CANADIAN THESES ON MICROFICHE

THÈSES CANADIENNES SUR MICROFICHE

National Library of Canada
Collections Development Branch

Canadian Theses on
Microfiche Service

Ottawa, Canada
K1A 0N4

Bibliothèque nationale du Canada
Direction du développement des collections

Service des thèses canadiennes
sur microfiche

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED

LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE

NL 339 (r. 86/09)

Canada

AB INITIO CALCULATIONS OF THE PROPERTIES OF INDIUM ANTIMONIDE AND INDIUM ARSENIDE

by

David Singh

Submitted to the School of Graduate Studies in

partial fulfillment of the requirements

for the degree of Ph.D. in Physics

Department of Physics

University of Ottawa

Ottawa, Canada

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

"The underlying physical laws necessary for the mathematical theory of a large part of physics and the whole of chemistry are thus completely known, and the difficulty is only that the exact application of these laws leads to equations much too complicated to be soluble. It therefore becomes desirable that approximate practical methods of applying quantum mechanics should be developed, which can lead to an explanation of the main features of complex atomic systems without too much computation."

P.A.M. Dirac

Proc. Roy. Soc. (London) 123, 714 (1929)

ABSTRACT

In this thesis ab initio pseudopotential calculations of structural properties and valence charge densities in InSb are reported both under normal conditions and at high pressures. The method is also applied to calculate valence charge densities in the related compound InAs. The results thus obtained are discussed in relation to the high pressure semiconductor-metal phase transitions.

In addition a calculation of second sound velocities for a number of hexagonal crystals is reported and related to experimental data on solid Helium. It is found that a previously noted discrepancy between the experimental and theoretical velocities in this material can be significantly reduced by using a different set of elastic constants.

The convergence of the special points method for Brillouin zone averages, which was used in the pseudopotential calculations, is studied in the context of self-consistent phonon calculations for rare gas crystals, the method being found to converge rapidly. Lastly a simple intuitive method for evaluating Madelung constants in layer crystals is given. The method is applied to the transition metal dichalcogenides and in particular to TaS_2.

ACKNOWLEDGEMENTS

I would like to use this space to sincerely thank my supervisor, Dr. Y.P. Varshni, without whose support and encouragement the work described here could not have been undertaken.

I would like also to thank Dr. H. Glyde who originally made the suggestion that it would be worthwhile for me to try a pseudopotential method on covalently bonded materials and Dr. L.K. Moleko with whom I discussed the self consistent phonon theory where the convergence of the special points scheme was tested. I also want to express my appreciation to Dr. Moleko for making it a pleasure to share an office with him. I would also like to thank Dr. K.S. Song, Dr. B. Joos and Dr. G.C. Aers for the extremely helpful conversations I have had with them and the suggestions they have made.

Thanks are also due to NSERC for a postgraduate scholarship which I held for some of the time this work was in progress and to the University of Ottawa for making its facilities available. The staff of the physics department have gone out of their way to make my stay here as pleasant as possible and I wish here to thank them all, and in particular Lorraine Johnston who typed the equations in this thesis.

Finally and most importantly I would like to thank my parents for without their support this work would not have been accomplished.

CONTENTS

INTRODUCTION

The bonding and chemical nature of semiconductors has been a focal point of interest to solid state physicists for some time (1). - An important motivation has been to be able to calculate the properties of bulk semiconductors from a microscopic point of view in order to obtain a better understanding of the physical and chemical properties of these materials. The open 4-fold coordinated structure of these materials and the importance of chemical bonding in determining their properties makes an understanding of these materials in terms of the distribution of electronic charge in them particularly attractive. Although the work described here was undertaken for this reason it is worth noting that with the increasing commercial interest in novel structures such as strained layers (eg. Germanium Silicon alloys on Silicon) and quantum well structures the interest in the nature of the bonding in semiconductors has become more than academic, the ability to reliably predict and understand the properties of semiconductor structures being of great importance.

InSb is an interesting material with potentially important device applications (2). Over a relatively small pressure and temperature range it is found to occur in a number of structures. In particular InSb occurs in zinc blende, body centered tetragonal and two orthorhombic structures in the range 0-400 degrees Celsius and 0-5 gigapascals (3). Amorphous InSb has been observed to transform to a crystalline NaCl type structure under pressure at room temperature and to remain in this phase at atmospheric pressure (4). It is intriguing that this structure cannot be obtained from crystalline InSb by applying pressure. The tetragonal phase is also metastable at atmospheric pressure and low temperatures. The extent of this polymorphism suggests that

InSb is a borderline material with considerable instability in its bonding. It seems reasonable that worthwhile insights into the nature of covalently bonded materials may be obtained from the study of such borderline substances and particularly their similarities with and differences from prototypical materials such as Silicon. Here structural properties of InSb are studied in relation to its electronic charge distribution both under normal conditions and also under pressure.

The theoretical basis for the calculation is reviewed in Chapter 1, the methods used being discussed in detail in Chapter 2. A brief overview of the implementation of the total energy calculations is given in Appendix A. In Chapter 3 valence charge densities of InSb and the related material, InAs, are given both under normal conditions and also at reduced volumes corresponding to high pressures. These are discussed in terms of the high pressure semiconductor-metal phase transition and recent X-ray scattering measurements. Extending the work described in Chapter 3 polarization charge densities for a variety of distortions are reported in Chapter 4 along with calculated structural properties again at the free volume as well as at reduced volumes.

In Chapters 5, 6 and 7 some work in solid state physics not directly related to the above problem is reported. In Chapter 5 a calculation of second sound velocities in hexagonal crystals is described and it is shown that a discrepancy between the experimental and theoretical values for ^{4}He pointed out by Maris can be reduced if another set of elastic constants is used. In Chapter 6 the convergence of the special points method is studied for the evaluation of Brillouin zone averages used in self-consistent phonon calculations. It is found that the method converges rapidly in the cases studied. Finally in Chapter 7 formulae are derived for the evaluation of Madelung constants in layer crystals. These are applied to the transition

metal dichalcogenides. It is found that the results of Shen and Liang for the cation charges in TaS_2 can be significantly improved by using our more accurate Madelung constants.

REFERENCES

(1) See, for example, W.A. Harrison, Electronic structure and the properties of solids, (W.H. Freeman, San Francisco, 1980) or J.C. Phillips, Bonds and bands in semiconductors, (Academic Press, New York, 1973)

(2) See, for example, C.T. Seaton, F.A.P. Tooley and S.D. Smith, Second European Conference on Integrated Optics, Florence Italy, October 1983 (IEE, London England, 1983) page 102

(3) S.-C. Yu, I.L. Spain and E.F. Skelton, J. Appl. Phys. 49, 4741 (1978) and references therein

(4) K. Asaumi, O. Shimomura and S. Minomura, J. Phys. Soc. Japan 41, 1630 (1976)

CHAPTER 1

Theoretical Background

A) Introduction

In solid state physics it has been a long standing goal to be able to calculate bulk properties of solids without resorting to empirical models. It was realized that one way to do this was to calculate the total energy of a solid under various conditions and thereby derive its response (1-3). From such calculations it is possible to derive a large amount of information. For example, it is possible to calculate the equilibrium structures of solids and the stability of different phases at T=0. By applying static external influences the elastic properties and other responses can be determined. Moreover, within the adiabatic approximation lattice dynamical properties such as phonon frequencies, anharmonic terms and phonon-phonon interactions can be calculated (4). The ability to calculate such anharmonic terms is of considerable importance since they are usually not directly measurable and cannot be reliably obtained using phenomenological models. For some time interatomic potentials have been used to calculate various properties of simple solids such as rare gas crystals. This work has been reviewed in detail elsewhere (5). To the extent that the atoms in a solid can be regarded as unaffected by the environment, if the potentials used are derived from atomic calculations, these results may be regarded as being obtained by an ab initio method.

However, in more complex solids such as semiconductors one would not expect such an approach to succeed. In the case of semiconductors it was recognized that a successful calculation would likely involve consideration of the electronic structure as opposed to regarding the solid as a collection of

atoms which interact amongst themselves as separate units.

While it was realized that in principle by using quantum mechanics the total energy of a system of electrons and ions could be calculated (1) until the late 1970's it was not recognized that such work might be feasible for semiconductors. The reasons why such calculations were regarded as impractical are several. First of all it is important to recognize that the changes in the total energy of semiconductors associated with, for example, structural changes or phonons are very small in relation to the total energy. This point is illustrated by some typical precisions that might be required to evaluate some properties from an energy calculation (see Table 1-1). In comparison typically the total energy might be of the order of 10 Ry/atom in a pseudopotential calculation and 1000 Ry/atom in an all electron calculation. Thus it is easy to see why it was not at all clear that the changes of interest would not be lost in numerical 'noise' were such a calculation undertaken. Special considerations for semiconductors include the following:

(1) The charge density in these materials is highly nonspherical; in fact covalent bonds may be characterized by the buildup of charge in the bonding region.

(2) The valence electrons interact fairly strongly with the ion cores compared with the situation in simple metals.

(3) Both exchange and correlation (which is important in the bonds, i.e., in the region which is expected to be of interest) are important in these materials. To illustrate the importance of correlation, one may examine the case of Silicon (6) for which the Hartree-Fock band structure in which exchange is treated exactly is even worse than the results of SCF Hartree calculations which completely ignore exchange. The failure of the Hartree-Fock method arises from its neglect of correlation and subsequent overweighting of ionic configurations. In fact even for

Diamond the Hartree-Fock method is not entirely satisfactory (7).

TABLE 1-1. Precision Requirements for Total Energy Calculations

Quantity Desired	Precision Required (Ry/atom)
Cohesive energy	0.01
Lattice constant	0.001
Bulk modulus	0.001
Phonon eigenvectors	0.0001
Phonon anharmonicity	0.0001

B) Density Functionals and the Local Density Approximation

The key ingredient in treating the electronic structure of a covalent material is a way to include correlation in the calculation. Until a scheme could be devised for doing this calculations of the total energy in semiconductors could not be reliably accomplished. The breakthrough came in 1964 with the development of the density functional formalism by Hohenberg, Kohn and Sham (8,9). They showed:

(1) All aspects of the electronic structure of a nondegenerate ground state are completely determined by its density, n(r). That is that all physical properties of the ground state are functionals of its density and in particular the total energy is a functional of the density, E(n).

(2) For a given applied potential the true density is that which minimizes the total energy, E(n). This property is known as the variational principle. In addition to establishing a way to determine the density the variational principle has an important consequence. An error in the charge density results in only a second order error in the energy.

The above is known as the Hohenberg-Kohn theorem. This result is particularly elegant because it shows that any physically interesting property

of the ground state can in principle be determined from an observable as opposed to a wave function which is not observable and which is to some degree arbitrary.

In 1965 Kohn and Sham (9) showed that the formalism could be expressed in terms of effective one particle equations.

$$[-\tfrac{1}{2}\nabla^2 + v(r) + V_H(n,r) + v_{xc}(n,r)]\,\phi_i = \varepsilon_i\,\phi_i \tag{1-1}$$

$$n(r) = \sum_i |\phi_i(r)|^2 \tag{1-2}$$

$$v_{xc} = \frac{\delta \mathcal{E}_{xc}}{\delta n(r)} \tag{1-3}$$

Where the sum in Equation (1-2) is over the occupied orbitals.

$$E(n) = T(n) + \int v(r)\, n(r)\, dr + \tfrac{1}{2}\iint \frac{n(r)\,n(r')}{|r - r'|}\, drdr' + \mathcal{E}_{xc}(n) \tag{1-4}$$

Equations (1-1)-(1-4) are the Kohn-Sham equations in atomic units. Equation (1-3) defines the exchange-correlation potential, $v_{xc}(n)$, in terms of the corresponding energy functional defined by Equation (1-4). Here E(n) is the total energy functional, T(n) is the kinetic energy, v is the external (to the electron system) potential and $V_H(n)$ is the Hartree potential. The total energy is given by

$$E(n) = \sum_i \varepsilon_i - \tfrac{1}{2}\iint \frac{n(r)\,n(r')}{|r - r'|}\, drdr' + \mathcal{E}_{xc}(n) - \int v_{xc}(n)\, n(r)\, dr \tag{1-5}$$

The above represents an exact formulation of the many particle problem. While the Kohn-Sham equations may not seem much more complicated than the Hartree SCF equations, in fact the many body problem is 'hidden' in the exchange-correlation potential which is an unknown functional of the charge density. Determining it would involve an exact solution of the many body problem. The utility of the formulation lies in the possibility of obtaining

accurate results using approximate functionals. The most commonly used method revolves around the local density approximation (LDA). The LDA consists of replacing the exchange correlation energy by the same local function of the density in all cases. The function used is that for the homogeneous electron gas which is for practical purposes known from Monte Carlo simulations (10). This approximation was first proposed in 1965 by Kohn and Sham in the paper they derived the Kohn-Sham equations in (9), although its origins can be traced back to the Thomas-Fermi approach and to Slater's local exchange potential (11).

Although solving the Kohn-Sham equations in the LDA is from a computational point of view not much more difficult than a Hartree calculation, in many cases the results are much better than the results of Hartree-Fock calculations which for solids are extremely time consuming. For the electron gas the LDA is exact by construction.

In general one would expect the LDA to work best in situations characterized by a nearly uniform charge density and to be unreliable in situations where the charge density fluctuates significantly on a scale of the reciprocal of the Fermi wavevector. In fact, the cases of interest here, atoms, molecules and solids, are in the latter category. In spite of this, good results have been obtained using the LDA in these cases.

Arguments explaining this unexpectedly good performance of the LDA have been given by Gunnarsson and Lundqvist (12). The explanation can be summarized as follows:

(1) The exchange correlation energy depends on a spherical average over the exchange correlation hole.

(2) In the LDA the charge excluded from the exchange correlation hole is exact.

(3) The LDA gives a fairly good description of the spherically averaged exchange correlation hole (13). In particular its size varies with density in the correct way.

These arguments unfortunately fall short of a demonstration that the LDA will give good results in all solid state applications. Perhaps the best reason to expect the approximation to be a good one is the large number of calculations in diverse situations for which it has been successful. These include calculations on bulk semiconductors, semiconductor surfaces, interfaces, metals, molecules and defects (13-30). For semiconductors ab initio calculations based on the use of the LDA typically yield lattice parameters, bulk moduli and phonon frequencies within a few percent of experimental values. This is a remarkable result in view of the fact that the only inputs to these calculations are atomic numbers.

It should be noted that in spite of its successes the LDA is not a panacea. First of all the density functional formalism is useful only for the ground state. Secondly, excitation energies and one-electron removal energies are not correctly described in the density functional formalism (31-36) and the LDA does not lead to accurate band structures (though it does normally yield the correct topology (37)). In particular the LDA tends to underestimate band gaps. This results from an incomplete cancellation of the self-interaction terms between the Coulomb interaction and the approximate exchange term in the Kohn-Sham equations (31) which yields a raising of the valence band eigenvalues relative to the conduction bands which are less localized. This effect is most important in systems with highly localized states such as rare gas solids, ionic crystals, and transition metals. The LDA is not adequate for calculating the band gaps of semiconductors typically underestimating them by about 40%.

