWOPI = 2.*PI	THRMAL	58
	SURTPI = SURT(PI)	THRMAL	59
	SURTI2 = SURT(2.)	THRMAL	60
C		THRMAL	61
70	EN = NA	THRMAL	62
	TEM = TEMP	THRMAL	63
	IF (GENVS) EN = N	THRMAL	64
	IF (GENVS) TEM = 2.*TEM	THRMAL	65
	CONST = 2.*BLTZIN*TEM	THRMAL	66
	VP = SURT(CONST)	THRMAL	67
75	SOCUMV = VP/SURTI2	THRMAL	68
	VPV = VP	THRMAL	69
	CONSTV = CONST	THRMAL	70
	SOCUMV = SOCUMV	THRMAL	71
	IPRE = BLANK	THRMAL	72
80	IF (.NOT.GENVIS) GO TO 70	THRMAL	73
C		THRMAL	74
C---	GENERATE PREVIOUS POSITIONS VX(I)	THRMAL	75
	N = 0	THRMAL	76
	IF (N.LE.1) GO TO 710	THRMAL	77
85	DO 60 I=1,N	THRMAL	78
	AA = X(I)	THRMAL	79
	CALL UNPAL3 (AA,XI,YI,ZI)	THRMAL	80
	IF (TEMP.GT.0.) GO TO 20	THRMAL	81
	VX1 = XI	THRMAL	82
90	VY1 = YI	THRMAL	83
	VZ1 = ZI	THRMAL	84
	GO TO 50	THRMAL	85
20	NN = 0	THRMAL	86
30	XINN = RANF(M)	THRMAL	87
45	V = INGINV(A,XINN)	THRMAL	88
	IF (V.GE.0.) GO TO 40	THRMAL	89
	NN = NN+1	THRMAL	90
	IF (NN.LT.10) GO TO 30	THRMAL	91
	GO TO 710	THRMAL	92
100	40 V = SURT(V*CONST)*UTN	THRMAL	93
C---	NUN CHOOSE COS(SHETA) FROM A UNIFORM(-1,1) DISTRIBUTION AND	THRMAL	94
C	PHI FROM A UNIFORM(0,2*PI) DISTRIBUTION	THRMAL	95
	PHI = TWOPI*RANF(M)	THRMAL	96
	CTHETA = 2.*RANF(M)-1.	THRMAL	97
105	SHETA = AMAX1(1,-CTHETA**2,0.)	THRMAL	98
	SHETA = SQRT(SHETA)	THRMAL	99
	VZ1 = ZI-V*CTHETA	THRMAL	100
	VST = V*SHETA	THRMAL	101
	VX1 = XI-VST*COS(PHI)	THRMAL	102
110	VY1 = YI-VST*SIN(PHI)	THRMAL	103
C		THRMAL	104
50	CALL PAC3 (AA,VX1,VY1,VZ1)	THRMAL	105
	VX(I) = AA	THRMAL	106
60	CONTINUE	THRMAL	107

18-81

135	IPPL = 1.0111	THERMAL	106
	C	THERMAL	109
	C--- PREPARE SUMMARY	THERMAL	110
	C	THERMAL	111
	C--- JKL = 1 FOR DISPLACEMENT	THERMAL	112
140	C JKL = 2 FOR VELOCITY	THERMAL	113
	C	THERMAL	114
	70 DU 700 JKL=1,2	THERMAL	115
	DTM = 1.	THERMAL	116
	IF (JAL.EV.1) GJ TO 80	THERMAL	117
145	DTM = DTM	THERMAL	118
	VP = VPV	THERMAL	119
	CONST = CONSTV	THERMAL	120
	SUCOMP = SUCOMV	THERMAL	121
	90 IF (DIM.EV.0.) DTM = 1.E-38	THERMAL	122
150	DTNSQ = DIM**2	THERMAL	123
	C	THERMAL	124
	C--- IJK = 1, 2, 3 GAUSSIAN COMPONENTS	THERMAL	125
	C IJK = 4 MAXWELLIAN SPEED AND TOTAL DISPLACEMENT	THERMAL	126
	C	THERMAL	127
155	DU 700 IJK=1,4	THERMAL	128
	C	THERMAL	129
	C--- CALCULATE DISTRIBUTION PARAMETERS	THERMAL	130
	SV = 0.	THERMAL	131
	SVSQ = 0.	THERMAL	132
160	SVSQSQ = 0.	THERMAL	133
	DU 190 I=1,11	THERMAL	134
	II = IDENT(I)	THERMAL	135
	CALL MTBIN (II,K9,L9,M9,J9)	THERMAL	136
	IF (.NOT.GENV9.AND.M9.NE.1) GO TO 190	THERMAL	137
165	IF (JKL.EV.2) GO TO 100	THERMAL	138
	AA = TH(I)	THERMAL	139
	CALL UNPACS (AA,VX1,VY1,VZ1)	THERMAL	140
	IF (GENV9) GO TO 90	THERMAL	141
	GO TO 110	THERMAL	142
170	90 AA = VX(I)	THERMAL	143
	CALL UNPACS (AA,X1,Y1,Z1)	THERMAL	144
	GO TO 120	THERMAL	145
175	100 AA = VX(I)	THERMAL	146
	CALL UNPACS (AA,VX1,VY1,VZ1)	THERMAL	147
180	110 AA = X(1)	THERMAL	148
	CALL UNPACS (AA,X1,Y1,Z1)	THERMAL	149
	GO TO (130,140,150,170), IJK	THERMAL	150
185	130 V = X1-VX1	THERMAL	151
	GO TO 160	THERMAL	152
190	140 V = Y1-VY1	THERMAL	153
	GO TO 160	THERMAL	154
	150 V = Z1-VZ1	THERMAL	155
	160 V = V/DTM	THERMAL	156
	VSQ = V**2	THERMAL	157
195	GO TO 180	THERMAL	158
	170 VSQ = ((X1-VX1)**2+(Y1-VY1)**2+(Z1-VZ1)**2)/DTNSQ	THERMAL	159
	V = SQRT(VSQ)	THERMAL	160
	180 SV = SV+V	THERMAL	161
	SVSQ = SVSQ+VSQ	THERMAL	162
200	SVSQSQ = SVSQSQ+VSQ**2	THERMAL	163
	190 CONTINUE	THERMAL	164