C) Ab Initio Norm Conserving Pseudopotentials

The second key ingredient in making total energy calculations in semiconductors feasible came into being with the development of ab initio norm conserving pseudopotentials in 1979 (38-43). These enabled the carrying out of total energy calculations without explicitly treating the ion cores. Treating the cores explicitly in most systems would have resulted in a many fold increase in the computation required without contributing anything to the results. This is because most physically interesting properties of semiconductors are related to the arrangement of the valence electrons. Moreover, full nuclear potentials are very hard, that is plane wave matrix elements with these do not decrease rapidly with increasing wave vectors. This is in contrast with the situation using norm conserving pseudopotentials which are normally constructed to have a soft core. Thus the development of these pseudopotentials made it possible to use plane wave bases in ab initio calculations. This is of importance because although 'more efficient' basis sets exist they are in general optimal for a given structure. Thus in examining the changes in total energy with various distortions they would tend to yield unreliable results. For example while a basis of s and p localized orbitals might give good results in an undistorted tetrahedrally coordinated solid it would very likely yield very stiff bond bending force constants because of the lack of variational freedom in accommodating that distortion. This is in contrast to the situation using a plane wave basis which would likely require more computation to treat the undistorted solid but which could yield reliable results in the distorted solid as well. In fact it seems likely that in the case of a semiconductor with distortions, in order to obtain the same variational freedom using a localized basis as obtained with plane waves approximately the same number of basis functions may be required. This is related to the fact that the valence charge densities of the solid and

the free atoms are of similar magnitude suggesting that the nearly free electron and linear combination of atomic orbitals (LCAO) treatments are equally valid (or invalid) in these materials. Thus ab initio norm conserving pseudopotentials offer an advantage even in cases such as Hydrogen where treating the core presents no difficulty. Ho, Tao and Zhu (30) have recently treated NbH using norm conserving pseudopotentials with no localized basis functions at the Hydrogen atoms (they used localized d orbitals at the Nb sites to facilitate treatment of the highly localized d states in the transition metal).

The basic idea of a pseudopotential is to include the orthogonalization of the valence electron wavefunctions to the core by transforming the Hamiltonian. In this way the core electrons are included in an effective way without treating them explicitly. Thus all interactions between the core and valence electrons are transferred to the pseudopotential. The use of pseudopotentials can be traced back at least 50 years (44).

The procedure for generating these ab initio pseudopotentials is as follows:

(1) Do an all electron density functional calculation on the atomic system in the configuration desired.

(2) Construct a pseudopotential that gives the same lowest valence eigenvalues as the all electron calculation, and the same wavefunction outside a chosen core radius, r_c. These pseudopotentials are constructed to be norm conserving in that even when normalized the wavefunctions they yield are identical to those of the all electron calculation outside the core radius. Norm conservation is crucial in making these pseudopotentials transferable.

Hamann, Schluter and Chiang (38) showed that the above procedure leads to

pseudopotentials with two desirable features. The scattering properties of the all electron atoms are reproduced with minimum error and the electrostatic potential produced outside r_c is identical for the all electron and pseudocharge distributions. It should be noted that these pseudopotentials are not unique and that some schemes for generating them yield softer potentials than others. Moreover they are non-local, that is ℓ dependent.

Recently complete sets of norm conserving ab initio pseudopotentials have been published (45,46). These sets are expanded in error functions and Gaussians. As a result it is possible to evaluate the plane wave matrix elements analytically although the expressions that result are somewhat lengthy.

It should be noted that aside from the frozen core approximation, the use of pseudopotentials in a LDA calculation builds in the assumption that the overlap between the valence and core charge densities is small. This is because the exchange correlation potential is nonlinear. Thus if there is a significant overlap between the charge densities an error results from the implied linearization in using pseudopotentials, thereby reducing their transferability. This is not important in most materials with the exception of those containing 3d transition metal atoms. In these atoms the 3d electrons are highly localized in the core region. One scheme for dealing with this is to carry a representation of the core charge density along in the pseudopotential calculation using it in determining the exchange correlation potential (42,46). An alternate scheme is to treat both the outer-core and valence electrons electrons explicitly using a pseudopotential for the inner-core electrons (43). This scheme involves more computational effort but may be superior in cases where the outer-core contributes to the bonding.

D) The Special Points Method for Brillouin Zone Averages

In a density functional calculation the charge density must be evaluated in each iteration of the Kohn-Sham equations. This charge density is defined as a sum over the occupied states which implies a Brillouin zone average. Moreover at each point in the sum the Hamiltonian matrix must be diagonalized. Because of the large number of basis functions required for convergence in a calculation based on plane waves (normally more than 100 per atom in the unit cell) it is crucial to evaluate the sum as efficiently as possible.

The special points method (47-49) is a highly efficient method for evaluating Brillouin zone averages. Using this method it has proved possible to accurately evaluate the charge densities in semiconductors (50) using a small number of points in the Brillouin zone (typically 2-10, although in some cases just one point may be adequate (47)). Singh and Varshni (51) have studied the scheme's convergence in the context of self-consistent phonon formalism and found that in the cases studied the convergence is equally rapid. This work is described in Chapter 6.

The basic idea behind the special points method is to replace a Brillouin zone average by a weighted average over a few representative q points in the zone. In this way the evaluation of the function to be averaged at a large number of points is avoided as is the explicit use of an interpolation formula. Since the method is discussed in detail elsewhere (47,48) here only the main results are given.

A function, f(q), with the complete symmetry of the lattice can be expanded in symmetrized plane waves

$$f(q) = f_o + \sum_{m=1}^{\infty} f_m A_m(q) \qquad (1\text{-}6)$$

where

$$A_m(q) = \sum_j \exp(iq \cdot R_{jm}) \tag{1-7}$$

where the sum in equation (1-7) is over the m'th set of equivalent lattice vectors. The average of f(q) over the Brillouin zone is f_o. A set of special points is a set of n points, q_i, with weights, w_i, such that

$$\sum_{i=1}^{n} w_i A_m(q_i) = 0 \qquad m = 1, \ldots N \tag{1-8}$$

$$\sum_{i=1}^{n} w_i = 1 \tag{1-9}$$

If N is sufficiently large it may be expected that the weighted average of f(q) should be a good approximation to the average since the first N terms in the sum of equation (1-6) cancel while for a smooth function the coefficients f_m may be expected to decrease with increasing m. The method can also be applied in a direct way to functions lacking the complete symmetry of the lattice as discussed in reference 49. This is done by constructing a symmetric function with the same average. For electron densities this is done using information about the space group of the crystal in question. We note that the choice of the representative points does not depend on the function, f, although the convergence of the procedure depends critically on f. In particular the procedure does not converge rapidly for the charge density of metals because in this case the function to be averaged has a sharp edge (the Fermi surface) which gives rise to large Fourier components, f_m, even for large m.

REFERENCES

(1) R.P. Feynman, undergraduate thesis, MIT, unpublished (1938)

(2) See, for example, C. Kittel, Introduction to Solid State Physics, fourth edition, (Wiley, New York, 1971), Chapter 3

(3) See, for example, J.M. Ziman, Principles of the Theory of Solids, second edition (Cambridge University Press, Cambridge, 1972) p. 129 ff, also p. 204

(4) V. Heine and D. Weaire, in Solid State Physics, vol. 24, eds. H. Ehrenreich et al., (Academic Press, New York, 1970) p. 366

(5) M.L. Klein and J.A. Venables, eds., Rare Gas Solids, (Academic Press, New York, 1976), and M.P. Tosi in Solid State Physics, vol. 16, p. 1; eds. F. Seitz and D. Turnbull, (Academic Press, New York, 1964)

(6) W. Hanke, unpublished (1984); W. Hanke, Th. Gozler and H.J. Matausch, Solid State Commun. 51, 23 (1984)

(7) G. Strinati, H.J. Mattausch and W. Hanke, Phys. Rev. B 25, 2867 (1982); R.N. Euwema, D.L. Wilhite and G.T. Surratt, Phys. Rev. B 7, 818 (1973)

(8) P. Hohenberg and W. Kohn, Phys. Rev. 136, B864 (1964)

(9) W. Kohn and L.J. Sham, Phys. Rev. 140, A1133 (1965), and L.J. Sham and W. Kohn, Phys. Rev. 145, B561 (1966)

(10) C.M. Ceperly and B.J. Alder, Phys. Rev. Lett. 45, 566 (1980)

(11) J.C. Slater, Phys. Rev. 81, 385 (1951)

(12) O. Gunnarsson and B.I. Lundqvist, Phys. Rev. B 13, 4274 (1976), see also S. Lundqvist and N.H. March, eds., Theory of the Inhomogeneous Electron Gas, (Plenum Press, New York, 1983)

(13) A.R. Williams and U. Von Barth, in Theory of the Inhomogeneous Electron Gas, (Plenum Press, New York, 1983), Chapter 4

(14) V.L. Moruzzi, J.F. Janak and A.R. Williams, Calculated Electronic Properties of Metals, (Pergamon, New York, 1978) and references therein

(15) V. Heine, in Solid State Physics, vol. 35, p. 1, eds. F. Seitz and D. Turnbull, (Academic Press, New York, 1980)

(16) M.L. Cohen, Proc. 15th Int. Conf. on the Physics of Semiconductors, J. Phys. Soc. Japan 49, 13 (1980)

(17) See the papers in Ab Initio Calculation of Phonon Spectra, ed. J.T. Devreese (Plenum, New York, 1983) and Electronic Structure, Dynamics and Quantum Structural Properties of Condensed Matter (1985)

(18) J.E. Northrup, M.T. Yin and M.L. Cohen, Phys. Rev. A 28, 1945 (1983)

(19) J.L. Martins and W. Andreoni, Phys. Rev. A 28, 3637 (1983)

(20) I.P. Batra, Phys. Rev. B 29, 7108 (1984)

(21) S. Froyen and M.L. Cohen, Phys. Rev. B 28, 3258 (1983)

(22) G.A. Baraff and M. Schluter, Phys. Rev. B 30, 1853 (1984)

(23) M.T. Yin, Phys. Rev. B 30, 1773 (1984)

(24) R. Biswas, R.M. Martin, R.J. Needs and O.H. Nielsen, Phys. Rev. B 30, 3210 (1984)

(25) M.Y. Chou, S.G. Louie, M.L. Cohen and N.A.W. Holzwarth, Phys. Rev. B 30, 1062 (1984)

(26) D. Vanderbilt, S.G. Louie and M.L. Cohen, Phys. Rev. Lett. 53, 1477 (1984)

(27) K.-M. Ho, C.L. Fu and B.N. Harmon, Phys. Rev. B 29, 1575 (1984)

(28) A.A. Mbaye, C. Verie and F. Aymerich, Phys. Rev. B 29, 3756 (1984)

(29) J.E. Northrup, Phys. Rev. Lett. 53, 683 (1984)

(30) K.-M. Ho, H.-J. Tao and X.-Y. Zhu, Phys. Rev. Lett. 53, 1586 (1984)

(31) J.P. Perdew and A. Zunger, Phys. Rev. B 23, 5048 (1981) and references therein; also M.R. Norman, Phys. Rev. B 29, 2956 (1984)

(32) W. Hanke and L.J. Sham, Phys. Rev. B 21, 4656 (1980)

(33) R.A. Heaton and C.C. Lin, J. Phys. C 17, 1853 (1984), and references therein

(34) L.J. Sham and W. Kohn, Phys. Rev. 145, A561 (1966)

(35) L. Hedin and B.I. Lundqvist, J. Phys. C 4, 2064 (1971)

(36) C.-O. Almbladh and A.C. Pedroza, Phys. Rev. A 29, 2322 (1984)

(37) J.E. Jaffe and A. Zunger, Phys. Rev. B 28, 5822 (1983)

(38) D.R. Hamann, M. Schluter and C. Chiang, Phys. Rev. Lett. 43, 1494 (1979)

(39) L. Kleinman, Phys. Rev. B 21, 2630 (1980)

(40) G.P. Kerker, J. Phys. C 13, L189 (1980)

(41) G.B. Bachelet and M. Schluter, Phys. Rev. B 25, 2103 (1982)

(42) S.G. Louie, S. Froyen and M.L. Cohen, Phys. Rev. B 26, 1738 (1982)

(43) D.M. Bylander and L. Kleinman, Phys. Rev. B 29, 2274 (1984)

(44) E. Fermi, Nuovo Cimento II, 157 (1934)

(45) G.B. Bachelet, D.R. Hamann and M. Schluter, Phys. Rev. B 26, 4199 (1982); erratum: Phys. Rev. B 29, 2309 (1984)

(46) H.S. Greenside and M. Schluter, Phys. Rev. B 28, 535 (1983)

(47) A. Baldereschi, Phys. Rev. B 7, 5212 (1973); Bull. Am. Phys. Soc. 17, 237 (1972)

(48) D.J. Chadi and M.L. Cohen, Phys. Rev. B 8, 5747 (1973)

(49) H.J. Monkhorst and J.D. Pack, Phys. Rev. B 13, 5188 (1976)

(50) E. Holzschuh, Phys. Rev. B 28, 7346 (1983)

(51) D. Singh and Y.P. Varshni, Phys Rev. B 30, 6917 (1984)

CHAPTER 2

Implementation of the Ab Initio Method

A) Introduction

In this chapter implementation of the LDA pseudopotential calculations for zinc blende structure semiconductors is discussed. The main equations used are given and some of the computational considerations are discussed.

B) Formulation in Terms of a Plane Wave Basis

The equations giving the total energy of a crystal in the context of a pseudopotential calculation using the LDA have been presented elsewhere in detail (1,2). Here the main results are restated.

In order to use a plane wave basis the pseudopotential V(r) must be divided into short and long range components. This is because the long ranged part must be treated differently as its Fourier transform diverges at G=0. For a physical pseudopotential the long ranged part is local. That is for an ℓ-dependent (non-local) pseudopotential all the angular momentum components have the same form for large r. The pseudopotential is divided as follows

$$W(r) = V_L(r) + \sum_{\ell=0}^{\infty} V_\ell(r)\ \hat{P}_\ell \qquad (2\text{-}1)$$

where $\hat{P}_\ell$ are the angular momentum projection operators, V_L is the local component of the pseudopotential with the V_ℓ's being the non-local components.

In a plane wave basis the pseudo Hamiltonian matrix elements for the Kohn-Sham equations at wavevector k in the Brillouin zone are given by:

$$H(k,G,G') = V_H(G'-G)+v_{xc}(G'-G) + \sum_\tau S_\tau(G'-G)$$

$$\times [V_{L\tau}(G'-G) + \sum_\ell V_{\ell\tau}(k+G, k+G')] \qquad G \neq G' \qquad (2\text{-}2)$$

$$H(k,G,G') = \frac{(k+G)^2}{2} + \sum_\tau \sum_\ell V_\ell(k+G, k+G') \qquad G = G' \qquad (2\text{-}3)$$

where V_H is the Hartree potential and v_{xc} is the exchange correlation potential. In the G=G' matrix elements the Hartree, exchange correlation and local pseudopotential terms have been set to zero. Although the Hartree and pseudopotential matrix elements diverge at G=G' the result of setting them to zero is a constant finite shift in the eigenvalues because of the overall electrical neutrality of the crystal. The expression for the total energy per unit cell is

$$E = \sum_i \langle\varepsilon_i\rangle_k - V_c \sum_{G\neq 0} [V_H^{in}(G) + v_{xc}^{in}(G)]\, n^{out}(G) +$$
$$\tfrac{1}{2} V_c \sum_{G\neq 0} V_H^{out}(G)\, n^{out}(G) + V_c \sum_G \varepsilon_{xc}^{out}(G)\, n^{out}(G) +$$
$$\gamma_{Ewald} + \left(\sum_\tau \alpha_{1\tau}\right)\left(\sum_\tau Z_\tau\right) \qquad (2\text{-}4)$$

Where V_c is the volume of the unit cell, Z_τ is the charge on core τ, γ_{Ewald} is the Ewald energy of the cores in a uniform compensating background and

$$\alpha_{1\tau} = \frac{1}{V_c} \int (V_{L\tau} + Z_\tau/r)\, d^3r \qquad (2\text{-}5)$$

arises from the neglect of the $V_L(G=0)$ terms in the Hamiltonian. In equation (2-4) the superscripts in and out refer to the input and output of a cycle of the self-consistent loop. Although at the end of a true self-consistent calculation the inputs and outputs would be identical, the distinction has been retained in order to show which terms cancel self interactions in the eigenvalue sum. Chelikowsky and Louie (2) have shown that for a charge density close to the self-consistent charge density retaining the distinction

between inputs and outputs as in equation (2-4) ensures that the error in the energy is second order in the charge density error. Thus by retaining the distinction the energy can be made to converge much more rapidly in the self-consistent calculation. In order to calculate the exchange correlation energy the charge density was evaluated at about 12,000 points in the unit cell. At each point the exchange correlation energy was evaluated and thus the required average was obtained.

C) Pseudopotentials

The set of pseudopotentials used here are those tabulated by Bachelet, Hamann and Schluter (3). The exchange correlation energy used by them was generated from Monte Carlo simulations (4) and parameterized in Reference 5. In order to maintain consistency the same parameterization was used here. In this set the local component is expanded in error functions

$$V_L(r) = -\frac{Z_v}{r}\left[\sum_{i=1}^{2} c_i^{\text{core}} \operatorname{erf}\left[(\alpha_i^{\text{core}})^{\frac{1}{2}} r\right]\right] \qquad (2\text{-}6)$$

and the non-local components are expanded in Gaussians

$$V_\ell(r) = \sum_{i=1}^{3} (A_i^\ell + r^2 A_{i+3}^\ell) \exp(-\alpha_i r^2) \qquad (2\text{-}7)$$

Here Z_v is the number of valence electrons in the neutral atom and the c's, A's and α's are parameters. Because of degree of linear dependence among Gaussians, a transformed set of weights c was tabulated in Reference 3, instead of the A's. Tabulating the latter would have required the retention of large numbers of significant figures.

$$C_i = -\sum_{j=1}^{6} A_j Q_{ij} \qquad (2\text{-}8)$$

where Q is the orthogonality matrix,

$$Q_{ij} = \begin{cases} 0 & i > j \\ S_{ij} - \sum_{k=1}^{i-1} Q_{ki}^2 & i = j \\ [S_{ij} - \sum_{k=1}^{i-1} Q_{ki}\, Q_{kj}]/Q_{ii} & i < j \end{cases} \qquad (2\text{-}9)$$

and

$$S_{ij} = \int_0^\infty r^2\, \phi_i(r)\, \phi_j(r)\, dr \qquad (2\text{-}10)$$

with

$$\phi_i(r) = \begin{cases} e^{-\alpha_i r^2} & i = 1, 2, 3 \\ r^2\, e^{-\alpha_i r^2} & i = 4, 5, 6 \end{cases} \qquad (2\text{-}11)$$

Thus

$$S_{ij} = \begin{cases} \dfrac{\sqrt{\pi}}{4(\alpha_i+\alpha_j)^{3/2}} & i, j \le 3 \\ \dfrac{3\sqrt{\pi}}{8(\alpha_i+\alpha_j)^{5/2}} & i \text{ or } j > 3 \\ \dfrac{5\sqrt{\pi}}{16(\alpha_i+\alpha_j)^{7/2}} & i, j > 3 \end{cases} \qquad (2\text{-}12)$$

Equation (2-8) was inverted by factoring Q into a product of lower and upper triangular matrices and then performing back and forward substitutions. This algorithm was selected because of its high numerical stability which was important in view of the non-orthogonality of the basis. In terms of these pseudopotentials the required integrals can be evaluated analytically. They are

$$I_\alpha = \int_o^\infty r\ \mathrm{erfc}(ar)dr = 1/4a^2 \tag{2-13}$$

for the evaluation of α_1,

$$\frac{4\pi Z_v}{G\ V_c} \sum_{i=1}^{2} \int_o^\infty c_i^{core}\ \mathrm{erf}(\sqrt{\alpha_i}\ r)\ \sin(gr)dr = \frac{-4\pi Z_v}{G^2\ V_c} \sum_{i=1}^{2} c_i^{core}\ e^{-g^2/4\alpha_i} \tag{2-14}$$

for the local matrix elements and

$$V_\ell(k,k') = (2\ell+1)\ \frac{4\pi}{V_c} \int_o^\infty V_\ell(r)\ j_\ell(|k|r)\ j_\ell(|k'|r)\ r^2dr \tag{2-15}$$

$$x\ P_\ell(\cos\gamma)$$

where

$$\cos(\gamma) = \frac{k\cdot k'}{|k||k'|} \tag{2-16}$$

This requires the evaluation of two integrals

$$I_1 = \int_o^\infty \exp(-\alpha r^2)\ j_\ell(\beta r)\ j_\ell(\gamma r)\ r^2dr \tag{2-17}$$

$$I_2 = \int_o^\infty \exp(-\alpha r^2)\ j_\ell(\beta r)\ j_\ell(\gamma r)\ r^4dr \tag{2-18}$$

These are as follows.

$$I_1 = \frac{\pi}{4\alpha}\ e^{-(\beta^2+\gamma^2)/4\alpha} \frac{1}{\sqrt{\beta\gamma}}\ I_{\ell+\frac{1}{2}}\left(\frac{\beta\gamma}{2\alpha}\right) \tag{2-19}$$

where I_n is a modified Bessel function. These were evaluated using a recursion relation for arguments greater in magnitude than 1.0 and a series expansion for smaller arguments.

$$I_2 = \frac{2}{\pi\sqrt{\beta\gamma}}\ U_\ell(\beta,\gamma,\sqrt{a}) \tag{2-20}$$

where

$$U_\ell(a,b,p) = \frac{\Gamma(\ell+\frac{5}{2})}{2\pi p^4 \Gamma(2\ell+2)} \left(\frac{ab}{p^2}\right)^{\ell+\frac{1}{2}} \left\{\left[1 - \frac{(a^2+b^2)}{(\ell+\frac{3}{2})4p^2}\right] S_\ell(a,b,p) + \frac{2ab}{(\ell+\frac{3}{2})4p^2} T_\ell(a,b,p)\right\} \quad (2\text{-}21)$$

with

$$S_\ell(a,b,p) = e^{-(a^2+b^2)/4p^2} \sum_{i=0}^{\ell} \frac{(-1)^i\, i!\, (2i)!\, (\ell-i)!}{\ell!} F_i^- \left(\frac{ab}{2p^2}\right) \quad (2\text{-}22)$$

$$T_\ell(a,b,p) = e^{-(a^2+b^2)/4p^2} \sum_{i=0}^{\ell} \frac{(-1)^i\, i!\, (2i+1)!\, (\ell-i)!}{\ell!} F_{i+\frac{1}{2}}^+ \left(\frac{ab}{2p^2}\right) \quad (2\text{-}23)$$

and

$$F_i^{\pm}(x) = e^x \sum_{j=0}^{2i} \frac{(-1)^j\, x^{-(j+1)}}{(2i-j)!} \pm e^{-x} \sum_{j=0}^{2i} \frac{x^{-(j+1)}}{(2i-j)!} \quad (2\text{-}24)$$

Equation (2-24) was evaluated directly as written for arguments greater than 1.0 in magnitude. For smaller arguments it was evaluated by a power series expansion in the argument.

In this calculation spin-orbit interactions were neglected. While such an approximation is certainly not justified for a calculation of the band structure of a material such as InSb where spin-orbit splittings can be larger than the band gap (6) it may be expected to be reasonable in the context of the present calculation which is concerned with averaged properties i.e. charge densities and total energies and not the properties of states in a specific band at a specific point in the Brillouin zone. Recent relativistic calculations on SnTe and PbTe (7) tend to confirm this indicating that neglecting spin-orbit couplings is adequate for most purposes. Further support to this conclusion is provided by energy minimization calculations of

the surface structure of InSb using a tight binding method in which it was found that the structure of InSb surfaces is not affected very much by spin-orbit interactions (8). Errors resulting from this approximation are probably largest in the calculation of anharmonic force constants which are expected to be more dependent on the details of the band structure than harmonic quantities.

D) Ewald, Hartree and exchange correlation terms

In each iteration of the Kohn-Sham equations above the charge density, n(g), must be evaluated in order to calculate the Hartree and exchange correlation potentials. In a plane wave basis the calculation of n(g) is simplified by the fact that each of the basis functions has a definite crystal momentum so that the Fourier components n(g) could be computed directly in reciprocal space.

In evaluating the charge density for a NxN Hamiltonian, N corresponding to the plane wave, G', giving the largest kinetic energy, in principle Fourier components up to G = 2G' result. Thus about 8N Fourier components of the charge density are produced which may contribute to the Hartree and exchange correlation potentials on the next iteration of the Kohn-Sham equations. In this calculation, however, Fourier components beyond n(G') were set to zero. This had the advantage of a substantial saving in the computational resources required both because these terms did not have to be evaluated and because the reduced off-diagonal elements made the eigenvalue problem less time consuming. Thus an increase in the size of the Hamiltonians which could be considered resulted. Because only the lowest states are used in evaluating the ground state properties this procedure yields only a small change in the results for a given size, N, which is compensated by the larger Hamiltonian size which

could be treated.

The local exchange correlation functional used here was that calculated for the uniform electron gas by Ceperly and Alder (4) and parameterized by Perdew and Zunger (5). This is the same functional which was used in generating the pseudopotentials of Reference 3. Because the resulting local density functional is non-linear, the Fourier components of the exchange correlation potential could not be evaluated directly from the Fourier components of the charge density. Thus it was necessary to transform the charge densities at a number of points in the direct space unit cell, evaluate the exchange correlation potential at these points and then back transform these into reciprocal space. For this purpose a grid of about 12,000 points in the zinc blende unit cell were used, the exchange correlation energy term in the total energy expression also being evaluated in direct space as previously noted.

The remaining term in the total energy expression is the electrostatic interaction of the ion cores with each other and with a uniform compensating background. This was evaluated directly using Ewald's method (9) which permits the rapid evaluation of such energies in periodic structures. Ewald's method has also been used in calculating the energies of layers (see Chapter 7). It should be noted that all other interactions between the ion cores have been neglected in this calculation. In particular Born-Mayer repulsions and van der Waals' interactions were neglected. This was done because these were expected to result in negligible contributions to the total energy, the former because of the open tetrahedral structures in question and the later because the strong covalent bonds in these materials would render such effects negligible. Moreover, consideration of correlation between electrons in different ion cores is clearly beyond the scope of a pseudopotential calculation.

REFERENCES

(1) J. Ihm, A. Zunger and M.L. Cohen, J. Phys. C 12, 4409 (1979); corrigenda: J. Phys. C 13, 3095 (1980)

(2) J.R. Chelikowsky and S.G. Louie, Phys. Rev. B 29, 3470 (1984)

(3) G.B. Bachelet, D.R. Hamann and M. Schluter, Phys. Rev. B 26, 4199 (1982); erratum: Phys. Rev. B 29, 2309 (1984)

(4) D.M. Ceperly and B.J. Alder, Phys. Rev. Lett. 45, 566 (1980)

(5) J. Perdew and A. Zunger, Phys. Rev. B 23, 5048 (1981)

(6) M. Cardona, K.L. Shaklee and F.H. Pollak, Phys. Rev. 154, 699 (1967)

(7) K.M. Rabe and J.D. Joannopoulos, Bull. Am. Phys. Soc. 30, no.3, 466 (1985)

(8) D.J. Chadi, Phys. Rev. B 19, 2074 (1979)

(9) P.P. Ewald, Ann. Phys., Lpz. 64, 253 (1921)

CHAPTER 3

Valence charge densities in InSb and InAs

A) Introduction

For some time there has been considerable interest in the semiconductor-metal phase transitions diamond and zinc blende structure materials undergo when subjected to high pressures. With some exceptions (1-4) the transition at low temperatures is to a tetragonal β-Sn like structure or a rock salt structure, the high pressure structure being related to the ionicity of the material in question (4). Materials with low ionicities tend to have β-Sn like high pressure phases while more ionic materials tend to transform to rock salt structures. Among the III-V Indium compounds InP and InAs have rock salt high pressure phases while InSb has a β-Sn like phase. In either case the phase transition under pressure involves a change in coordination number and the breaking of covalent bonds.

Covalently bonded materials may be characterized by their highly nonspherical charge densities which are associated with a charge buildup in the bonds. The first (5) and perhaps the most direct experimental evidence of the existence of these bond charges was the observation of quasiforbidden X-ray reflections such as the (222) reflection in diamond structure materials. For zinc blende materials whose constituent atoms have similar atomic numbers such as AlP, GaAs and InSb the X-ray structure factors F(222) and $F(\bar{2}\bar{2}\bar{2})$ can yield a considerable amount of information regarding the valence electron distributions (6), specifically the (222) component of the Fourier transform. In zinc blende materials the (222) components of the valence charge densities may be regarded as containing contributions from two sources, the bond charges and charge buildups about the anions. This is in contrast to the situation in

diamond structure materials where only the first contribution can occur.

Recently the intensities of these X-ray reflections have been measured as a function of pressure for both Silicon and InSb (7). It was found that as the pressure was increased both materials behaved similarly in that these intensities fell off sharply just before the transition pressures were reached. This is a remarkable result in view of the fact that for such a strongly first order transition large fluctuations in the vicinity of the phase transition are not normally expected. Subsequently it was suggested that the (222) X-ray intensity may be a useful order parameter for the semiconductor-metal phase transition in diamond structure materials (8). Two interpretations of the experimental observations were offered (7). Firstly it was suggested that the valence electrons may have redistributed just before the transition volume was reached, this redistribution presumably destabilizing the tetrahedrally coordinated structures and leading to the phase transition. The second interpretation offered was that the decrease in the intensities may have resulted from the enhancement of an anharmonic effect related to the phase transition. The first possibility has been excluded for Silicon by an ab initio calculation of the valence charge density as a function of pressure (9), which showed that the low temperature X-ray structure factor, F(222), does not display any rapid changes near the transition volume. As yet the exact cause of the experimental observation is not known.

In this chapter charge densities and X-ray structure factors for InSb, calculated using the ab initio pseudopotential method described in Chapter 2, are reported at several volumes. In addition since it was felt that it would be desirable to study the behaviour of the valence electrons as a function of volume for a more ionic material having a rock salt high pressure phase,

valence charge densities for InAs were also calculated and are reported here.

B) Method and Results

For both InSb and InAs the calculations were done using a set of two special points in the irreducible wedge of the Brillouin zone. A basis of about 600 plane waves was employed for the calculations on InAs whereas for InSb about 530 plane waves were used. Increasing the number of plane waves to 725 at the experimental free volume, Ω_o, was found to change the lower Fourier components of the charge density by less than 0.002 electrons per cell for InSb. For InAs comparison with a calculation using about 530 plane waves implied that a similar convergence had been attained.