5-82

		VBAR = SV/EN	THRMAL	165
		VAR = (SVSDU-EN*VBAR**2)/(EN-1.)	THRMAL	166
		IF (VAR.LE.0.) VAR = 1.E-38	THRMAL	167
175		VSDBAR = SVSD/EN	THRMAL	168
		IF (IJK.E4.4) GO TO 200	THRMAL	169
		VARVSD = (SVSDSDU-EN*VSDBAR**2)/(EN-1.)	THRMAL	170
		IF (VARVSD.LE.0.) VARVSD = 1.E-38	THRMAL	171
		VPKALC = 1.E-38	THRMAL	172
180		IF (VSDBAR.GT.0.) VPKALC = VARVSD/VSDBAR	THRMAL	173
		ALPHA = VSDBAR/VPKALC	THRMAL	174
		VPKALC = SGRT(VPKALC)	THRMAL	175
		IF (JAL.E4.2) GO TO 230	THRMAL	176
		VP = VPKALC	THRMAL	177
185		GO TO 210	THRMAL	178
	200	IF (JAL.E4.2) GO TO 230	THRMAL	179
		VP = SGRT(VSDBAR**2.)	THRMAL	180
	210	CUNST = VP**2	THRMAL	181
		SUCOMP = VP/SWRT2	THRMAL	182
190		ENRGKN = VAR	THRMAL	183
		IF (IJK.E4.4) GO TO 220	THRMAL	184
		EXPKE = SUCOMP**2	THRMAL	185
		EXPV = 0.	THRMAL	186
		GO TO 240	THRMAL	187
195	220	EXPV = 2./SGRTPI*VP	THRMAL	188
		EXPKE = 1.5*CUNST-EXPV**2	THRMAL	189
		GO TO 260	THRMAL	190
	230	ENRGKN = SVSD/EN*.5	THRMAL	191
		IF (IJK.E4.4) GO TO 250	THRMAL	192
200		EXPV = 0.	THRMAL	193
		EXPKE = CUNST/4.	THRMAL	194
	240	KDIFV = 0.	THRMAL	195
		KHNUDE = 0.	THRMAL	196
		IF (SUCOMP.LE.0.) GO TO 270	THRMAL	197
205		KDIFV = (VBAR-EXPV)/SUCOMP	THRMAL	198
		KHNUDE = VPKALC/SUCOMP	THRMAL	199
		GO TO 270	THRMAL	200
	250	EXPV = 2./SGRTPI*VP	THRMAL	201
		EXPKE = .75*CUNST	THRMAL	202
210	260	KDIFV = 0.	THRMAL	203
		IF (EXPV.GT.0.) KDIFV = (VBAR-EXPV)/EXPV*100.	THRMAL	204
		KHNUDE = 0.	THRMAL	205
		IF (VP.NE.0.) KHNUDE = (VPKALC-VP)/VP*100.	THRMAL	206
	270	KDIFKE = 0.	THRMAL	207
215		IF (EXPKE.GT.0.) KDIFKE = (ENRGKN-EXPKE)/EXPKE*100.	THRMAL	208
	C		THRMAL	209
	C----	SET UP HISTOGRAM BINS AND THEORETICAL NUMBER OF ENTRIES IN THEM	THRMAL	210
		IF (IJK.E4.4) GO TO 280	THRMAL	211
		NVP = 21	THRMAL	212
220		DV = 7./((SGRT2*39.)	THRMAL	213
		GO TO 240	THRMAL	214
	280	NVP = 15	THRMAL	215
		K = 2	THRMAL	216
		DV = (2./SGRTPI-1.)/FLOAT(K)	THRMAL	217
225	290	PK = 0.	THRMAL	218
		ENSUM = 0.	THRMAL	219
		DO 320 I=1,41	THRMAL	220
		NUHSRV(I) = 0	THRMAL	221

5-83

	THFJNY(1) = 0.	THRMAL	222
230	VHIST(1) = 0.	THRMAL	223
	IF (TEMP.LT.0.) GO TO 320	THRMAL	224
	IF (I.EQ.1) GO TO 320	THRMAL	225
	PKM1 = PK	THRMAL	226
	VHIST1 = (FLD4T(1-NVP)+.5)*DV	THRMAL	227
235	IF (IJK.EQ.4) GO TO 300	THRMAL	228
	PK = (ENF(VHIST1)+1.)/2.	THRMAL	229
	GO TO 310	THRMAL	230
300	VHIST1 = VHIST1+1.	THRMAL	231
	VGAMIN = VHIST1**2	THRMAL	232
240	PK = GAMIN(VGAMIN,A)	THRMAL	233
310	THFJNY(I) = (PK-PKM1)*EN	THRMAL	234
	ENSUM = ENSUM+THFJNY(I)	THRMAL	235
	VHIST(I) = VHIST1*VP	THRMAL	236
320	CONTINUE	THRMAL	237
245	THFJNY(41) = EN*ENSUM	THRMAL	238
	DV = DV*VP	THRMAL	239
	C	THRMAL	240
	C--- FILL HISTOGRAM BINS WITH DATA ENTRIES	THRMAL	241
	DO 460 I=1,N	THRMAL	242
250	II = IDEN(I)	THRMAL	243
	CALL MIBIN (II,K9,L9,M9,J9)	THRMAL	244
	IF (.NOT.(ENVS.AND.M9.NE.1)) GO TO 460	THRMAL	245
	IF (JAL.EQ.2) GO TO 330	THRMAL	246
	AA = TB(I)	THRMAL	247
255	CALL UNPAC3 (AA,VX1,VY1,VZ1)	THRMAL	248
	GO TO 340	THRMAL	249
330	AA = VX(I)	THRMAL	250
	CALL UNPAC3 (AA,VX1,VY1,VZ1)	THRMAL	251
340	AA = X(I)	THRMAL	252
260	CALL UNPAC3 (AA,X1,Y1,Z1)	THRMAL	253
	GO TO (350,360,370,390), IJK	THRMAL	254
350	V = X1-VX1	THRMAL	255
	GO TO 380	THRMAL	256
360	V = Y1-VY1	THRMAL	257
265	GO TO 360	THRMAL	258
370	V = Z1-VZ1	THRMAL	259
380	V = V/DEN	THRMAL	260
	VSQ = V**2	THRMAL	261
	GO TO 400	THRMAL	262
270	390 VSQ = ((X1-VX1)**2+(Y1-VY1)**2+(Z1-VZ1)**2)/DTNSQ	THRMAL	263
	V = SQRT(VSQ)	THRMAL	264
	C	THRMAL	265
	C--- DETERMINE THE BIN INDEX K	THRMAL	266
	K = 1 IF V .LT. VHIST(1)	THRMAL	267
275	C K = 41 IF V .GE. VHIST(41)	THRMAL	268
	C OTHERWISE K SATISFIES VHIST(K-1) .LE. V .LT. VHIST(K)	THRMAL	269
400	K = NVP	THRMAL	270
	IF (DV.GT.0.) K = (V-VHIST(1))/DV	THRMAL	271
	K = MIN0(K,40)	THRMAL	272
280	K = MAX0(K,1)	THRMAL	273
410	IF (V.LT.VHIST(K)) GO TO 420	THRMAL	274
	K = K+1	THRMAL	275
	IF (K=41) 410,450,450	THRMAL	276
420	IF (K=1) 440,450,450	THRMAL	277
285	430 IF (V.GE.VHIST(K-1)) GO TO 450	THRMAL	278