In Table 3-1 the first few Fourier components of the charge densities are given at the free volume. These are compared with the results of an earlier calculation by Chelikowsky and Cohen (10). In spite of the fact that they used an empirical pseudopotential and only about 90 plane waves in their calculation, it may be noted that the results of their calculation agree fairly well with the results presented here. In Table 3-2 values of the X-ray structure factors F(222) and F($\bar{2}\bar{2}\bar{2}$) of zinc blende structure InSb for Cu Kα radiation are presented. For InSb the transition occurs at about $\Omega/\Omega_o = 0.945$. The requisite ionic scattering factors were taken from published results of Hartree-Fock calculations (11,12). It will be noted that the free volume results are in fairly good agreement with the low temperature values obtained by Bilderback and Collela. The effect of any errors in the ionic scattering factors would be to change the values obtained somewhat but leave the trend, i.e., the absence of any dramatic changes under pressure, intact. Correspondingly for InAs values of n(222) at several volumes are presented in Table 3-3, the transition of InAs occuring at about $\Omega/\Omega_o = 0.91$.

In addition the positions of the bond charge maxima are given in Table 3-4.

In Figure 3-1 contour plots of the calculated valence charge densities of zinc blende structure InSb at Ω_o and at $\Omega/\Omega_o = 0.847$ are presented in a (110) section. The difference is also shown. The results are qualitatively very similar to those obtained for Silicon in Reference 9. From these contour plots and the X-ray structure factors it may be concluded that there is no evidence for a dramatic redistribution of the valence charge density in InSb near the transition volume or even at volumes much smaller than it and moreover that the redistribution which does occur is mainly spherical about the ions. This result implies that the experimentally observed drop in the X-ray structure factors is very likely due to an anharmonic effect.

In Figure 3-2 (110) sections of the valence charge densities of zinc blende structure InAs are shown at $\Omega/\Omega_o = 1.000$, 0.898 and 0.850. These indicate that just as in the case of InSb there is no dramatic redistribution of the valence electrons as InAs is compressed and in particular that the bond charge remains a prominent feature of the zinc blende structure. There is however a gradual shift of the bond charge maxima away from the anion sites under pressure. This is reflected in the values of n(222) which show a gradual phase shift and a small decrease in magnitude as the volume is reduced. This observation is consistent with the results of Raman scattering measurements which show that the splitting of the optical phonon decreases with increasing pressure implying that the effective charge, e_T^*, also decreases under pressure (14). There is a similar but smaller effect present in InSb as may be noted from Figure 3-1.

These results lend support to the idea that the semiconductor-metal phase transition is not caused by a redistribution of the electrons in the semiconductor but is related to the competition between the covalent bonding

energy with Madelung and repulsive energies as suggested by Yin and Cohen (9). This point will be discussed further in Chapter 4 where polarization charge densities associated with various distortions of the zinc blende lattice are reported.

TABLE 3-1 Fourier components of the charge density in electrons per cell for InSb and InAs at the free volume. The origin is at the cation site.

	Fourier Components			
Component	InAs		InSb	
	Ref. 10	This calculation	Ref. 10	This calculation
111	0.885, -1.998	0.659, -2.202	0.994, -1.915	0.846, -1.882
200	-0.860, 0	-1.238, 0	-0.718, 0	-0.766, 0
220	0.135, 0	0.177, 0	0.144, 0	-0.024, 0
311	-0.200, -0.151	-0.222, -0.092	-0.223, -0.161	-0.227, -0.267
222	0.030, -0.437	-0.035, -0.228	-0.020, -0.409	0:091, -0.252
400	-0.215, 0	-0.368, 0	-0.245, 0	-0.467, 0
331	0.040, 0.047	-0.022, 0.209	0.045, 0.038	-0.054, 0.223
420	0.055, 0	0.167, 0	0.048, 0	0.152, 0
422	-0.042, 0.020	-0.232, 0.031	-0.043, 0.024	-0.235, 0.015

TABLE 3-2 X-ray structure factors for InSb in the ZnS structure for Cu kα radiation.

Ω/Ω_o	F(222)	$F(\bar{2}\bar{2}\bar{2})$
1.035	6.41	5.23
1.000	6.42	5.26
0.985	6.42	5.26
0.938	6.44	5.29
0.892	6.45	5.33
0.847	6.46	5.37
0.804	6.48	5.41
0.762	6.49	5.46
0.722	6.50	5.51
0.646	6.51	5.61

TABLE 3-3 Fourier components of the charge density for InAs

Ω/Ω_o	n(222)	magnitude
1.055	(-0.060,-0.230)	0.238
1.000	(-0.035,-0.228)	0.231
0.948	(-0.001,-0.225)	0.225
0.898	(0.001,-0.221)	0.221
0.850	(0.037,-0.216)	0.219
0.804	(0.058,-0.211)	0.219

TABLE 3-4 Location of the bond charge maximum in zinc blende InAs. x = 0 is a cation site whereas x = 1 is the anion site. As-BC is the distance in atomic units of the bond charge from an anion site.

Ω/Ω_o	x	As-BC
1.055	0.712	1.447
1.000	0.706	1.451
0.948	0.698	1.465
0.898	0.690	1.477
0.850	0.682	1.487
0.804	0.674	1.496

REFERENCES

(1) S.-C. Yu, I.L. Spain and E.F. Skelton, J. Appl. Phys. 49, 4741 (1978)

(2) A. Werner, H.D. Hochheimer, K. Strossner and A. Jayaraman, Phys. Rev. B 28, 3330 (1983)

(3) K.J. Smolander, J. Phys. C 16, 3673 (1983) and references therein

(4) S.-C. Yu, I.L. Spain and E.F. Skelton, Solid State Commun. 25, 49 (1978)

(5) P.P. Ewald and H. Honl, Ann. Phys. 25, 281 (1931)

(6) R. Collela, Phys. Scr. 15, 143 (1977) and references therein

(7) D.R. Yoder-Short, R. Collela and B.A. Weinstein, Phys. Rev. Lett. 49, 1438 (1982)

(8) N.H. March and F. Flores, Solid State Commun. 49, 649 (1984)

(9) M.T. Yin and M.L. Cohen, Phys. Rev. Lett. 50, 1172 (1983)

(10) J.R. Chelikowsky and M.L. Cohen, Phys. Rev. B 14, 556 (1976)

(11) D.T. Cromer and J.B. Mann, Acta. Cryst. A24, 321 (1968)

(12) D.T. Cromer and D. Liberman, J. Chem. Phys. 53, 1891 (1970)

(13) D.H. Bilderback and R. Collela, Phys. Rev. Lett. 35, 858 (1975); Phys. Rev. B 13, 2479 (1976)

(14) A. Jayaraman, V. Swaminathan and B. Batlogg, Pramana 23, 405 (1984); K. Aoki, E. Anastassakis and M. Cardona, Phys. Rev. B 30, 681 (1984)

CHAPTER 4

Structural Properties and Polarization Charges in InSb

A) Introduction

In Chapter 3 valence charge densities for zinc blende structure InSb at its free volume and at a reduced volume for which the semiconducting phase is unstable were reported. It was concluded that under pressure the behaviour of the valence electrons in InSb is qualitatively very similar to that found in Silicon (1) and InAs.

Here polarization charge densities and energies associated with various zone center phonons are reported for zinc blende structure InSb at its free volume and also at a reduced volume. These are discussed in relation to the corresponding quantities in other covalently bonded materials (2-4).

B) Results and Discussion

i) Total energy calculations

In Table 4-1 the equilibrium lattice constant, a_o, bulk modulus, B, and its pressure derivative, B', obtained by fitting total energies calculated using the method described in Chapter 2, at a variety of volumes, Ω, to Murnaghan's equation of state (5), are presented. The shear elastic constant, $2c_s=c_{11}-c_{12}$, c_{44} and the internal strain parameter, ζ, calculated both at the experimental free volume, Ω_o, and at a reduced volume for which the zinc blende structure is unstable are also presented. (In InSb the semiconductor-metal phase transition occurs at about $\Omega/\Omega_o=0.945$.) The shear constant, c_s, was calculated from the total energy changes associated with homogeneous

tetragonal distortions in which one of the cube axes was stretched or compressed by a small amount (typically 1%) while the other two axes were also changed in order to keep the volume of the unit cell constant. The internal strain parameter and c_{44} were calculated from the energy changes associated with homogeneous trigonal distortions, relative displacements of the anion and cation sublattices along a bond direction and combinations of these. The values of c_s and c_{44} were found to change somewhat as the volume was reduced to $\Omega/\Omega_o=0.935$ from Ω_o. However in view of the probable accuracy of this calculation it is difficult to attach much significance to these changes. However the increase in the internal strain parameter as the volume is reduced is significant, this increase reflecting an expected weakening of bond bending forces relative to bond stretching forces under pressure. The calculated elastic constants are compared with the low temperature experimental values obtained by Slutsky and Garland (6), the pressure derivative of the bulk modulus being compared to that measured by Bashkin and Peresada (7). It may be noted that the agreement between the experimental and theoretical values is reasonable although not quite as good as obtained for certain other materials using similar methods (8). The calculated internal strain parameter at the free volume, $\zeta=0.43$, is somewhat lower than experimental values (9-12) for Silicon (0.74,0.72), Germanium (0.72) and GaAs (0.764) but is comparable to previous ab initio calculations (13,14) on these materials which all yielded values lower than those found experimentally. As noted in Reference 13 the discrepancy is outside the probable limits of the theoretical uncertainty and may be due to the assumption of overlapping spherical atoms in the analysis of the experimental data (see Chapter 3) and anharmonicity due to finite strains. In fact the magnitude of n(600) obtained in this calculation is 0.05 at zero strain and increases by about 6% under a 1% strain along a (111) direction. More importantly it is found that the transverse optical phonon becomes highly

anharmonic under strain as discussed below.

In Table 4-2 calculated frequencies of the zone center transverse optical phonon, ω_{TO}, and the corresponding anharmonic force constant, k_{xyz}, defined by:

$$\Delta E_{tot}(u) = ku^2 + 8\, k_{xyz}\left(\frac{u}{\sqrt{3}}\right)^3 \qquad (4\text{-}1)$$

are presented for the zinc blende structure. Here $\Delta E_{tot}(u)$ is the change in total energy per unit cell for a relative displacement between the two sublattices of 2u. ω_{TO} and k_{xyz} were calculated with the same distortions used for c_{44} and ζ. It will be noted that at the free volume the experimental (15) and theoretical frequencies are in excellent agreement. The phonon frequency and anharmonic force constant increase smoothly over the entire volume range studied including the range for which the structure is unstable and can be well described by a Gruneisen parameter, γ_{TO}=0.97. The corresponding parameters for InP and InAs have been measured to be 1.44 (16) and 1.2 (17) respectively. Thus the present results would seem to suggest that InSb follows the chemical trend. However a recent state of the art Raman scattering study (18) has yielded a value of γ_{TO}=1.41. Although the phonon frequency obtained in that study does not agree to within the stated precision with a previous neutron scattering measurement (15), it seems unlikely that the discrepancy in the Gruneisen parameter can be explained by an experimental error. At present its source is unknown. In any case the present results do show that neither ω_{TO} nor k_{xyz} display any exceptional behaviour near the transition volume.

In Table 4-3 the anharmonic force constant k_{xyz} is given for (111) strains of 0.5% and 1%. The frequency, ω_{TO}, remained unchanged under these strains to within the accuracy of this calculation and thus is not shown. Under strain

k_{xyz} changes sign and increases dramatically in magnitude. While not conclusive, this result suggests that experimental determinations of the internal strain parameters in similar materials may indeed be affected by anharmonic effects caused by finite strains as suggested by Nielsen and Martin (13). This might also be related to a recently reported anisotropic stress sensitive temperature dependence of the lattice constant of Silicon (19) although it is not possible to arrive at any definite conclusion based on the present results.

ii) Charge densities

In Figure 4-1 (110) sections of the valence charge densities (a) and polarization charge densities for a frozen zone center TO phonon (b) and a homogeneous tetragonal distortion (c) are shown for zinc blende structure InSb both at the free volume and at a reduced volume, $\Omega/\Omega_o=0.935$. The polarizations shown correspond to 1% stretches of the 111 bonds for the optical phonon whereas for the tetragonal distortion a 1% stretch of the z-axes with a corresponding compression of the other two axes in order to maintain a constant volume was used. In order to make the redistribution more evident for the later case (c) the component corresponding to a rigid homogeneous distortion of the electron density has been eliminated.

As discussed in Chapter 3 the valence charge densities at the two volumes are very similar, the main difference consisting of a small movement of the bond charge maximum away from the anion sites under pressure consistent with Raman scattering measurements which indicate that under pressure the zone center LO-TO phonon splitting decreases in most zinc blende materials (17,18,20-24) implying a reduction in the effective charge, e_T^*. It is interesting to note that similar reductions are observed at high temperatures (25) perhaps reflecting the the similar nature of the high

pressure and high temperature (melting) phase transitions (26). Like the charge density it will be noted from Figure 4-1 that both the polarization charge of the TO phonon and that of the tetragonal shear distortion are very similar under pressure to the corresponding quantities at the free volume. While this may be expected for the TO phonon for which the forces are dominated by ion-ion interactions (27), this result for the tetragonal distortion lends further support to the conclusion of Yin and Cohen (1) that the semiconductor-metal phase transition is not caused by an instability in the electron distribution but rather a competition between the covalent bonding energies and the Ewald interactions.

The polarization associated with the TO phonon contains a strong dipole-like contribution which corresponds to a rigid displacement of much of the bond charge with the anion. There is also a small charge transfer from the stretched bond to the compressed bonds. This transfer is slightly off center in that the charge tends to flow to the region of the compressed bonds furthest from the stretched bond. The dominance of this dipole-like contribution in InSb is very similar to that found in ZnSe (3,4) in spite of the fact that InSb is much less ionic, having a Phillips ionicity of 0.321 as compared to 0.63 for ZnSe. The charge transfer among the bonds is less evident than that found in Silicon (2,14) and Germanium (3). The present results indicate that even for covalent materials with moderate ionicities the bond charges are fairly strongly associated with the anions.

The polarizations for the tetragonal distortion consist of small movements of the valence electrons towards the ions between bonds where the bond angles are reduced and conversely, a redistribution away from the ions between bonds where the bond angle is increased. The total bond charge is not changed much by this distortion.

To summarize, the behaviour of the valence electrons under high pressures, for which the zinc blende structure is unstable, is almost the same as that at the free volume. The fact that the covalent bonds are not destroyed under pressure as discussed here and in Chapter 3 may be qualitatively understood by making use of some of Phillips' ideas (28) and noting the strengthening of the pseudopotential in the bonding region as the volume is reduced. This arises because under pressure the minima in the Indium pseudopotentials move towards the bond charges which are located near the minima of the Antimony pseudopotentials. (The minimum of the Antimony pseudopotential is about 1.7 a.u. away from the anion sites while at the free volume the minimum of the Indium pseudopotential is about 3.7 a.u. away. The bond charge maximum is located about 1.85 a.u. from the anion.) This mechanism also helps to account for the movement of the bond charges away from the anions as the crystal is compressed since as the anion-cation distance is reduced the bond charges are expected to be influenced to a greater extent by the cation pseudopotentials.

C) Conclusion

In conclusion charge densities, polarizations and structural properties of zinc blende InSb have been calculated, reasonable agreement with experimental data being obtained. InSb is found not to behave much differently than other zinc blende materials in that the near instability of its bonding was found not to manifest itself in a substantially different response of its valence electrons to the distortions studied. This is consistent with the fact that even at greatly reduced volumes the responses were found not to be largely different from those at the free volume.

TABLE 4-1 Structural properties of zinc blende InSb. n denotes the size of the plane wave basis. The elastic constants are given in GPa while the lattice constant, a_o, is in angstroms.

	Ω/Ω_o	This study n=531	This study n=725	Experiment
a_o		6.34	6.34	6.49
B		50	50.7	48.31
B'		4.7	4.9	4.91
c_{44}	1.000	35		31.32
	0.935	34		
$c_{11}-c_{12}$	1.000	36		31.30
	0.935	38		
ζ	1.000	0.43		
	0.935	0.55		

TABLE 4-2 Frequency of the transverse optical phonon and the anharmonic force constant associated with it for zinc blende InSb. The frequencies are given in THz, whereas the force constants are in units of eV/angstrom^3. The experimental value is denoted by (exp).

Ω/Ω_o	ω_{TO}	k_{xyz}
1.000	35.0	-38
1.0 (exp)	34.8	
0.985	35.6	-42
0.935	37.4	-44
0.892	39.3	-58
0.847	41.2	-71

TABLE 4-3 Anharmonic force constant as a function of strain, $e_{12}=e_{23}=e_{13}=e$, at the free volume.

e	k_{xyz}
0	-38
0.005	0
0.010	214

REFERENCES

(1) M.T. Yin and M.L. Cohen, Phys. Rev. Lett. 50, 1172 (1983)

(2) B.N. Harmon, W. Weber and D.R. Hamann, J. Phys. (Paris) 42, C6-628 (1981); See also U. Pietsch, Phys. Stat. Sol.(b) 120, 183 (1983) and Phys. Stat. Sol.(b) 126, 595 (1984) for a treatment based on a simple model but note that the results are suspect as the pressure variation is markedly different from that given in reference 1.

(3) R. Resta, Phys. Rev. B 27, 3620 (1983)

(4) R. Resta and A. Baldereschi, J. Phys. (Paris) 42, C6-661 (1981)

(5) F.D. Murnaghan, Proc. Nat. Acad. Sci. USA, 3, 244 (1944)

(6) L.J. Slutsky and C.W. Garland, Phys. Rev. 113, 167 (1959)

(7) I.O. Bashkin and G.I. Peresada, Fiz. Tverd. Tela 16, 3166 (1974), [Sov. Phys. Solid State 16, 2058 (1975)]

(8) See for example S.G. Louie, to be published in Electronic Structure, Dynamics and Quantum Structural Properties of Condensed Matter (1985) and references therein

(9) H. D'Amour, W. Denner, H. Schulz and M. Cardona, J. Appl. Cryst. 15, 148 (1982)

(10) C.S.G. Cousins, L. Gerward, J. Staun Olsen, B. Selsmark and B.J. Sheldon, J. Appl. Cryst. 15, 154 (1982)

(11) C.S.G. Cousins, L. Gerward, K. Nielsen, J. Staun Olsen, B. Selsmark, B.J. Sheldon and G.E. Webster, J. Phys. C 15, L651 (1982)

(12) C.N. Koumelis, G.E. Zardas, C.A. Landos and D.K. Leventuri, Acta Cryst. Sect. A 32, 84 (1975)

(13) O.H. Nielsen and R.M. Martin, Phys. Rev. Lett. 50, 697 (1983); R.M. Martin and O.H. Nielsen, to be published in Electronic Structure, Dynamics and Quantum Structural Properties of Condensed Matter (1985)

(14) B.N. Harmon, W. Weber and D.R. Hamann, Phys. Rev. B 25, 1109 (1982)

(15) D.L. Price, J.M. Rowe and R.M. Nicklow, Phys. Rev B 3, 1268 (1971)

(16) R. Trommer, H. Muller, M. Cardona and P. Vogl, Phys. Rev. B 21, 4869 (1980)

(17) A. Jayaraman, V. Swaminathan and B. Batlogg, Pramana 23, 405 (1984)

(18) K. Aoki, E. Anastassakis and M. Cardona, Phys. Rev. B 30, 681 (1984)

(19) Y. Soejima, K. Inoue, T. Kawahara, N. Ohama and A. Okazaki, J. Phys. C 18, 2431 (1985)

(20) B.A. Weinstein and G.J. Piermarini, Phys. Rev. B 12, 1172 (1975)

(21) B.A. Weinstein, J.B. Renucci and M. Cardona, Solid State Commun. 9, 1651 (1971)

(22) S.S. Mitra, O. Brafman, W.B. Daniels and R.K. Crawford, Phys. Rev. 186, 942 (1969)

(23) R. Trommer, E. Anastassakis and M. Cardona in Light Scattering in Solids, eds. M. Balkanski, R.C.C. Leite and S.P.S. Porto (Flamarion, Paris, 1976), page 396

(24) This is not the case in AlSb - C.J. Buchenauer, F. Cerdeira and M. Cardona in Light Scattering in Solids, eds. M. Balkanski, R.C.C Leite and S.P.S. Porto (Flamarion, Paris, 1976), page 280

(25) See, for example, E. Liarokapis and E. Anastassakis, Phys. Rev. B 30, 2270 (1984)

(26) N.H. March and F. Flores, Solid State Commun. 49, 649 (1984)

(27) H. Wendel and R.M. Martin, Phys. Rev. Lett. 40, 950 (1978)

(28) J.C. Phillips, Bonds and Bands in Semiconductors, (Academic Press, New York, 1973)

CHAPTER 5

Second Sound Velocities in Hexagonal Crystals

A) Introduction

In 1944 Peshkov (1) observed second sound in superfluid ^{4}He and in 1947 proposed that similar propagating temperature waves may exist in solids (2). The first observation of second sound in a solid was for hcp ^{4}He (3). Second sound has since then provided valuable information on phonon scattering processes (4) and particularly normal processes. The condition for the existence of second sound is that there exist scattering mechanisms which cause a nonequilibrium phonon distribution to relax to an intermediate thermalized distribution with a net energy flux in a time that is short compared with that required for other processes to destroy it (5,6). Based on this two types of second sound have been identified by Enz (5), drifting second sound with a propagation velocity v_{II}, which occurs when normal processes yield a drifting phonon distribution in a time short compared with that required for other mechanisms to destroy it, and driftless second sound with a propagation velocity v'_{II}, which may occur when momentum (wave vector) correlations decay rapidly as compared with the energy flux.

All observations of second sound appear to be of the drifting variety with the possibility of observing driftless second sound uncertain at best (7). Despite this both v_{II} and v'_{II} are of considerable interest in studying scattering processes, the drifting velocity directly in experiments involving second sound and the driftless velocity because it enters in the Callaway expression for the thermal conductivity (8-10) which is frequently used in the analysis of experimental data on heat conduction.

Numerical calculations of second sound velocities for some cubic crystals have been performed by Varshni and Konti (11), Hardy and Jaswal (12) and Jaswal and Hardy (13). For anisotropic materials drifting second sound velocities have been calculated for hcp ^{4}He at 20.97 cm^3/mole and Bismuth by Maris (14).

Here values of the drifting and driftless second sound velocities are presented for 12 hexagonal crystals with arbitrary orientations. The variation of the velocities with density for solid Helium is also given. Further it is shown that the discrepancy pointed out by Maris (14) between the experimental and theoretical values of v_{II} in hcp ^{4}He can be reduced if another set of elastic constants is used in the calculations.

B) Theory

i) Drifting second sound

It has recently been shown that neglecting dispersion, that is at temperatures much smaller the Debye temperature, the drifting second sound propagation velocity is given by (14):

$$v_{II}^2 = n_i n_j (c_d^2)_{ij} \tag{5-1}$$

where (n_1,n_2,n_3) is a unit vector specifying the direction of second sound propagation and where

$$(c_d^2)_{ij} = \frac{1}{9} T_o C \rho_{ij}^{-1} \tag{5-2}$$

in which

$$C = (2\pi^2 k_B^4 T_o^3 / 5\hbar^3) \int_{\Omega=1} \langle c_\ell^{-3}(\hat{\kappa}) \rangle \tag{5-3}$$

$$\rho_{ij} = (2\pi^2 k_B^4 T_o^4/5\hbar^3) \sum_{\zeta=1}^{3} \langle k_i k_j c_\zeta^{-5}(\hat{k})\rangle \qquad (5\text{-}4)$$

Here k_B is the Boltzmann constant, $\hbar$ is Plank's constant and T_o is the crystal temperature. The sum on ζ is over the three acoustic modes, c_ζ being the phase velocity of mode ζ in the direction $\hat{k}$. The averages are over $\hat{k}$.

Thus for hexagonal crystals the nonzero elements of $(c_d^2)_{ij}$ are

$$(c_d^2)_{xx} = (c_d^2)_{yy} = \frac{1}{9}\left[\sum_{\zeta=1}^{3} \langle c_\zeta^{-3}(\hat{k})\rangle\right] \left[\sum_{\zeta=1}^{3} \langle k_y^2\, c_\zeta^{-5}(\hat{k})\rangle\right]^{-1}$$

$$(c_d^2)_{zz} = \frac{1}{9}\left[\sum_{\zeta=1}^{3} \langle c_\zeta^{-3}(\hat{k})\rangle\right] \left[\sum_{\zeta=1}^{3} \langle k_z^2\, c_\zeta^{-5}(\hat{k})\rangle\right]^{-1} \qquad (5\text{-}5)$$

x,y,z being orthogonal with z coinciding with the hexad axis.

ii) Driftless second sound

For the driftless case the following expression for the time derivative of the energy flux, $\vec{S}(\vec{r},t)$, may be obtained (6):

$$\left[\frac{\partial}{\partial t} + \frac{1}{\tau}\right] S_i(r,t) = -\frac{1}{\tau} K_{ij} \frac{\partial T(r,t)}{\partial r_j} \qquad (5\text{-}6)$$

where τ is a relaxation time characterizing the decay of modes giving rise to the energy flux, T is a local temperature and $\vec{K}_{ij}$ is a single relaxation time approximation to the thermal conductivity,

$$K_{ij} = \tau(2\pi^2 k_B^4 T_o^3/5\hbar^3) \sum_{\zeta=1}^{3} \langle \tilde{c}_{i\zeta}(k)\, \tilde{c}_{j\zeta}(k)/c_\zeta^3(k)\rangle \qquad (5\text{-}7)$$

where $\tilde{c}$ is the group velocity. When Equation (5-6) is combined with the energy balance equation,

$$C \frac{\partial}{\partial t} T(r,t) + \frac{\partial}{\partial r_i} S_i(r,t) = 0 \qquad (5\text{-}8)$$

a damped wave equation for T(r,t) results

$$\frac{\partial^2 T(r,t)}{\partial t^2} + \frac{1}{\tau} \frac{\partial T(r,t)}{\partial t} - \frac{K_{ij}}{\tau C} \frac{\partial^2 T(r,t)}{\partial r_i \partial r_j} = 0 \qquad (5\text{-}9)$$

which has three solutions, two of which do not involve heat propagation, the third one having a propagation velocity v'_{II} given by

$$v'_{II} = n_i n_j (c_{dl}^2)_{ij} \qquad (5\text{-}10)$$

where the unit vector (n_1,n_2,n_3) specifies the direction of driftless second sound propagation.

For hexagonal crystals the nonzero elements of $(c_{dl}^2)_{ij}$ are

$$(c_{dl}^2)_{xx} = (c_{dl}^2)_{yy} = \left[\sum_{\zeta=1}^{3} \langle (\tilde{c}_{\zeta x}(k)/c_\zeta^3(k)\rangle \right] \left[\sum_{\zeta=1}^{3} \langle c_\zeta^{-3}(k)\rangle\right]^{-1}$$

$$(c_{dl}^2)_{zz} = \left[\sum_{\zeta=1}^{3} \langle \tilde{c}_{\zeta z}(k)/c_\zeta^3(k)\rangle\right]\left[\sum_{\zeta=1}^{3} \langle c_\zeta^{-3}(k)\rangle\right]^{-1} \qquad (5\text{-}11)$$

C) Hexagonal Symmetry

At low temperatures where second sound is dominated by low energy acoustic phonons the phonon frequencies $\omega_\zeta(k)$ and hence velocities $\tilde{c}_\zeta(k)$ are determined by the elastic constants c_{ijkl} of the material in question.

$$|c_{ijk\ell} k_j k_\ell - \rho\omega^2 \delta_{ik}| = 0 \qquad (5\text{-}12)$$

$$\tilde{c}_\zeta(k) = \nabla_k \omega_\zeta(k) \qquad (5\text{-}13)$$

For hexagonal crystals there are five independent elastic constants in terms of which the above eigenvalue problem can be solved exactly (15).

In this dispersionless case in order to obtain the second sound velocities it is necessary only to average over the direction of k. Thus

$$(c_d^2)_{xx} = (c_d^2)_{yy} = \frac{2}{9}\left[\int_0^{\pi/2} \sum_{\zeta=1}^{3} c_\zeta^{-3}(\theta) \sin\theta\, d\theta\right]\left[\int_0^{\pi/2} \sum_{\zeta=1}^{3} c_\zeta^{-5}(\theta) \sin^3\theta\, d\theta\right]^{-1}$$

$$(c_d^2)_{zz} = \frac{1}{9}\left[\int_0^{\pi/2} \sum_{\zeta=1}^{3} c_\zeta^{-3}(\theta) \sin\theta\, d\theta\right]\left[\int_0^{\pi/2} \sum_{\zeta=1}^{3} c_\zeta^{-5}(\theta) \cos^2\theta \sin\theta\, d\theta\right]^{-1}$$

$$(c_{d\ell}^2)_{xx} = (c_{d\ell}^2)_{yy} = \left[\int_0^{\pi/2} \sum_{\zeta=1}^{3} c_{\zeta x}^2(\theta) c_\zeta^{-3}(\theta) \sin\theta\, d\theta\right] \qquad (5\text{-}14)$$

$$\times \left[\int_0^{\pi/2} \sum_{\zeta=1}^{3} \tilde{c}_\zeta^{-3}(\theta) \sin\theta\, d\theta\right]^{-1}$$

$$(c_{d\ell}^2)_{zz} = \left[\int_0^{\pi/2} \sum_{\zeta=1}^{3} \tilde{c}_{\zeta z}^2(\theta)\, c_\zeta^{-3}(\theta) \sin\theta\, d\theta\right]\left[\int_0^{\pi/2} \sum_{\zeta=1}^{3} c_\zeta^{-3}(\theta) \sin\theta\, d\theta\right]^{-1}$$

where the $\tilde{c}$'s are appropriate averages over the angle $\phi=\tan^{-1}(k_2/k_1)$ and we have used the fact that for hexagonal crystals it is not necessary to average the phase velocities over ϕ.

D) Data and Results

The experimental data employed in the calculation are summarized in Table 5-1. For BeO two sets of elastic constants have been measured which differ from each other and appear to have the same sort of accuracy;

calculations were carried out for both sets and these are represented by BeO-1 and BeO-2 in the tables. The required integrals were evaluated numerically. The calculated values of the second sound velocities are shown in Table 5-2.

E) Discussion

There are a number of substances in Table 5-1 for which elastic constants are available only at room temperature. In such cases it is expected that the true low temperature second sound velocities will be slightly higher than those given in Table 5-2. Based on the temperature variations of the elastic constants of several tetrahedrally coordinated compounds and the melting points of the materials in question it is estimated that the deviation is less than 1% for SiC and BeO, 2% for ZnO, ZnS and CdS and 3% for ZnTe and CdSe. For CeF_3 we have not been able to find any data on the temperature variation of the elastic constants of similar compounds but it seems unlikely that the actual second sound velocities at 0K will be higher than those given in Table 5-2 by more than 10%.

To our knowledge so far the only detailed measurements of the anisotropy of v_{II} are those on hcp ^{4}He at 25 bar made by Mueller and Fairbank (16). Table 5-2 shows that the anisotropy of v_{II} is expected to be about 5% for AgI. For all other substances in the table with the exception of Helium the anisotropy in the v_{II} turns out to be less than 3%.