5-84

	K = K-1	THERMAL	279
	GO TO 420	THERMAL	280
440	K = 1	THERMAL	281
450	NUBSRV(K) = NUBSRV(K)+1	THERMAL	282
290	460 CONTINUE	THERMAL	283
	C	THERMAL	284
	C--- DETERMINE MAXIMUM BIN COUNT	THERMAL	285
	C NBIG DETERMINED FROM BINS 2,3,...,40	THERMAL	286
	C NBIG DETERMINED FROM BINS 1,2,3,...,40,41	THERMAL	287
245	K = 0	THERMAL	288
	NBIG = 0	THERMAL	289
	DO 470 I=1,41	THERMAL	290
	IF (NBIG.GE.NUBSRV(I)) GO TO 470	THERMAL	291
	NBIG = NUBSRV(I)	THERMAL	292
300	K = I	THERMAL	293
	470 CONTINUE	THERMAL	294
	C	THERMAL	295
	C--- ESTIMATE THE MODE OF THE OBSERVED DISTRIBUTION BY FINDING THE	THERMAL	296
	C MAXIMUM OF A LEAST SQUARE PARABOLA CALCULATED FROM ALL BINS WHOSE	THERMAL	297
305	C COUNT EXCEEDS HALF THE MAXIMUM BIN COUNT.	THERMAL	298
	S = 0.	THERMAL	299
	SX = 0.	THERMAL	300
	SX2 = 0.	THERMAL	301
	SX3 = 0.	THERMAL	302
310	SX4 = 0.	THERMAL	303
	SF = 0.	THERMAL	304
	SXF = 0.	THERMAL	305
	SX2F = 0.	THERMAL	306
	AJ = 0.	THERMAL	307
315	DO 490 I=2,40	THERMAL	308
	AJ = AJ+1.	THERMAL	309
	IF (2*NUBSRV(I).LT.NBIG) GO TO 480	THERMAL	310
	AJ2 = AJ**2	THERMAL	311
	F = NUBSRV(I)	THERMAL	312
320	S = S+1.	THERMAL	313
	SX = SX+AJ	THERMAL	314
	SX2 = SX2+AJ2	THERMAL	315
	SX3 = SX3+AJ2*AJ	THERMAL	316
	SX4 = SX4+AJ2*AJ2	THERMAL	317
325	SF = SF+F	THERMAL	318
	SXF = SXF+F*AJ	THERMAL	319
	SX2F = SX2F+F*AJ**2	THERMAL	320
480	CONTINUE	THERMAL	321
	IF (S.LE.0.) GO TO 490	THERMAL	322
330	XDEN = S*SX2*SX2F+SX*SX3*SF+SX*SX2*SXF-SX2*SX2*SF-SX*SX*SX2F-S*SX3	THERMAL	323
	+*SXF	THERMAL	324
	IF (XDEN.EQ.0.) GO TO 490	THERMAL	325
	XNUM = S*SX4*SXF+SX2*SX3*SF+SX*SX2*SX2F-SX2*SX2*SXF-SX*SX4*SF-S*SX	THERMAL	326
	+*SXF	THERMAL	327
335	VPCALC = XNUM/(S*(S+1)+XDEN)	THERMAL	328
	GO TO 520	THERMAL	329
490	IF (IJK.EQ.4) GO TO 500	THERMAL	330
	IF (K.GT.1) GO TO 500	THERMAL	331
	VPCALC = 100.*VPCALC	THERMAL	332
540	GO TO 520	THERMAL	333
500	IF (K.LT.41) GO TO 510	THERMAL	334
	VPCALC = 100.*VPCALC	THERMAL	335

5-85

```

      (J TO 52)
      VPK1 = V.
      IF (K.GI.1) VARI = VHIST(K-1)
      VPCALC = (VARI+VHIST(K))/2.
      C
      520 J = 1
      IF (IJK,EN.4) J = 2
      350 WRITE (6,820) IPKE, (MESSAGE(1,JKL,J),I=1,3), ICOMP(IJK)
      NO = EN
      WRITE (6,750) TEMP,NO,DIN
      RDIFVP = V.
      IF (IJK,EN.4) GO TO 530
      355 VPKALC = SQRT(VAR)
      NDSIG = 0.
      IF (SDCUMP,NE.0.) NDSIG = (VPKALC-SDCUMP)/SDCUMP
      JTAG = ISIGMA
      IF (SDCUMP,GT.0.) RDIFVP = VPCALC/SDCUMP
      360 WRITE (6,800) ITLET(JKL,1),VBAN,EAPV,RDIFV,ISIGMA
      WRITE (6,850) ITLET(JKL,2),ENHGN,EXPKE,RDIFKE,IPERCN
      WRITE (6,840) ITLET(JKL,3),VPCALC,RDIFVP,ISIGMA
      WRITE (6,830) JTAG,VPKALC,SDCUMP,NDSIG,ISIGMA
      ITAG = BLANK
      365 IF (JKL,NE.1) ITAG = IVEE
      JTAG = ICOMP(IJK)
      GO TO 540
      530 NDSIG = (ALPHA-A)/A*100.
      JTAG = IALPHA
      IF (VP,GT.0.) RDIFVP = (VPCALC-VP)/VP*100.
      WRITE (6,850) ITLET(JKL,1),VBAN,EAPV,RDIFV,IPERCN
      WRITE (6,850) ITLET(JKL,2),ENHGN,EXPKE,RDIFKE,IPERCN
      WRITE (6,850) ITLET(JKL,3),VPKALC,VP,NOMNOE,IPERCN,VPCALC,RDIFVP,I
      +PERCN
      370 WRITE (6,850) JTAG,ALPHA,A,NDSIG,IPERCN
      ITAG = BLANK
      JTAG = IANR
      IF (JKL,NE.1) JTAG = IVEE
      C
      380 C--- HISTOGRAM PRINTOUT
      540 IF (VPKALC,EO.0.) VPKALC = 1.E-38
      DENS = 1.
      IF (NDIG,GT.102) DENS = 102./FLOAT(NDIG)
      IF (NDIG,GE.153) DENS = DENS/1.5
      385 PK = 1./DENS
      WRITE (6,810) ITAG,JTAG,PK
      PK = 0.
      ENSUM = 0.
      KALC = 0
      390 DO 600 I=1,41
      IF (TEMP,LF.0.) GO TO 580
      IF (1.LI.41) GO TO 550
      KALC = EN*ENSUM
      GO TO 560
      395 550 PKM1 = P1
      VHIST1 = VHIST(1)
      IF (IJK,EN.4) GO TO 560
      VHIST1 = (VHIST1+VAR)/VPKALC/SQRT2
      PK = (ENF(VHIST1)+1.)/2.
      THERMAL 336
      THERMAL 337
      THERMAL 338
      THERMAL 339
      THERMAL 340
      THERMAL 341
      THERMAL 342
      THERMAL 343
      THERMAL 344
      THERMAL 345
      THERMAL 346
      THERMAL 347
      THERMAL 348
      THERMAL 349
      THERMAL 350
      THERMAL 351
      THERMAL 352
      THERMAL 353
      THERMAL 354
      THERMAL 355
      THERMAL 356
      THERMAL 357
      THERMAL 358
      THERMAL 359
      THERMAL 360
      THERMAL 361
      THERMAL 362
      THERMAL 363
      THERMAL 364
      THERMAL 365
      THERMAL 366
      THERMAL 367
      THERMAL 368
      THERMAL 369
      THERMAL 370
      THERMAL 371
      THERMAL 372
      THERMAL 373
      THERMAL 374
      THERMAL 375
      THERMAL 376
      THERMAL 377
      THERMAL 378
      THERMAL 379
      THERMAL 380
      THERMAL 381
      THERMAL 382
      THERMAL 383
      THERMAL 384
      THERMAL 385
      THERMAL 386
      THERMAL 387
      THERMAL 388
      THERMAL 389
      THERMAL 390
      THERMAL 391
      THERMAL 392