The propagation of second sound in ^{4}He has been studied by a number of workers. It was of some interest to examine the variation of second sound velocities with density for this substance. Greywall (17) has proposed that the elastic constants of ^{4}He have the following molar volume dependence:

$$c_{ij} = c_{ij}\,(V_o)\,(V_o/V)^{5.73} \qquad (5\text{-}15)$$

This implies that the variations of second sound velocities with density are given by

$$v_{II}(\rho) = v_{II}(\rho_o)\,(\rho/\rho_o)^{2.365} \qquad (5\text{-}16)$$

The variations of the velocities with density as given by this relation are shown by the full line curves in Figure 5-1 taking the velocities at ρ_o to be those calculated using the elastic constants of Crepeau et al. (18). Also shown in Figure 5-1 are the second sound velocities calculated from the elastic constants of Frank and Wanner (19) at ρ=0.19698 g/cm^3 and at ρ=0.20760 g/cm^3 (second and third entries for ^{4}He in Table 5-2). It will be noticed that several of the points lie substantially above the corresponding curves. Thus it seems that there is an inconsistency between the elastic constants of Crepeau et al (18) and those of Frank and Wanner (19) or else that the relation (5-15) is not good enough. To further illustrate this point in Figure 5-2 the variation of the second sound velocities with density as calculated from the elastic constants of Frank and Wanner at ρ=0.19698 g/cm^3 and Equation (5-16). The points calculated from the elastic constants at ρ=0.20760 g/cm^3 lie close enough to the theoretical curve but those from the elastic constants of Crepeau et al. show substantial discrepancies.

Maris (14) has recently calculated the anisotropy of v_{II} in ^{4}He using elastic constants which are almost the same as those of Crepeau et al. and has found that the experimental second sound velocities measured by Mueller and Fairbank (16) are somewhat higher than the theoretical values. In view of the findings of the last paragraph it was of interest to examine whether this discrepancy can be accounted for by using the elastic constants of Frank and

Wanner. In Figure 5-3 we show the second sound velocity as a function of propagation direction in ^{4}He. The solid line theoretical curve calculated from the elastic constants of Frank and Wanner extrapolated to ρ=0.19087 g/cm^3 by Equation 5-15 is seen to lie nearer the experimental points than the dashed-line curve obtained by Maris. However it seems that a certain amount of discrepancy between experiment and theory still remains to be explained.

ACKNOWLEDGEMENT

The author would like to express his gratitude to Dr. Richard Martin for kindly pointing out an error in the elastic constants of AgI. The author regrets any inconvenience which may have resulted from his error.

TABLE 5-1 Experimental data used in the calculations. For Hydrogen and Deuterium X represents the concentration of molecules with rotational angular momentum J=1. The source of the elastic constant data is given under Ref. RT stands for room temperature. The elastic constants are in GPa.

Substance	T(K)	Density (g/cm^3)	c_{11}	c_{33}	c_{44}	c_{12}	c_{13}	Ref.
^{4}He	1.4	0.19087	0.0405	0.0554	0.0124	0.0213	0.0105	a
^{4}He		0.19698	0.055	0.071	0.0140	0.029	0.0131	b
^{4}He		0.20760	0.076	0.098	0.0196	0.042	0.0198	b
H_2 (X=0)	4.2	0.0880	0.42	0.51	0.11	0.18	0.05	c
(X=.75)	4.2	0.0882	0.362	0.440	0.083	0.119	0.041	c
(X=.75)	4.2	0.0936	0.537	0.644	0.132	0.179	0.072	c
D_2 (X=.02)	4.2	0.2000	0.82	1.02	0.23	0.29	0.09	c
(X=.33)	4.2	0.1996	0.668	0.788	0.164	0.232	0.111	c
(X=.33)	4.2	0.2045	0.791	0.904	0.196	0.270	0.158	c
AgI	10	5.994	37.4	43.9	4.10	28.8	23.8	d
BeO-1	RT	3.010	460.6	491.6	147.7	126.5	88.48	e
BeO-2	RT	3.010	470.0	494.0	153.0	168.0	119.0	f
ZnO	RT	5.676	207.0	209.5	44.8	117.7	106.1	g
ZnS	RT	4.089	124.2	140.0	28.64	60.15	45.4	e
ZnTe	RT	6.34	86.0	93.0	20.2	37.0	30.0	h
CdS	4.2	4.8217	87.5	95.70	14.80	54.55	48.25	i
CdSe	RT	5.684	74.90	84.51	13.15	46.09	39.26	e
SiC	RT	3.2145	502.0	565.0	169.0	95.0	56.0	j
CeF_3	RT	6.160	180.0	225.0	34.2	88.0	64.0	k

a) Reference 18

b) Reference 19

c) R. Wanner and H. Meyer, J. Low Temp. Phys. 11, 715 (1973)

d) T.A. Fjeldy and R.C. Hanson, Phys. Rev. B 10, 3569 (1974)

e) C.F. Cline, H.L. Dunegan and G.W. Henderson, J. Appl. Phys. 38, 1944 (1967)

f) G.G. Bentle, J. Am. Ceram. Soc. 49, 125 (1966)

g) E.F. Tokarev, I.B. Kobyakov, I.P. Kuzimina, A.N. Lobachev and G.S. Pado, Fiz. Tverd. Tela 17, 2487 (1975) [Sov. Phys. Solid State 17, 629 (1975)]

h) R.F.S. Hearmon in Londolt-Bornstein Numerical Data and Functional Relationships in Science and Technology, (Springer, New York, 1979), Group III, Vol. 11, p. 39

i) D. Gerlich, J. Phys. Chem. Solids 28, 2575 (1967)

j) G. Arlt and G.R. Schodder, J. Acoust. Soc. Am. 37, 384 (1965)

k) S. Hart, Phys. Status Solidi A 17, K107 (1973)

TABLE 5-2 Calculated values of the two second sound velocities in m/s. The densities have been repeated for Hydrogen and Helium for identification purposes.

Substance	Density	v_{IIx}	v_{IIz}	v'_{IIx}	v'_{IIz}
^{4}He	0.19087	141.3	158.9	153.5	174.9
^{4}He	0.19698	158.3	169.7	172.3	194.4
^{4}He	0.20760	177.7	194.3	193.9	220.1
H_2 (X=0)	0.0880	702.9	723.7	761.9	840.7
(X=.75)	0.0882	661.5	645.9	725.9	792.6
(X=.75)	0.0936	793.2	782.4	860.9	927.5
D_2 (X=.02)	0.2000	685.8	695.1	736.7	800.1
(X=.33)	0.1996	601.2	594.6	650.7	695.9
(X=.33)	0.2045	647.1	639.2	697.4	733.8
AgI		504.6	528.5	556.7	637.2
BeO-1		4396	4404	4634	4711
BeO-2		4283	4388	4519	4639
ZnO		1670	1701	1779	1810
ZnS		1656	1680	1775	1864
ZnTe		1140	1136	1219	1253
CdS		1085	1100	1166	1219
CdSe		941	960	1014	1074
SiC		4651	4623	4894	5003
CeF_3		1564	1543	1711	1871

REFERENCES

(1) V. Peshkov, J. Phys. (Moscow) 8, 381 (1944)

(2) V. Peshkov, Report on an International Conference on Fundamental Particles and Low Temperature Physics, (The Physical Society, London, 1947), Vol. II, p. 19

(3) C.C. Ackerman, B. Bertman, H.A. Fairbank and R.A. Guyer, Phys. Rev. Lett. 16, 789 (1966)

(4) H. Beck in Dynamical Properties of Solids, edited by G.K. Horton and A.A. Maradudin (North-Holland, New York, 1975), Vol. II, pp. 205-284, especially pp. 227-234

(5) C.P. Enz, Ann. Phys. 46, 114 (1968)

(6) R.J. Hardy, Phys. Rev. B 2, 1193 (1970)

(7) H. Beck, P.F. Meier and A. Thellung, Phys. Status Solidi A 24, 11 (1974), see especially pages 60 and 61

(8) J. Callaway, Phys. Rev. 113, 1046 (1959)

(9) M.G. Holland, Phys. Rev. 132, 2461 (1963)

(10) H. Beck in Dynamical Properties of Solids, edited by G.K. Horton and A.A. Maradudin, (North-Holland, New York, 1975), Vol. II, pp. 225-226

(11) Y.P. Varshni and A. Konti, Phys. Rev. B 6, 1532 (1972)

(12) R.J. Hardy and S.S. Jaswal, Phys. Rev. B 3, 4385 (1971)

(13) S.S. Jaswal and R.J. Hardy, Phys. Rev. B 5, 753 (1972)

(14) H.J. Maris, Phys. Rev. B 24, 1205 (1981)

(15) M.J.P. Musgrave, Crystal Acoustics, (Holden-Day, San Francisco, 1970), p. 94ff

(16) K.H. Mueller and H.A. Fairbank in Proceedings of the 13th International Conference on Low Temperature Physics, LT13, Boulder, Colorado, edited by W.J. O'Sullivan, K.D. Timmerhaus and E.F. Hammel (Plenum, New York, 1974), Vol. 2, p. 90

(17) D.S. Greywall, Phys. Rev. B 16, 5127 (1977)

(18) R.H. Crepeau, O. Heybey, D.M. Lee and S.A. Strauss, Phys. Rev. A 3, 1162 (1971)

(19) J.P. Franck and R. Wanner, Phys. Rev. Lett. 25, 345 (1970)

CHAPTER 6

Special Points in Self-Consistent Phonon Calculations

A) Introduction

The self-consistent phonon (SCP) formalism (1-5) has proved quite successful in accurately describing various dynamical and bulk thermodynamic properties of anharmonic solids. Unfortunately these calculations are very time consuming since they require the calculation of averages of functions of eigenvectors and eigenvalues over the Brillouin zone at each iteration. Thus it would be desirable if a scheme could be found which would allow the averages to be rapidly but accurately evaluated. Here the convergence of the special points method for this purpose is studied. We test the scheme's applicability in the self-consistent harmonic (SCH) approximation for ^{36}Ar and Kr at various temperatures. Although this method has proved rapidly convergent in electronic problems (6-9), it has not been studied in this context.

The results given show that the scheme is rapidly convergent in the cases studied and thus should be of use in this context.

B) Special Points and the SCH Approximation

The SCP formalism has been derived by a number of authors (1-4). Here only the main equations are stated. The lowest order SCP theory is the SCH approximation in which the frequency of the phonon with wave vector q and branch λ is given by

$$\omega_{q\lambda}^2 = \sum_{\alpha,\beta} \mathcal{E}_\alpha(q,\lambda)\ D_{\alpha\beta}(q)\ \mathcal{E}_\beta(q,\lambda) \qquad (6\text{-}1)$$

where ε is the corresponding polarization vector and D is the dynamical matrix.

$$D_{\alpha\beta}(q) = \frac{1}{M} \sum_{\ell}{}' (e^{-iq\cdot R_\ell} - 1)\, \phi_{\alpha\beta}(0,\ell) \tag{6-2}$$

Here

$$\phi_{\alpha\beta}(0,\ell) = \langle \nabla_\alpha(0)\, \nabla_\beta(\ell)\, v(r_{0\ell}) \rangle$$

$$= [(2\pi)^3\, |\Lambda|]^{-\frac{1}{2}} \int d^3u\, \exp(-\tfrac{1}{2}\, u\Lambda^{-1}\, u)\, \nabla_\alpha(0)\, \nabla_\beta(\ell)\, v(r_{0\ell}) \tag{6-3}$$

is the averaged force constant between atoms 0 and ℓ. The displacement-displacement correlation function is taken to be Gaussian with width

$$\Lambda_{\alpha\beta}(0,\ell) = \langle u_\alpha(0,\ell)\, u_\beta(0,\ell) \rangle$$

$$= \frac{\hbar}{MN} \sum_{q\lambda} (1 - e^{iq\cdot R_\ell})\, \mathcal{E}_\alpha(q,\lambda)\, \mathcal{E}_\beta(q,\lambda)\, \coth\left[\frac{\beta\hbar\omega_{q\lambda}}{2}\right] \frac{1}{\omega_{q\lambda}} \tag{6-4}$$

where h, M, N and β have their usual meanings and Equations (6-1)-(6-4) are solved iteratively for a given lattice parameter and temperature. As discussed in Chapter 1 the convergence of the special points method depends critically on the form of the function to be averaged. Thus it was of considerable interest to examine its applicability in the context of SCH calculations. Here the special points method is used to reduce the average in Equation (6-4).

C) Results

In order to study the applicability of the special points method we performed calculations using the Baldereschi point (6) which satisfies Equations (1-8) and (1-9) for N = 2 and minimizes the m = 3 term, as well as sets of 2, 10 and 60 points generated using the method of Chadi and Cohen (7) which satisfy Equations (1-8) and (1-9) for N = 7,36 and about 150

respectively. In the tables these are referred to as (1B), (2), (10) and (60) respectively. In order to ensure that the 60 point scheme was in fact converged, calculations were also carried out using a 408 point scheme. These data are not shown as they are the same as the results using the 60 point scheme excepting changes of one in the last figure. As a further test the data were compared to SCH dispersion curves for Kr calculated under the same conditions by L.K. Moleko. The discrepancies found were negligible and could be explained by the different techniques for the numerical integration of the three dimensional integral in equation (6-3). In Table 6-1 phonon energies at some selected wave vectors for ^{36}Ar in the SCH approximation are given at temperatures of 10K, 35K, 55K and 82K. These were calculated using the Aziz-Chen potential (10). In Table 6-2 similar data for Kr at 10K and 115K calculated using the potential of Aziz (11) is presented.

From these data it may be observed that reasonable results (as compared to the fully converged calculation) are obtained using the one point scheme. Computationally this scheme is not substantially more complicated than a self-consistent Einstein approximation (12,13). To within the accuracy of the calculation performed, the scheme is fully converged using 10 points for the cases studied. This represents a substantial improvement over conventional methods for which it is not uncommon to use several hundred points.

D) Conclusion

In view of the above the special points method should be applicable to SCP calculations for the rare gas solids. These materials are characterized by the large contribution of nearest neighbour interactions to their lattice dynamics. It would be worthwhile to examine the applicability of the method to other classes of materials, in particular the alkali metals and the Wigner electron solid.

TABLE 6-1. SCH Phonon Energies for ^{36}Ar (meV)

Phonon	T(K)	Points 1B	2	10	60
0.2 0 0 T	10	1.8970	1.8944	1.8946	1.8946
	35	1.8604	1.8591	1.8596	1.8596
	55	1.8118	1.8102	1.8109	1.8109
	82	1.7193	1.7171	1.7180	1.7181
0.2 0 0 L	10	2.5235	2.5188	2.5192	2.5192
	35	2.4716	2.4686	2.4695	2.4696
	55	2.4043	2.4003	2.4016	2.4017
	82	2.2747	2.2693	2.2713	2.2715
0.6 0 0 T	10	4.9514	4.9446	4.9451	4.9452
	35	4.8560	4.8524	4.8535	4.8536
	55	4.7292	4.7247	4.7264	4.7266
	82	4.4876	4.4816	4.4839	4.4842
0.6 0 0 L	10	6.9756	6.9639	6.9648	6.9649
	35	6.8331	6.8265	6.8286	6.8288
	55	6.6467	6.6380	6.6411	6.6414
	82	6.2902	6.2785	6.2829	6.2833
1 0 0 T	10	6.1116	6.1031	6.1037	6.1038
	35	5.9938	5.9892	5.9906	5.9908
	55	5.8374	5.8316	5.8337	5.8339
	82	5.5394	5.5314	5.5344	5.5346
1 0 0 L	10	8.8373	8.8232	8.8243	8.8244
	35	8.6576	8.6498	8.6523	8.6525
	55	8.4214	8.4115	8.4152	8.4156
	82	7.9712	7.9578	7.9629	7.9634
0.2 0.2 0.2 T	10	2.4030	2.4000	2.4003	2.4003
	35	2.3593	2.3581	2.3585	2.3585
	55	2.3009	2.2993	2.2999	2.2999
	82	2.1888	2.1867	2.1878	2.1879
0.2 0.2 0.2 L	10	5.0578	5.0489	5.0497	5.0497
	35	4.9533	4.9479	4.9496	4.9497
	55	4.8170	4.8098	4.8123	4.8125
	82	4.5561	4.5463	4.5499	4.5502
0.5 0.5 0.5 T	10	4.0861	4.0811	4.0814	4.0815
	35	4.0119	4.0098	4.0104	4.0105
	55	3.9125	3.9098	3.9108	3.9109
	82	3.7220	3.7186	3.7203	3.7205

(TABLE 6-1 continued on next page)

TABLE 6-1, continued

Phonon	T(K)	Points 1B	2	10	60
0.5 0.5 0.5 L	10	8.7882	8.7732	8.7745	8.7745
	35	8.6070	8.5980	8.6008	8.6010
	55	8.3696	8.3578	8.3620	8.3623
	82	7.9163	7.9004	7.9063	7.9068
0.4 0.4 0.4 T_1	10	3.3253	3.3213	3.3216	3.3216
	35	3.2662	3.2647	3.2651	3.2652
	55	3.1869	3.1849	3.1856	3.1856
	82	3.0344	3.0319	3.0333	3.0334
0.4 0.4 0.4 T_2	10	5.1439	5.1365	5.1371	5.1371
	35	5.0428	5.0390	5.0403	5.0404
	55	4.9090	4.9043	4.9061	4.9063
	82	4.6541	4.6478	4.6503	4.6506
0.4 0.4 0.4 L	10	6.8212	6.8095	6.8105	6.8105
	35	6.6814	6.6743	6.6765	6.6767
	55	6.4984	6.4892	6.4924	6.4927
	82	6.1486	6.1360	6.1407	6.1411

TABLE 6-2. SCH Phonon Energies for Kr (meV)

Phonon	T(K)	Points 1B	2	10	60
0.2 0 0 T	10	1.3655	1.3647	1.3648	1.3648
	115	1.2269	1.2253	1.2259	1.2260
0.2 0 0 L	10	1.8089	1.8074	1.8075	1.8075
	115	1.6186	1.6141	1.6156	1.6157
0.6 0 0 T	10	3.5637	3.5616	3.5618	3.5618
	115	3.2023	3.1977	3.1995	3.1996
0.6 0 0 L	10	5.0131	5.0093	5.0097	5.0098
	115	4.4837	4.4746	4.4779	4.4782
1 0 0 T	10	4.3987	4.3961	4.3963	4.3964
	115	3.9527	3.9467	3.9490	3.9492
1 0 0 L	10	6.3563	6.3518	6.3523	6.3523
	115	5.6848	5.6750	5.6788	5.6791
0.2 0.2 0.2 T	10	1.7300	1.7290	1.7291	1.7291
	115	1.5626	1.5606	1.5614	1.5614
0.2 0.2 0.2 L	10	3.6318	3.6290	3.6293	3.6293
	115	3.2455	3.2380	3.2406	3.2409
0.5 0.5 0.5 T	10	2.9415	2.9399	2.9400	2.9400
	115	2.6570	2.6537	2.6549	2.6550
0.5 0.5 0.5 L	10	6.3181	6.3133	6.3138	6.3138
	115	5.6441	5.6320	5.6363	5.6367
0.4 0.4 0 T_1	10	2.3940	2.3927	2.3928	2.3928
	115	2.1666	2.1639	2.1648	2.1648
0.4 0.4 0 T_2	10	3.7016	3.6993	3.6996	3.6996
	115	3.3205	3.3158	3.3177	3.3179
0.4 0.4 0 L	10	4.9014	4.8977	4.8981	4.8981
	115	4.3823	4.3725	4.3760	4.3763

REFERENCES

(1) N. Boccara and G. Sharma, Physics 1, 219 (1965)

(2) T.R. Koehler, Phys. Rev. Lett. 17, 89 (1966)

(3) H. Horner, Z. Phys. 205, 72 (1967)

(4) H.R. Glyde, Can. J. Phys. 52, 2281 (1974)

(5) H.R. Glyde and M.L. Klein, Crit. Rev. Solid State Sci. 2, 181 (1971)

(6) A. Baldereschi, Phys. Rev. B 7, 5212 (1973)

(7) E. Holzschuh, Phys. Rev. B 28, 7346 (1983)

(8) D.J. Chadi and M.L. Cohen, Phys. Rev. B 7, 692 (1973)

(9) J.D. Joannopoulos and M.L. Cohen, J. Phys. C 6, 1572 (1973)

(10) R.A. Aziz and H.H. Chen, J. Chem. Phys. 67, 5719 (1977)

(11) R.A. Aziz, Mol. Phys. 38, 177 (1979)

(12) H.R. Glyde and G.H. Keech, Ann. Phys. (N.Y.) 127, 330 (1980)

(13) L.K. Moleko and H.R. Glyde, Phys. Rev. B 27, 6019 (1983)

CHAPTER 7

Madelung Constants for Layer Crystals

The transition metal dichalcogenides are a family of layer compounds, each layer consisting of two close packed sheets of chalcogen atoms sandwiching a layer of transition metal atoms. Within a layer two arrangements occur: the trigonal prismatic (TP) and octahedral (OCTA) coordinations (referring to the metal atoms). The former structure is favoured by the group 6B compounds and the later by the 4B compounds. The group 5B compounds may be found in either structure.

In order to understand why one structure is favoured over the other it is necessary to examine the energy difference between them. We note that the interlayer forces are weak in these materials and thus probably do not play a role in distinguishing the energies of the two structures.

An important component of the energy difference between the two structures is the Madelung energy. Recently Shen and Liang (1) calculated Madelung constants for the two structures using a real space scheme. This method converged slowly and as a result they were only able to report values accurate to about 1% leading to an uncertainty of about 10% in the electrostatic energy difference between the two structures. Here Ewald's method (2) is applied to this problem in order to obtain a more accurate result.

This method is in my view more intuitive and computationally easier to implement than the Poisson summation formula for planes (3) used by Johnston and Scholl (4) when they treated this problem concurrently with us (5). In particular using Ewald's method the entire layer is treated at once rather than as an assembly of planes of atoms and the evaluation of modified Bessel

functions is avoided.

To apply Ewald's method to layers one may note that the energy of an isolated layer can be obtained by considering the energy of a stack of layers in the limit as the interlayer separation becomes infinite. To this end we consider a unit cell defined by $\underline{a}$, $\underline{b}$, and $\underline{c}$ with $\underline{a}$ and $\underline{b}$ in the plane of the layer and $\underline{c}$ perpendicular to the layers. The z axis is also taken to be perpendicular to the layers so that $\underline{c} = (0,0,c)$ where c is the interlayer spacing. By Ewald's method we have

$$\sum_{\ell} \frac{1}{|\ell-r|} = \frac{\pi}{v_c} \frac{1}{G^2} \sum_{g} \exp(ig\cdot r) \frac{\exp(-g^2/4G^2)}{g^2/4G^2} + \sum_{\ell} \frac{1}{|\ell-r|} \operatorname{erfc}(G|\ell-r|) \qquad (7\text{-}1)$$

and

$$\sum_{\ell\neq 0} \frac{1}{|\ell|} = \frac{\pi}{v_c} \frac{1}{G^2} \sum_{g} \frac{\exp(-g^2/4G^2)}{g^2/4G^2} + \sum_{\ell\neq 0} \frac{1}{|\ell|} \operatorname{erfc}(G|\ell|) - \frac{2G}{\sqrt{\pi}} \qquad (7\text{-}2)$$

where v_c is the volume of the unit cell, $\underline{r}$ is an arbitrary vector, $\underline{\ell}$ is a lattice vector, $\underline{g}$ is a reciprocal lattice vector and G is an arbitrary parameter chosen to accelerate the convergence of the sums. In the limit where c becomes infinite the real space sums reduce to sums over a single layer. For the reciprocal space sums let $\underline{g} = (g_x, g_y, g_z)$ and $\underline{r} = (r_x, r_y, r_z)$ so that

$$\frac{4\pi}{v_c} \sum_{g} \exp(ig\cdot r) \frac{\exp(-g^2/4G^2)}{g^2} = \frac{4\pi}{v_c} \sum \frac{}{g_x g_y} \cos(g_x r_x + g_y r_y)$$

$$\times \exp[-(g_x^2 + g_y^2)/4G^2] \sum_{g_z} \frac{\exp(-g_z^2/4G^2)\cos(g_z r_z)}{g_x^2 + g_y^2 + g_z^2} \qquad (7\text{-}3)$$

Letting $v_c = A\,c$ and noting that the sum over g_z becomes an integral in the limit

$$\frac{4}{A} \sum_{g_x g_y} \cos(g_x r_x + g_y r_y) \exp\left[-(g_x^2 + g_y^2)/4G^2\right] \int_0^\infty \frac{\exp(-g_z^2/4G^2)\cos(g_z r_z)}{g_x^2 + g_y^2 + g_z^2} dg_z$$

$$= \frac{4}{A} \sum_{g_x g_y} \cos(g_x r_x + g_y r_y) \exp\left[-(g_x^2 + g_y^2)/4G^2\right] I\left(r_z, \frac{1}{4G^2}, \sqrt{g_x^2 + g_y^2}\right) \tag{7-4}$$

The integral can be evaluated analytically (6).

$$I(\alpha,\beta,\gamma) = \frac{\pi \exp(\beta\gamma^2)}{4\gamma}$$

$$x \left[\exp(-\alpha\gamma)\ \mathrm{erfc}(\gamma\sqrt{\beta} - \alpha/2\sqrt{\beta}) + \exp(\alpha\gamma)\ \mathrm{erfc}(\gamma\sqrt{\beta} + \alpha/2\sqrt{\beta})\right] \tag{7-5}$$

The reciprocal space sums in equations (1) and (2) are identical for $\underline{r}=0$. The $g_x=g_y=0$ term in the sum of equation (4) diverges, as might be expetted from the long range of the Coulomb potential. However, since the layers are electrically neutral, in evaluating the Madelung constant the divergences cancel and only the remaining part is required. We note that for small γ

$$I(\alpha,\beta,\gamma) \simeq \left(\frac{\pi}{4\gamma}\right) \{2 - \gamma[2\alpha\ \mathrm{erfc}\ (\alpha/2\sqrt{\beta}) + 4\sqrt{\beta/\pi}\ \exp(-\alpha^2/4\beta)]\} \tag{7-6}$$

so that in evaluating the Madelung constant $I(\alpha,\beta,\gamma)$ is to be replaced by $I_o(\alpha,\beta,\gamma)$ in the $g_x=g_y=0$ terms of the reciprocal lattice sums where

$$I_o(\alpha,\beta,0) = \left(-\frac{\pi\alpha}{2}\right) \mathrm{erf}\left(\frac{\alpha}{2\sqrt{\beta}}\right) - \sqrt{\pi\beta}\ \exp(-\alpha^2/4\beta) \tag{7-7}$$

Applying this to the ideal structures of Table 7-1, Madelung constants of 10.166078 and 12.334424 are obtained for the TP and OCTA structures respectively. These values are much more accurate than those obtained by Shen and Liang.

TaS_2, which is of particular interest because it occurs in both structures,

has been studied in detail by Shen and Liang. Because in this case the two structures must have similar energies they set the sum of the electrostatic and electron band structure energy differences between the two structures to be zero. It was of interest to re-examine their conclusions using the Madelung constants obtained above. Following them the difference in the electron band structure energy between the OCTA and TP structures is taken to be in the range 1.0 eV to 1.5 eV. Thus assuming the Ta charge to be the same for both structures the Ta ions carry a charge of between 0.95e and 1.2e. This overlaps the lower end of the range 0.97e to 1.5e given by Shen and Liang and is considerably narrower than it. Allowing the electron band structure energy difference to be as high as 2.0 eV increases the upper limit of the the range obtained here to about 1.35e.

By the argument given by Lucovsky et. al. (7) that the TP structure favours the formation of covalent bonds in these materials it may be expected that the Ta charge might be somewhat larger in the OCTA structure. This implies that the actual charges should lie in the lower end of the range given above or perhaps somewhat below the lower limit.

TABLE 7-1 Ideal Structures for the MX_2 Compounds
(lattice constant = 1.0)

	Trigonal prismatic	Octahedral
a	(1,0,0)	(1,0,0)
b	(-1/2,√3/2,0)	(-1/2,√3/2,0)
r_m	(0,0,0)	(0,0,0)
r_{x1}	(0,1/√3,1/2)	(0,1/√3,1/√6)
r_{x2}	(0,1/√3,-1/2)	(1/2,1/2√3,-1/√6)

REFERENCES

(1) T.H. Shen and W.T. Liang, J. Phys. C 16, L883 (1983)

(2) P.P. Ewald, Ann. Phys., Lpz. 64, 253 (1921)

(3) C.A. Scholl, Proc. Phys. Soc. 92, 434 (1967)

(4) N.A. Johnston and C.A. Sholl, J. Phys. C 17, L73 (1984)

(5) D. Singh and Y.P. Varshni, J. Phys. C 17, L321 (1984)

(6) A. Erdelyi et. al., Tables of Integral Transforms, vol. 1 (McGraw Hill, New York, 1954), page 15

(7) G. Lucovsky, R.M. White, J.A. Benda and J.F. Revelli, Phys. Rev. B 7, 3859 (1973)

APPENDIX A

A Brief Overview of the Total Energy Program

A) The SCF loop

As previously discussed in Chapters 1 and 2, the calculation of the total energies and charge densities proceeded by the iteration of the Kohn-Sham equations, (1-1) and (1-2), to self-consistency, the SCF loop consisting of the construction of an output charge density based on a given input charge density. Accordingly, for each iteration the Hamiltonian matrix (see equations 2-2 and 2-3) was constructed at each special point in the irreducible wedge of the Brillouin zone, using the input charge density and the results from Chapter 2, and its first few eigenvectors and eigenvalues calculated. Using the occupied eigenvectors an unsymmetrized valence charge density was accumulated in reciprocal space. The weighted sum of the eigenvalues of the occupied states was also accumulated to be used in the calculation of the total energy. Once the sum over the irreducible wedge of the Brillouin zone was completed the resulting charge density was symmetrized to obtain the output charge density. This was done using the space group elements of the crystal under consideration, which relate different Fourier components of the charge density within a given star (a set consisting of a reciprocal lattice vector and those vectors which may be obtained by applying operations in the point group of the crystal in question to it). Having thus obtained the output charge density, it was Fourier transformed into real space in order to calculate the Fourier components of the exchange-correlation potential and the exchange-correlation energy, needed for the evaluation of the total energy as discussed in Chapter 2. Next the Ewald term, γ_{Ewald}, was calculated for the lattice in question and the total energy evaluated using

Equation (2-4), thereby completing the iteration in question.