```

400	500	GO TO 570	THERMAL	393
		VGAMIN = (VHIST1/VPRALC)**2	THERMAL	394
		PK = 1.	THERMAL	395
		IF (VGAMIN.LT.100.) PK = GAMIN/VGAMIN,ALPHA)	THERMAL	396
	570	CALC = (PK-PRMI)*EN	THERMAL	397
405		ENSUM = ENSUM+CALC	THERMAL	398
		KALC = CALC+.5	THERMAL	399
	580	NUPSR = NUSSRV(1)	THERMAL	400
		J2 = FLOAT(NOBSS)*DENOM+.5	THERMAL	401
		J2 = MIN(J2,102)	THERMAL	402
410		LINE(1) = STAR	THERMAL	403
		IF (J2.GT.0) GO TO 590	THERMAL	404
		J2 = 1	THERMAL	405
		LINE(1) = BLANK	THERMAL	406
	590	NTHED = THEORY(1)+.5	THERMAL	407
415		IF (I.EQ.41) GO TO 600	THERMAL	408
		WRITE (6,740) VHIST(I),KALC,NTHED,NOBSS,(LINE(J),J=1,J2)	THERMAL	409
	600	CONTINUE	THERMAL	410
		WRITE (6,750) KALC,NTHED,NOBSS,(LINE(J),J=1,J2)	THERMAL	411
	C		THERMAL	412
420	C---=	CALCULATE CHI SQUARE STATISTIC	THERMAL	413
		IF (N.GE.10) GO TO 610	THERMAL	414
		WRITE (6,770)	THERMAL	415
		GO TO 700	THERMAL	416
	610	K = 1	THERMAL	417
425		I = 0	THERMAL	418
	620	NUB = 0	THERMAL	419
		PK = 0.	THERMAL	420
	630	PK = PK+THEORY(K)	THERMAL	421
		NUB = NUB+NOBSSRV(K)	THERMAL	422
430		K = K+1	THERMAL	423
		IF (K.GT.41) GO TO 640	THERMAL	424
		IF (PK.LT.5.) GO TO 630	THERMAL	425
		I = I+1	THERMAL	426
		THEORY(I) = PK	THERMAL	427
435		NOBSSRV(I) = NUB	THERMAL	428
		GO TO 620	THERMAL	429
	640	IF (PK.GE.5.) GO TO 650	THERMAL	430
		THEORY(I) = PK+THEORY(I)	THERMAL	431
		NOBSSRV(I) = NOBSSRV(I)+NUB	THERMAL	432
440		GO TO 660	THERMAL	433
	650	I = I+1	THERMAL	434
		THEORY(I) = PK	THERMAL	435
		NOBSSRV(I) = NUB	THERMAL	436
	660	NDF = I-1	THERMAL	437
445		IF (NDF.GE.0) GO TO 670	THERMAL	438
		WRITE (6,780)	THERMAL	439
		GO TO 700	THERMAL	440
	670	SV = 0.	THERMAL	441
		DO 680 K=1,I	THERMAL	442
450		UBS = NOBSSRV(K)	THERMAL	443
	680	SV = SV+((ABS(UBS-THEORY(K))-.)**2)/THEORY(K)	THERMAL	444
		WRITE (6,790) SV,NDF	THERMAL	445
		IF (SV.LT.200.) GO TO 690	THERMAL	446
		WRITE (6,830)	THERMAL	447
455		GO TO 700	THERMAL	448
	690	UBS = FLOAT(NDF)/2.	THERMAL	449

460 700 PHI = SV/2.
 PA = 1.0E+01 (P11,005)
 WRITE (6,790) SV,PA,SGLVL5,CHISQ5(NDF),SGLVL1,CHISQ1(NDF)
 CONTINUE
 RETURN
 C
 (--- ERROR CORRUPTED DETECTED
 710 WRITE (6,720)
 STOP
 465 C
 C
 C
 720 FORMAT ('HOSTIP IN THRMAL BECAUSE OF ERROR.')

470 730 FORMAT ('HOTTEMPERATURE 7.1, 24M DEG. K NO. OF ATOMS 15, 15H
 +TIME INCM. 1PE11.4/1H0YX4HDATA9A55HTHEORY RELATIVE DIFF. HIS
 +TUGRA1 RELATIVE DIFF./')

740 FORMAT ('M FB.5,316,102A1')
 750 FORMAT ('41 IN INIIV,316,102A1')
 475 760 FORMAT ('MOCHI**2,P10.3,5X33HDEG.FREEDOM USED FOR CHI**2 CALC.,13')
 770 FORMAT ('H0B000050H TOO FEW ATOMS TREATED TO WARRANT COMPUTING CHI
 +**2,0M >>>>')
 780 FORMAT ('610600050H DEGREES OF FREEDOM MUST BE POSITIVE FOR MEANIN
 +GFUL CHI**2,0M >>>>')

480 790 FORMAT ('10HOPKUS(CHI**2 > F10.3,4M) = 2PFB.1,0H PERCENT//1H FB.1,4
 +2H PERCENT SIGNIFICANCE LEVEL CRITICAL VALUEOPF10.2/1H 2PFB.1,42H
 +PERCENT SIGNIFICANCE LEVEL CRITICAL VALUEOPF10.2')

800 FORMAT ('1A45,1PE13.5,OPFB.4,1A7,1X1PF13.5,OPFB.4,1XA7')
 810 FORMAT ('4H0 2A1,27H DATA THEORY OBSERVED 50X14HEACH * DENOT
 +FB,FB.2,7H COUNTS/')

820 FORMAT ('30H15SUMMARY OF THERMALIZATION PROCEDURESX3A10,A4,A1')
 830 FORMAT ('70HUA CHI**2 IN EXCESS OF 5*(DEGREES OF FREEDOM) IS CONSID
 +ERED TOO LARGE.')

840 FORMAT ('1A45,43X1PE13.5,OPFB.4,1XA7')
 490 850 FORMAT ('1A45,1PE13.5,OPFB.2,1XA7,1X1PE13.5,OPFB.2,1XA7')
 END
 THRMAL 450
 THRMAL 451
 THRMAL 452
 THRMAL 453
 THRMAL 454
 THRMAL 455
 THRMAL 456
 THRMAL 457
 THRMAL 458
 THRMAL 459
 THRMAL 460
 THRMAL 461
 THRMAL 462
 THRMAL 463
 THRMAL 464
 THRMAL 465
 THRMAL 466
 THRMAL 467
 THRMAL 468
 THRMAL 469
 THRMAL 470
 THRMAL 471
 THRMAL 472
 THRMAL 473
 THRMAL 474
 THRMAL 475
 THRMAL 476
 THRMAL 477
 THRMAL 478
 THRMAL 479
 THRMAL 480
 THRMAL 481
 THRMAL 482
 THRMAL 483
 THRMAL 484

```

1      FUNCTION ERF (Y)                                ERF      2
      C                                                ERF      3
      C--- REFERENCE HANDBOOK OF MATH FUNCTIONS NBS PUBLICATION NO. 55 ERF      4
      C      EQUATION 7.1.28 P.299 ERF      5
      C       $ERF(X) = 2/\sqrt{\pi} * (INTEGRAL FROM 0 TO X OF EXP(-X**2)DX)$  ERF      6
      C      MAXIMUM ERROR IS .LE. 3.E-7 ERF      7
      C                                                ERF      8
      C      DIMENSION A(7) ERF      9
      C      DATA (A(I),I=1,7)/4.30638E-5,2.70567E-4,1.520143E-4,9.2705272E-3, ERF     10
      C      + 42.2540123E-3,70.5230784E-3,1./ ERF     11
      C      YY = ABS(Y) ERF     12
      C      IF (YY=6.) 30,20,20 ERF     13
      C      ERF = 1. ERF     14
      C      GO TO 70 ERF     15
      C      IF (YY=.000010) 40,40,50 ERF     16
      C      ERF = 2.*YY/1.7725 ERF     17
      C      GO TO 70 ERF     18
      C      SDFG = A(1) ERF     19
      C      DO 60 I=2,7 ERF     20
      C      SDFG = SDFG*YY+A(I) ERF     21
      C      ERF = 1.-1./(((3072**2)**2)**2)**2) ERF     22
      C      ERF = SIGN(ERF,Y) ERF     23
      C      RETURN ERF     24
      C      END ERF     25

```

5-89

5-90

```

1      REAL FUNCTION INGVN(ALPHA,INGAM)
C
C----- THIS FUNCTION COMPUTES THE INVERSE OF THE INCOMPLETE GAMMA
C FUNCTION WITH PARAMETER ALPHA, WHOSE VALUE IS INGAM. NEWTONS
5      METHOD IS USED.
C
C      REAL INGAM
C
C      DATA GAM302/.8682264254/
10     DATA NMAX,HIGNO/100,9.99E99/
C
C      A = ALPHA
C      XI = INGAM
C      AM1 = A-1.
15     IF (A.LE.0.) GO TO 110
C      IF (XI.LT.0.) GO TO 110
C      IF (XI.GT.1.) GO TO 110
C      IF (XI.EQ.0.) GO TO 130
C      IF (XI.EQ.1.) GO TO 120
20
C
C----- GET STARTING VALUE FOR X
C      XN = SQRT(A)
C      IF (XI-AM1) 20,30,40
25     20    XN = -XN
C      GO TO 40
30     30    XN = 0.
C      40    XN = A+XN
C
C----- SOLVE FOR XN BY NEWTONS METHOD
C      N = 0
35     50    F = GAMMA(XN,A)-XI
C      IF (F.EQ.0.) GO TO 90
C      FP = EXP(-XN)/GAM302*XN**AM1
C      IF (FP.LE.0.) GO TO 110
40     75    DX = F/FP
C      IF (ABS(DX/XN).LE..5E-7) GO TO 80
C      DX = SIGN(AMIN1(XN/4.,ABS(DX)),DX)
C      XN = XN+DX
C      N = N+1
45     60    IF (N.GE.NMAX) GO TO 100
C      IF (XN) 70,70,50
50     70    XN = XN+DX/10.
C      GO TO 60
55     80    XN = XN+DX
C      INGVN = XN
C      RETURN
C
C----- ERROR ENCOUNTERED
100    XN = -XN
C      GO TO 90
110    XN = -HIGNO
C      GO TO 90
C
C----- SPECIAL VALUES
120    XN = HIGNO
C      GO TO 90
130    XN = 0.

```

```

INGVN 2
INGVN 3
INGVN 4
INGVN 5
INGVN 6
INGVN 7
INGVN 8
INGVN 9
INGVN 10
INGVN 11
INGVN 12
INGVN 13
INGVN 14
INGVN 15
INGVN 16
INGVN 17
INGVN 18
INGVN 19
INGVN 20
INGVN 21
INGVN 22
INGVN 23
INGVN 24
INGVN 25
INGVN 26
INGVN 27
INGVN 28
INGVN 29
INGVN 30
INGVN 31
INGVN 32
INGVN 33
INGVN 34
INGVN 35
INGVN 36
INGVN 37
INGVN 38
INGVN 39
INGVN 40
INGVN 41
INGVN 42
INGVN 43
INGVN 44
INGVN 45
INGVN 46
INGVN 47
INGVN 48
INGVN 49
INGVN 50
INGVN 51
INGVN 52
INGVN 53
INGVN 54
INGVN 55
INGVN 56
INGVN 57
INGVN 58

```

FUNCTION INGV 76/76 DPT=0 TRACE

FTN 4.5+414

11/15/79 14.17.08

PAGE 2

GO TO 90
END

INGINV 59
INGINV 60

5-92

```

IDENT NEAF
* THIS SUBROUTINE, WRITTEN IN COMPASS, CALCULATES THE RESTORING FORCE,
* THE COMPONENTS OF WHICH ARE ZZX, ZZY, AND ZZZ, AND CONFIGURATIONAL
* ENERGY ENGPU FOR EACH ATOM IN EITHER THE ENTIRE LATTICE OR THE MINI-
* LATTICE, EACH TIME STEP.
*
* THIS ROUTINE IS WRITTEN SPECIFICALLY FOR LCM VARIABLES. IF FOR
* ANY REASON, IT IS RE-WRITTEN, PLEASE REVIEW ALL INSTRUCTIONS
* BEFORE ATTEMPTING EXECUTION.
ENTRY NEAF
USE /ELMT/
0      1      ENGPU  BSS  1
1      1      ID      BSS  1
2      1      E1      BSS  1
3      1      E2      BSS  1
4      1      J3      BSS  1
5      1      DIV      BSS  1
6      1      4MAX2    BSS  1
7      1      INIS     BSS  1
10     1      VJ2      BSS  1
11     1      ML       BSS  1
12     1      IELE     BSS  1
13     1      L        BSS  1
14     1      LY       BSS  1
15     1      IDT      BSS  1
      USE /CENTR/
0      1      X11      BSS  1
1      1      Y11      BSS  1
2      1      Z11      BSS  1
3      1      XX1      BSS  1
4      1      YY1      BSS  1
5      1      ZZ1      BSS  1
6      13     EXTRA    BSS  11
      USE LCM /ALPHA/
J      14474     X      BSS  6460
14474  14474     ZZX     BSS  6460
3117J  14474     ZZY     BSS  6460
45664  14474     ZZZ     BSS  6460
6236J  14474     ENGPU   BSS  6460
77054  14474     IDENT   BSS  6460
J13550 34410     IGUX     BSS  14000
J50150 1751     EA       BSS  1001
J52131 1751     EEA      BSS  1001
J54102 1751     EB       BSS  1001
J56053 1751     EEU      BSS  1001
J60024 1751     EAU      BSS  1001
J61775 1751     EEA0     BSS  1001
      USE 0
J      1      IPP      BSS  1
1      1      X22      BSS  1
2      1      Y22      BSS  1
3      1      Z22      BSS  1
4      1      JX       BSS  1
5      1      JY       BSS  1
6      1      JZ       BSS  1
7      1      IE       BSS  1
10     1      IE1      BSS  1
NEAF 2
NEAF 3
NEAF 4
NEAF 5
NEAF 6
NEAF 7
NEAF 8
NEAF 9
NEAF 10
NEAF 11
NEAF 12
NEAF 13
NEAF 14
NEAF 15
NEAF 16
NEAF 17
NEAF 18
NEAF 19
NEAF 20
NEAF 21
NEAF 22
NEAF 23
NEAF 24
NEAF 25
NEAF 26
NEAF 27
NEAF 28
NEAF 29
NEAF 30
NEAF 31
NEAF 32
NEAF 33
NEAF 34
COM2 2
COM2 3
COM2 4
COM2 5
COM2 6
COM2 7
COM2 8
COM2 9
NEAF 36
NEAF 37
NEAF 38
NEAF 39
NEAF 40
NEAF 41
NEAF 42
NEAF 43
NEAF 44
NEAF 45
NEAF 46
NEAF 47
NEAF 48
NEAF 49
NEAF 50
NEAF 51

```


NEAF

COMPASS 5.0-397.

11/15/79 14.17.43.

PAGE

3

11		1	RP	BSS	1		NEAF	52
12		1	AMARK	BSS	1		NEAF	53
13		4	100	BSS	4		NEAF	54
17		1	F1	BSS	1		NEAF	55
20		1	P1	BSS	1		NEAF	56
21		1	JELM	BSS	1		NEAF	57
22		1	IOVE	BSS	1		NEAF	58
23		1	NEAF	BSS	1		NEAF	59
24	43700			MA7	0		NEAF	60
	5170000015 +			SA7	10N		NEAF	61
25	5170000014 +			SA7	10N+1		NEAF	62
	5170000015 +			SA7	10N+2		NEAF	63
26	5170000010 +			SA7	10N+3		NEAF	64
	5110000004 C			SA1	00+J3		NEAF	65
27	43217			MA2	10		NEAF	66
	21217			LX2	10		NEAF	67
	11612			BX6	X1*X2		NEAF	68
30	0500000025 +			ZK	X0, NEAF		NEAF	69
	5160000015 C			SA6	10T		NEAF	70
31	21117			AX1	10		NEAF	71
	11012			BX6	X1*X2		NEAF	72
	0306000030 +			ZK	X0, CUMPT		NEAF	73
32	5160000014 +			SA6	10N+1		NEAF	74
	21117			AX1	10		NEAF	75
	11612			BX6	X1*X2		NEAF	76
33	0306000035 +			ZK	X0, CUMPT		NEAF	77
	5160000015 +			SA6	10N+2		NEAF	78
34	21117			AX1	10		NEAF	79
	11012			BX6	X1*X2		NEAF	80
	5160000010 +			SA6	10N+3		NEAF	81
35	43601		COMPT	MX6	1		NEAF	82
	20001			LY6	1		NEAF	83
	5160000012 +			SA6	AMARK		NEAF	84
36	5160000022 +			SA6	10NE		NEAF	85
37	5110000015 C		COMPT1	SA1	00+10T		NEAF	86
	5120000001 C			SA2	00+10		NEAF	87
40	37321			IX3	X2=X1		NEAF	88
	5140000011 C			SA4	00+ML		NEAF	89
41	0304000042 +			ZK	X4, JUMP		NEAF	90
	0323000177 +			PL	X3, SATH		NEAF	91
42	0303000177 +		JUMP	ZK	X3, SATH		NEAF	92
	5120000022 +			SA2	10NE		NEAF	93
43	37117			IX1	X1=X2		NEAF	94
	10622			BX6	X2		NEAF	95
	5160000000 +			SA6	10T		NEAF	96
44	5140000012 C			SA4	00+10LE		NEAF	97
	37642			IX6	X4=X2		NEAF	98
45	0306000052 +			ZK	X0, MET		NEAF	99
	7201007705 + C			SK0	X1+10ENT		NEAF	100
46	01430			MX3	X0		NEAF	101
	21230			AX3	10		NEAF	102
	45417			MX4	10		NEAF	103
	20417			LY4	10		NEAF	104
47	11634			BX6	X3=X4		NEAF	105
	5160000014 C			SA6	10T		NEAF	106
50	5150000015 C			SA5	00+L		NEAF	107
	50756			IX7	X5=X0		NEAF	108

5-93

NEAF

COMPASS 3.0-347.

11/13/79 14.17.43.

PAGE

4

51 5160000000 + 37672
 52 7201000000 C 01430 43424
 53 5150000005 + 20457 20351
 54 11631 12765 5170000004 +
 55 20324 11034 12765
 56 5170000002 + 20324 11634
 57 12765 5170000003 +
 60 5110000000 C 5120000001 +
 61 31612 24606 10566 40756
 62 5160000004 + 5110000001 C
 63 5120000002 + 31612 24606
 64 10566 40456 30374 24303
 65 5160000005 + 5110000002 C
 66 5120000003 + 31612 24606
 67 10566 40456 30743 24707
 70 5170000011 + 5160000000 +
 71 5110000006 C 31617
 72 0336000177 + 5110000002 C
 73 40217 5130000003 C 30423
 74 24404 20774 22677 43314
 75 20314

NET

IX6 X1=X2
 SA6 1P
 SX0 X1+X
 YX3 XU
 MX4 2U
 SA5 01AS
 LX4 4/
 LX3 4/
 BX6 X3*X4
 BX7 X0+X3
 SA7 BU+X22
 LX3 2U
 BX6 X3*X4
 BX7 X0+X3
 SA7 BU+Y22
 LX3 2U
 BX6 X3*X4
 BX7 X0+X3
 SA7 BU+Z22
 SA1 BU+X11
 SA2 BU+X22
 FX6 X1=X2
 NX6 BU,X6
 BX5 X6
 FX7 X5*X6
 SA6 BU+DX
 SA1 BU+Y11
 SA2 BU+Y22
 FX6 X1=X2
 NX6 BU,X6
 BX5 X6
 FX4 X5*X6
 FX3 X1+X4
 NX3 BU,X3
 SA6 BU+UY
 SA1 BU+Z11
 SA2 BU+Z22
 FX6 X1=X2
 NX6 BU,X6
 BX5 X6
 FX4 X5*X6
 FX7 X4+X3
 NX7 BU,X7
 SA7 BU+X11
 SA6 BU+UZ
 SA1 HMAX2
 FX6 X1=X7
 NG X6,BWTH
 SA1 E1
 FX2 X1*X7
 SA3 E2
 FX4 X2+X3
 NX4 BU,X4
 UX7 07,X4
 LX6 07,X7
 MX3 12
 LX3 12

DX**2
 DX

UY**2

UY

OZ**2

OZ

NEAF 109
 NEAF 110
 NEAF 111
 NEAF 112
 NEAF 113
 NEAF 114
 NEAF 115
 NEAF 116
 NEAF 117
 NEAF 118
 NEAF 119
 NEAF 120
 NEAF 121
 NEAF 122
 NEAF 123
 NEAF 124
 NEAF 125
 NEAF 126
 NEAF 127
 NEAF 128
 NEAF 129
 NEAF 130
 NEAF 131
 NEAF 132
 NEAF 133
 NEAF 134
 NEAF 135
 NEAF 136
 NEAF 137
 NEAF 138
 NEAF 139
 NEAF 140
 NEAF 141
 NEAF 142
 NEAF 143
 NEAF 144
 NEAF 145
 NEAF 146
 NEAF 147
 NEAF 148
 NEAF 149
 NEAF 150
 NEAF 151
 NEAF 152
 NEAF 153
 NEAF 154
 NEAF 155
 NEAF 156
 NEAF 157
 NEAF 158
 NEAF 159
 NEAF 160
 NEAF 161
 NEAF 162
 NEAF 163
 NEAF 164
 NEAF 165

5-94

74134

4E1A

4720409 949

COMFAS 5.0-247.

U7ED*((31)EE-(1E1)EE)

5 2974

NEAF

COMPASS 3.0-1997.

11/13/79 14.17.43.

PAGE

6

5-6-

125	5160000020 +	24606	4X6	00,X6	NEAF	223	
		5110000013 C	SA6	00+P1	NEAF	224	
126	5120000017 +		SA1	00+XX1	NEAF	225	
		5130000014 +	SA2	00+P1	NEAF	226	
127	40423		SA3	00+DX	NEAF	227	
	30641		FX4	X2+X3	F1*DX	NEAF	228
	24606		FX6	X4+X1	NEAF	229	
130	5160000013 C		NX6	00,X6	NEAF	230	
		5110000013 C	SA6	00+XX1	XX1=XX1+P1*DX	NEAF	231
131	5150000010 C		SA1	00+101	NEAF	232	
	37313		SA3	00+N12	NEAF	233	
132	0323000156 +		IX3	X1=X3	NEAF	234	
	43501		PL	X3,SKIP1	NEAF	235	
	20501		HX5	1	NEAF	236	
133	37114		LX5	1	NEAF	237	
	7201014474 C		IX1	X1=X3	NEAF	238	
	01430		SX0	X1+ZZX	NEAF	239	
134	31634		RX3	X0	NEAF	240	
	24606		FX6	X3=X4	NEAF	241	
	7201014474 C		NX6	00,X6	NEAF	242	
135	01560		SX0	X1+ZZX	NEAF	243	
136	5150000004 C	SKIP1	HX6	X0	NEAF	244	
	5130000003 +		SA5	00+YY1	NEAF	245	
137	40423		SA3	00+DY	F1*DY	NEAF	246
	30645		FX4	X2+X3	NEAF	247	
	24606		FX6	X4+X3	NEAF	248	
140	5160000004 C		NX6	00,X6	NEAF	249	
		5130000013 C	SA6	00+YY1	NEAF	250	
141	5150000010 C		SA3	00+101	NEAF	251	
	37535		SA5	00+N12	NEAF	252	
142	0325000145 +		IX5	X3=X3	NEAF	253	
	7201031170 C		PL	X3,SKIP2	NEAF	254	
143	01450		SX0	X1+ZZY	NEAF	255	
	31654		HX5	X0	NEAF	256	
	24606		FX6	X3=X4	NEAF	257	
144	7201031170 C		NX6	00,X6	NEAF	258	
	01560		SX0	X1+ZZY	NEAF	259	
145	5150000005 C	SKIP2	HX6	X0	NEAF	260	
	5130000006 +		SA5	00+ZZ1	NEAF	261	
146	40432		SA3	00+01	F1*DZ	NEAF	262
	30645		FX4	X3=X2	NEAF	263	
	24606		FX6	X4+X3	NEAF	264	
147	5160000005 C		NX6	00,X6	NEAF	265	
		5130000013 C	SA6	00+ZZ1	ZZ1+F1*DX	NEAF	266
150	5150000010 C		SA3	00+101	NEAF	267	
	37535		SA5	00+N12	NEAF	268	
151	0325000154 +		IX5	X3=X3	NEAF	269	
	7201045664 C		PL	X3,SKIP3	NEAF	270	
152	01450		SX0	X1+ZZZ	NEAF	271	
	31654		HX5	X0	NEAF	272	
	24606		FX6	X3=X4	NEAF	273	
153	7201045664 C		NX6	00,X6	NEAF	274	
	01560		SX0	X1+ZZZ	NEAF	275	
154	5110000010 C	SKIP3	HX6	X0	NEAF	276	
	5120000013 C		SA1	00+N12	NEAF	277	
155	37421		SA2	00+101	NEAF	278	
			IX4	X2=X1	NEAF	279	

HEAT

COMPASS 5.0-347.

11/15/79 14.17.43.

PAGE 7

5-97

0324000170 +
156 5110000020 +
5120000000 C
157 50012
24006
5160000000 C
160 5120000000 C
44312
161 5120000001 C
43401
20401
162 37224
7202062360 C
01410
163 50013
24006
7202062360 C
164 01500
5120000015 C
37224
165 7202062360 C
01410
50013
166 24006
7202062360 C
01500
167 0400000177 +
170 5110000020 +
5120000000 C
171 44312
5140000000 C
30643
172 24006
5160000000 C
173 5120000001 C
43401
20401
174 37224
7202062360 C
01410
175 50013
24006
7202062360 C
176 01500
177 5110000012 +
43201
20201
200 56621
5120000007 C
37450
201 0304000023 +
5221000013 +
202 5160000012 +
10722
203 0307000023 +
5170000013 C
204 0400000037 +

SIDE

SWIM

PL X4, SINE
SA1 BU+P1
SA2 BU+ENGPI
FX6 X1+X2
NX6 BU, X6
SA6 BU+ENGPI
SA2 BU+DIV
FX3 X1/X2
SA2 BU+IU
MX4 1
LX4 1
IX2 X2=X4
SX0 X2+ENGPI
MX1 XU
FX6 X1+X3
NX6 BU, X6
SX0 X2+ENGPI
MX6 XU
EQ BU, BU, SWTH
SA1 BU+P1
SA2 BU+DIV
FX3 X1/X2
SA4 BU+ENGPI
FX6 X4+X3
NX6 BU, X6
SA6 BU+ENGPI
SA2 BU+IU
MX4 1
LX4 1
IX2 X2=X4
SX0 X2+ENGPI
MX1 XU
FX6 X1+X3
NX6 BU, X6
SX0 X2+ENGPI
MX6 XU
SA1 BU+KMARK
MX2 1
LX2 1
IX6 X2+X1
SA5 BU+ININ
LX4 X3=X6
ZR X4, NEAF
SA2 X1+IUN
SA6 BU+KMARK
MX7 X2
ZR X1, NEAF
SA7 BU+IU
EQ BU, BU, COMPT1

KMARK=KMARK+1

NEAF 280
NEAF 281
NEAF 282
NEAF 283
NEAF 284
NEAF 285
NEAF 286
NEAF 287
NEAF 288
NEAF 289
NEAF 290
NEAF 291
NEAF 292
NEAF 293
NEAF 294
NEAF 295
NEAF 296
NEAF 297
NEAF 298
NEAF 299
NEAF 300
NEAF 301
NEAF 302
NEAF 303
NEAF 304
NEAF 305
NEAF 306
NEAF 307
NEAF 308
NEAF 309
NEAF 310
NEAF 311
NEAF 312
NEAF 313
NEAF 314
NEAF 315
NEAF 316
NEAF 317
NEAF 318
NEAF 319
NEAF 320
NEAF 321
NEAF 322
NEAF 323
NEAF 324
NEAF 325
NEAF 326
NEAF 327
NEAF 328
NEAF 329
NEAF 330
NEAF 331
NEAF 332
NEAF 333
NEAF 334
NEAF 335
NEAF 336

NEAF

COMPASS 3.0-397.

11/13/79 14.17.43.

PAGE 8

205 17254000000000000000
206

DATA 01725400000000000000
END

NEAF 337
NEAF 338

30100H SCI STORAGE USED
MODEL 76 ASSEMBLY

344 STATEMENTS
0.186 SECONDS

63 SYMBOLS
190 REFERENCES

PAC3

COMPASS 3.0-397.

11/13/79 14.17.43.

PAGE

6

		IDENT PAC3	PAC3	2
		* THIS SUBROUTINE, WRITTEN IN COMPASS, PACKS THREE REAL NUMBERS BETWEEN	PAC3	3
		* +32.0 AND +63.99 INTO A 60-BIT WORD.	PAC3	4
		ENTRY PAC3	PAC3	5
0	1	SS6 1	PAC3	6
1	5021006001	SA2 A1+1	PAC3	7
	5752000011	SA3 A2+1	PAC3	8
2	5043000001	SA4 A3+1	PAC3	9
	63110	SS1 A1	PAC3	10
	63220	SS2 A2	PAC3	11
3	53330	SS3 A3	PAC3	12
	63440	SS4 A4	PAC3	13
	56120	SA1 A2	PAC3	14
	60110	LX1 13	PAC3	15
4	53224	MA2 20	PAC3	16
	11512	MX5 A1*X2	PAC3	17
	56130	SA1 03	PAC3	18
	60250	LX2 40	PAC3	19
5	21107	AX1 7	PAC3	20
	11412	MX4 A1*X2	PAC3	21
	56140	SA1 04	PAC3	22
	21224	AX2 20	PAC3	23
6	21133	AX1 27	PAC3	24
	11512	MX3 A1*X2	PAC3	25
	12654	MX6 X5+X4	PAC3	26
	12763	MX7 X6+X3	PAC3	27
7	56710	SA7 01	PAC3	28
	0400000000 +	EQ PAC3	PAC3	29
10		END	PAC3	30

5-99

501008 SCP STORAGE USED
MODEL 76 ASSEMBLY

29 STATEMENTS
0.017 SECONDS

1 SYMBOLS
3 REFERENCES

UNPAC3

COMPASS 3.0-397.

11/13/79 14.17.43.

PAGE

2

		IDENT UNPAC3	UNPAC3	2
		* THIS SUBROUTINE, WRITTEN IN COMPASS, UNPACKS THREE REAL NUMBERS	UNPAC3	3
		* BETWEEN +32.0 AND +63.99 FROM A 60-BIT WORD.	UNPAC3	4
		ENTRY UNPAC3	UNPAC3	5
0	1	UNPAC3	UNPAC3	6
1	5021000001	BS4	1	7
	5032000001	SA2	A1+1	8
2	5043000001	SA3	A2+1	9
	63110	SA4	A3+1	10
	63220	SB1	X1	11
3	63330	SB2	X2	12
	63440	SB3	X3	13
	56110	SB4	X4	14
4	5120000011 +	SA1	B1	15
	43324	SA2	BIAS	16
	20357	MX3	20	17
5	20157	LX3	47	18
	11013	LX1	47	19
	12762	BX6	X1*X3	20
	56720	BX7	X6+X2	21
6	20124	SA7	B2	22
	11013	LX1	20	23
	12762	BX6	X1*X3	24
	56730	BX7	X6+X2	25
7	20124	SA7	B3	26
	11013	LX1	20	27
	12762	BX6	X1*X3	28
	56740	BX7	X6+X2	29
10	0400000000 +	SA7	B4	30
11	17254000000000000000	EQ	UNPAC3	31
12		DATA	17254000000000000000	32
		END		

5-100

30100H SCM STORAGE USED
MODEL 76 ASSEMBLY31 STATEMENTS
0.018 SECONDS2 SYMBOLS
5 REFERENCES

PACF

COMPASS 3.0-577.

11/15/79 14.17.43.

PAGE

2

0
 1 5021 100001
 2 63110 63220 63330 50120
 3 56230 43041 11101 43033
 4 11200 20233 12712 56710
 5 0400000000 +
 6

IDENT PACF
 * THIS SUBROUTINE, WRITTEN IN COMPASS, PACKS 2 REAL NUMBERS INTO A
 * 32-BIT WORD
 ENTRY PACF
 PACF BSS 1
 SA2 A1+1
 SA3 A+1
 SB1 X1
 SB2 X2
 SB3 X3
 SA1 B2
 SA2 B3
 MX0 S3
 BX1 X0*X1
 MX0 Z1
 BX2 X0*X2
 LX2 Z1
 BX7 X1+X2
 SA7 B1
 EQ PACF
 END

PACF 2
 PACF 3
 PACF 4
 PACF 5
 PACF 6
 PACF 7
 PACF 8
 PACF 9
 PACF 10
 PACF 11
 PACF 12
 PACF 13
 PACF 14
 PACF 15
 PACF 16
 PACF 17
 PACF 18
 PACF 19
 PACF 20
 PACF 21
 PACF 22

30100H SCH STORAGE USED
 MODEL 76 ASSEMBLY

21 STATEMENTS
 0.013 SECONDS

1 SYMBOLS
 3 REFERENCES

UNPACF

COMPASS 3.0-347.

11/13/79 14.17.43.

PAGE

2

		IDENT UNPACF	UNPACF	
		* THIS SUBROUTINE, WRITTEN IN COMPASS, UNPACKS 2 REAL NUMBERS FROM	UNPACF	2
		* A 60-BIT WORD	UNPACF	3
		ENTRY UNPACF	UNPACF	4
0		UNPACF DSS 1	UNPACF	5
1	5021000001	SA2 A1+1	UNPACF	6
	5032000001	SA3 A2+1	UNPACF	7
2	63110	SB1 X1	UNPACF	8
	63220	SB2 X2	UNPACF	9
	63330	SB3 X3	UNPACF	10
	56510	SA5 B1	UNPACF	11
3	10655	BX6 X5	UNPACF	12
	56620	SA6 B2	UNPACF	13
	20641	LX6 J3	UNPACF	14
	10760	BX7 X6	UNPACF	15
4	56730	SA7 B3	UNPACF	16
	0400000000 +	EQ UNPACF	UNPACF	17
5		END	UNPACF	18

30100H SCH STORAGE USED
MODEL 76 ASSEMBLY

18 STATEMENTS
0.012 SECONDS

1 SYMBOLS
3 REFERENCES

		IDENT INBIN	INBIN	2
		* THIS SUBROUTINE, WRITTEN IN COMPASS, PACKS FOUR INTEGER NUMBERS INTO	INBIN	3
		* A 60-BIT WORD, ONE NUMBER AT A TIME.	INBIN	4
		ENTRY INBIN	INBIN	5
0		INBIN BSS 1	INBIN	6
1	5021000001	SA2 A1+1	INBIN	7
	5032000001	SA3 A2+1	INBIN	8
2	63110	SB1 A1	INBIN	9
	63220	SB2 A2	INBIN	10
	63330	SB3 A3	INBIN	11
	26120	SA1 02	INBIN	12
3	20117	LA1 15	INBIN	13
	50230	SA2 03	INBIN	14
	12612	RX6 X1+X2	INBIN	15
	56010	SA6 01	INBIN	16
4	0400000000 +	ED INBIN	INBIN	17
5		END	INBIN	18

50100H SCH STORAGE USED
MODEL 76 ASSEMBLY

17 STATEMENTS
0.011 SECONDS

1 SYMBOLS
3 REFERENCES

```

IDENT FILBIN
* THIS SUBROUTINE, WRITTEN IN COMPASS, PACKS FOUR INTEGER NUMBERS INTO
* A 60-BIT WORD IN A SINGLE OPERATION.
ENTRY FILBIN
FILBIN BSS 1
1 5021000001 1 SA2 A1+1
5032000001 SA3 A2+1
2 5043000001 SA4 A3+1
5054000001 SA5 A4+1
3 63110 SB1 X1
63220 SB2 X2
63330 SB3 X3
63440 SB4 X4
4 63550 SB5 X5
56120 SA1 B2
20117 LX1 B5
56230 SA2 B3
5 12512 RX5 X1+X2
20517 LX5 B5
56340 SA3 B4
12653 RX6 X5+X3
6 20617 LX6 B5
56450 SA4 B5
12764 RX7 X6+X4
56710 SA7 B1
7 0400000000 + EQ FILBIN
10 END

```

```

FILBIN 2
FILBIN 3
FILBIN 4
FILBIN 5
FILBIN 6
FILBIN 7
FILBIN 8
FILBIN 9
FILBIN 10
FILBIN 11
FILBIN 12
FILBIN 13
FILBIN 14
FILBIN 15
FILBIN 16
FILBIN 17
FILBIN 18
FILBIN 19
FILBIN 20
FILBIN 21
FILBIN 22
FILBIN 23
FILBIN 24
FILBIN 25
FILBIN 26
FILBIN 27
FILBIN 28

```

501008 SCH STORAGE USED
MODEL 76 ASSEMBLY

27 STATEMENTS
0.016 SECONDS

1 SYMBOLS
3 REFERENCES

```

IDENT MTBIN
* THIS SUBROUTINE, WRITTEN IN COMPASS, UNPACKS FOUR INTEGER NUMBERS
* FROM A 60-BIT WORD IN A SINGLE OPERATION.
ENTRY MTBIN
    ITOIN  DSS  1
    SA2    A1+1
    SA3    A2+1
    SA4    A3+1
    SA5    A4+1
    SB1    X1
    SB2    X2
    SB3    X3
    SB4    X4
    SB5    X5
    SA1    B1
    MX2    15
    LA2    15
    MU      MACKO  BREG
    BA6    X1*X2
    SA6    BREG
    AX1    15
    ENDM
    MO     B5
    MO     B4
    MO     B3
    MO     B2
    EQ     MTBIN
    END

    0  5021000001  1
    1  5021000001  5032000001
    2  5043000001  5054000001
    3  63110      63220
           63330      63440
    4  63550      63110
           43217      20217
           MU
    5  11612      11612
           11612
           11612
    10 0400000000 +
    11

13500B LCM 30100B SCH STORAGE USED
MODEL 76 ASSEMBLY
40 STATEMENTS
0.020 SECONDS
1 SYMBOLS
3 REFERENCES

```

```

MTBIN  2
MTBIN  3
MTBIN  4
MTBIN  5
MTBIN  6
MTBIN  7
MTBIN  8
MTBIN  9
MTBIN 10
MTBIN 11
MTBIN 12
MTBIN 13
MTBIN 14
MTBIN 15
MTBIN 16
MTBIN 17
MTBIN 18
MTBIN 19
MTBIN 20
MTBIN 21
MTBIN 22
MTBIN 23
MTBIN 24
MTBIN 25
MTBIN 26
MTBIN 27
MTBIN 28
MTBIN 29

```

DISTRIBUTION

Volumes II and III

Pacific Northwest Laboratory (2)
P.O. Box 999
Richland, WA 99352

CA Oster (2)

HEDL (8)

DG Doran	W/A-57
FA Garner	W/A-57
NJ Graves (2)	W/A-59
HL Heinisch	W/A-57
JO Shiffgens (2)	W/C-35
RL Simons	W/A-57