B) Potential Mixing

As discussed above an input charge density is required for each pass through the SCF loop. The most obvious choice would be to use the output charge density and corresponding potential from the previous iteration as the input to the current pass. However this scheme does not converge sufficiently rapidly to obtain self-consistency in a reasonable number of iterations. In order to obtain faster convergence a simple mixing of the inputs and outputs was used.

$$(n^{in})_{i+1} = \alpha (n^{in})_i + (1-\alpha)(n^{out})_i \qquad \text{(A-1)}$$

$$(V_{xc}^{in})_{i+1} = \alpha (V_{xc}^{in})_i + (1-\alpha)(V_{xc}^{out})_i \qquad \text{(A-2)}$$

In Equations (A-1) and (A-2) the subscripts, i and i+1 refer to the i'th and (i+1)'th iterations respectively, α being a constant adjusted to obtain rapid convergence. It may be noted from (A-1) and (A-2) that the input exchange-correlation potential to be used in constructing the Hamiltonian matrices in the (i+1)'th iteration is not that corresponding to the input charge density except for the special case, $\alpha=0$, or at self-consistency. This is because the exchange-correlation potential is a non-linear function of the charge density. The reason for using (A-2) instead of calculating v_{xc} from the charge density at the start of each iteration is that it was found that this scheme did not seem to change the number of iterations required to obtain self-consistency while permitting the saving of one Fourier transform per iteration.

About four iterations were typically required to converge the total energy to 10^{-6} Ry. per unit cell using a mixing parameter, α, of about 0.2 with the input for the first iteration constructed as discussed below.

The simple mixing scheme used in the calculations described here would probably not have been adequate for calculations on a system with a large unit cell where a more powerful technique such as Broyden's method (1) might have been necessary. The success of the scheme used was probably due to the fact that for the lattices considered the unit cells were fairly small and thus there were no small G Fourier components of the charge density except of course n(G=0) which is fixed by the condition of overall electrical neutrality. This is significant because the Coulomb potential diverges as G^{-2}.

C) The Input Charge Density

In order for the self-consistent calculation to converge in a reasonable number of iterations it was necessary to have a reasonable estimate of the charge density to use as an input to the first pass through the SCF loop. In this calculation many of the required input charge densities were obtained from a calculation using a small basis consisting of about half the number of plane waves used in the main calculation. This small basis set calculation typically used seven or eight iterations to find a stable solution starting from an initial uniform charge distribution, the seven or eight iterations using about as much computer time as would have been required to do two iterations with the full basis set.

In some cases which involved small distortions of a unit cell for which a self-consistent charge density had already been obtained an alternate method

of obtaining an input charge density for the first iteration was used. This consisted of applying the same homogeneous deformation to charge density of the existing calculation as was to be applied to the unit cell. For the sort of distortions this scheme was used for (of the order of 1-2%) input charge densities of sufficient quality to reach the required degree of self-consistency in about five iterations were obtained. Thus in the cases for which it was applicable the use of this scheme resulted in the saving of a significant amount of computer time.

REFERENCE

(1) C.G. Broyden, Math. Comput. 19, 577 (1965)

APPENDIX B

The Fits Used in the Total Energy Calculations

In order to obtain structural properties from calculations of the total energy of a solid subjected to various distortions it is necessary to fit these to appropriate parameterized functions, the desired structural constants being obtained from the parameters. In this appendix the fits used are described and their accuracy discussed. In addition some total energies which were used in the fits are given in Table B-1. It should be noted that it was found that the total energies tended to converge more slowly than the charge densities and structural properties as the size of the basis was increased. The values given are not fully converged with respect to the size of the basis.

i) The lattice constant, bulk modulus and its pressure derivative

As previously discussed, the lattice constant, bulk modulus and the pressure derivative of the bulk modulus were calculated by fitting the total energies calculated at six volumes to Murnaghan's equation of state,

$$E = \frac{B_0 V}{B_0'(B_0'-1)}\left[B_0'\left(1-\frac{V_0}{V}\right)+\left(\frac{V_0}{V}\right)^{B_0'}-1\right]+E(V_0) \qquad \text{(B-1)}$$

In order to test the quality of this procedure several additional fits were performed using different subsets of the total energy data obtained by deleting various points. As a further test a fit was also performed using eight values of the total energy, the two additional values being at greatly reduced volumes ($\Omega/\Omega_o = 0.762$ and $\Omega/\Omega_o = 0.804$). Based on these tests it is

estimated that the errors due to the fit employed are most likely less than five in the last digits of the value of the bulk modulus using 731 plane waves, the error in the other quantities probably being beyond the last figure quoted.

ii) The shear elastic constant, c_s

The shear elastic constant was calculated from the total energy changes associated with 1% stretches and compressions of the unit cell along the z-axis with compensating changes in the other two axes designed to keep the volume of the unit cell unchanged. As expected, it was found that to within the numerical noise level (estimated to be less than 2×10^{-6} Ry./cell) the energy of a unit cell stretched by 1% along the z-axis was the same as that when compressed by the same amount. Accordingly, the shear constants were determined by fitting the energies of the deformed and undeformed unit cells to a quadratic, $E = E_o + au^2$, where E_o is the energy of the undeformed unit cell, u is the deformation (Here the strains are $e_{33}=u$ and $e_{11}=e_{22}=(1+u)^{-1/2}-1$) and a is the parameter to be determined. Thus

$$c_{11} - c_{12} = 1.961 \times 10^{13}\, a/\Omega \qquad \text{(B-2)}$$

where a is in Ry./cell, the cell volume is in atomic units and the elastic constants are in GPa. In order to test the reliability of the technique; a more complicated fit was undertaken at $\Omega/\Omega_o = 0.935$. This consisted of fitting the energies of the undistorted unit cell and unit cells with 1% and 2% stretches along the z-axis to a polynomial with both second and fourth order terms. Based on this test it is estimated that the fitting errors in the values given are probably less that 1 in the last figure given.

iii) The phonon frequency, ω_{TO}, and force constant, k_{xyz}

The transverse optical phonon frequencies and associated anharmonic force constants were calculated from changes in the energy resulting from alterations in the spacing between the cation and anion sublattices along a bond direction. In particular the energies for a stretch of about 1% and an equal compression were used, these being fitted to a cubic, $E = E_o + au^2 + bu^3$, E_o being the energy at u=0 (Here $u=t-t_o$, the sublattice separation being (2t,2t,2t) and t_o the equilibrium value of t.) and a and b the parameters. Thus

$$\omega_{TO} = 2.795 \times 10^{14} \sqrt{\frac{a}{m}} \qquad \text{(B-3)}$$

$$k_{xyz} = 22.95\, b \qquad \text{(B-4)}$$

where m is the reduced mass in atomic mass units, the energies are in Ry./cell, the frequency is in s^{-1} and k_{xyz} is in $eV/\AA^2$. As a test the same fit was performed at the reduced volume, $\Omega/\Omega_o = 0.935$, using energies for several displacements ranging between -2% and +2%. Based on this, the fitting errors in the frequencies were probably less than two in the last digit given whereas the fitting errors in the anharmonic force constants are likely somewhat larger being of the order of 5%.

iv) c_{44} and the internal strain parameter

The calculations of the internal strain parameters and c_{44} were those which involved the most computer time. This is as a result of the large number of distortions for which total energies were calculated and the fact that these were among the lowest symmetry distortions treated here. The calculations were performed by fitting the total energies of unit cells with various

sublattice spacings along a bond direction to cubics (This was done using a least squares algorithm.), the calculations differing from those used for the transverse optical phonon frequencies in that in the present case the unit cells being considered were simultaneously subjected to trigonal stretches of 0.5% or 1% along the same direction as the sublattice displacements. For each value of the trigonal deformation total energies for about five sublattice spacings were used in the fits. From the positions of the minima for each trigonal stretch an effective internal strain parameter for that stretch was calculated, the internal strain parameter quoted being calculated by linearly extrapolating the effective internal strain parameters corresponding to the 0.5% and 1% stretches to the undeformed limit. These effective internal strain parameters were 0.52 and 0.49 respectively at the free volume and 0.39 and 0.41 respectively at the reduced volume. The energy minima for the two trigonal deformations were then fitted to a cubic, $E = E_o + ae^2 + ce^3$, again using a least squares algorithm. Here e is the trigonal deformation (the strains are $e_{ij}=e$, for $i\neq j$ and $e_{ii}=0$), a and c are parameters, E is the energy minimum at e, determined from the fit above, and E_o is the energy of the undistorted unit cell. The elastic constants c_{44} were then calculated from the resulting values of the parameter, a. In order to estimate the reliability of this procedure it was repeated with subsets of the data, constructed by leaving out one or two values of the total energy. Because of the limited number of total energies available it was not feasible to test this procedure more thoroughly. From the tests performed it seems probable that the fitting errors in the internal parameters are less than three in the last digit quoted and that the errors in c_{44} are less than one in the last digit quoted. The fitting errors in the anharmonic force constant k_{xyz} for non-zero trigonal deformations are estimated to be of the order of 5%. It was

found that the cubic constant, c, was highly unreliable and therefore no attempt was made to use this number.

It should be noted that more reliable calculations of the internal strain parameters and the elastic constants, c_{44}, could probably have been carried out by using the force and stress theorems (1). This is because using these results several pieces of information can be obtained at each deformation, instead of just the energy, thereby reducing the number of distortions which need be considered. Thus given a fixed amount of computer resources more attention could presumably be paid to questions of convergence if these were used. Also the forces and stresses are linear in the strain whereas the energy is quadratic. Thus in view of the small energy differences treated here it seems likely that the use of these theorems could yield more accurate results. The force and stress theorems were not used because the necessary expressions using the non-local pseudopotentials employed here are somewhat unwieldy.

TABLE B-1 Some selected values of the total energy of InSb using 531 plane waves. The energies are in Ry./cell. The sublattice spacings are given in terms of t (in a.u.) where the sublattices are separated by (2t,2t,2t).

Calculation			Ω/Ω_o	Energy	
Cubic (undistorted)			1.035	-15.329593	
			1.000	-15.333358	
			0.985	-15.334552	
			0.938	-15.336470	
			0.935	-15.336474	
			0.892	-15.334747	
			0.847	-15.328687	
			0.804	-15.317482	
			0.762	-15.300200	
			0.722	-15.275754	
Trigonal	e=0	t=1.435	0.847	-15.328387	(at equilibrium
	e=0	t=1.465	0.847	-15.328408	t=1.45)
	e=0	t=1.46	0.892	-15.334476	(at equilibrium
	e=0	t=1.49	0.892	-15.334493	t=1.475)
	e=0	t=1.48	0.935	-15.336098	(at equilibrium
	e=0	t=1.483515	0.935	-15.336230	t=1.4985)
	e=0	t=1.49	0.935	-15.336396	
	e=0	t=1.51	0.935	-15.336337	
	e=0	t=1.513485	0.935	-15.336243	
	e=0	t=1.52	0.935	-15.336024	
	e=0	t=1.51	0.985	-15.334330	(at equilibrium
	e=0	t=1.54	0.985	-15.334342	t=1.525)
	e=0	t=1.517175	1.000	-15.333135	(at equilibrium
	e=0	t=1.5248375	1.000	-15.333303	t=1.5325)
	e=0	t=1.5401625	1.000	-15.333305	
	e=0	t=1.547825	1.000	-15.333147	
	e=.5%	t=1.4985	0.935	-15.336242	
	e=.5%	t=1.502	0.935	-15.336282	
	e=.5%	t=1.505	0.935	-15.336292	
	e=.5%	t=1.508	0.935	-15.336291	
	e=.5%	t=1.511	0.935	-15.336267	
	e=1%	t=1.4985	0.935	-15.335670	
	e=1%	t=1.505	0.935	-15.335802	
	e=1%	t=1.51	0.935	-15.335863	
	e=1%	t=1.515	0.935	-15.335872	
	e=1%	t=1.52	0.935	-15.335831	

(continues on next page)

TABLE B-1 (continued)

	e=.5% t=1.5325	1.000	-15.333118
	e=.5% t=1.5375	1.000	-15.333179
	e=.5% t=1.5425	1.000	-15.333194
	e=.5% t=1.5475	1.000	-15.333163
	e=1% t=1.5325	1.000	-15.332478
	e=1% t=1.5375	1.000	-15.332587
	e=1% t=1.5425	1.000	-15.332666
	e=1% t=1.5475	1.000	-15.332699
	e=1% t=1.5475	1.000	-15.332688
Tetragonal	stretch=+2%	0.935	-15.336137
	stretch=+1%	0.935	-15.336391
	stretch=+1%	1.000	-15.333273

REFERENCE

(1) See for example, O.H. Nielsen and R.M. Martin, to be published in Electronic Structure, Dynamics and Quantum Structural Properties of Condensed Matter (1985)

FIGURES

Figure 3-1

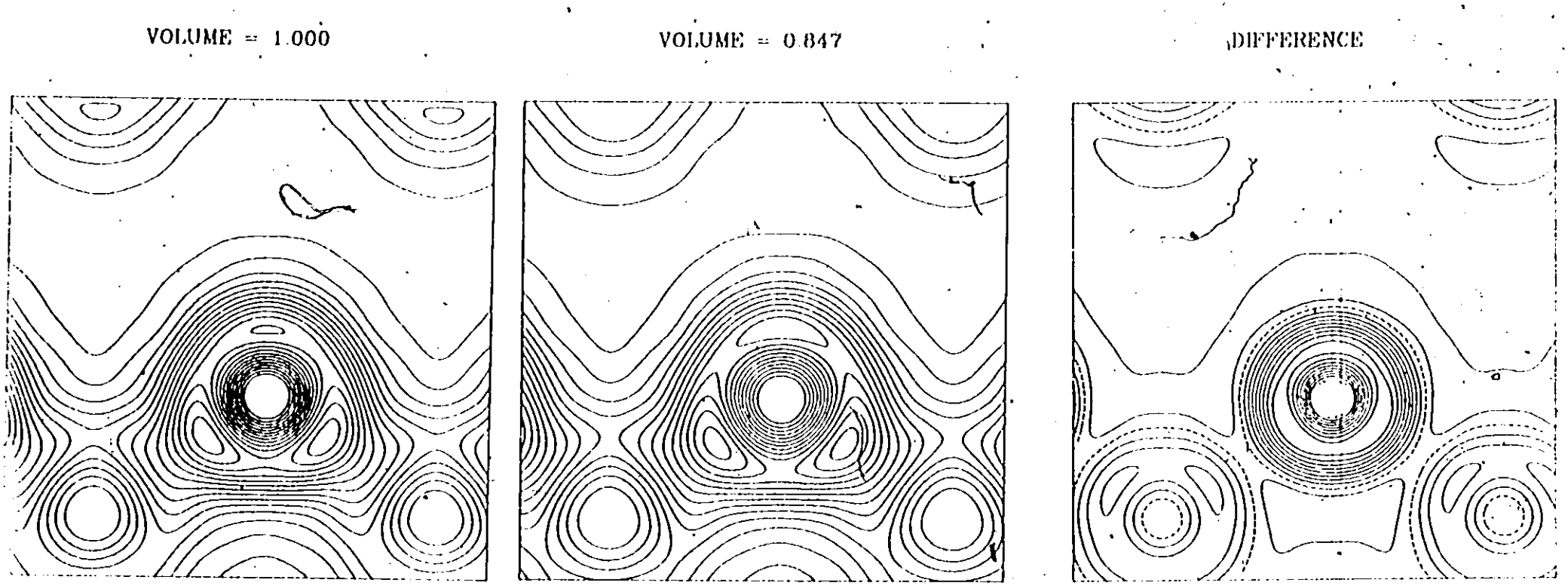
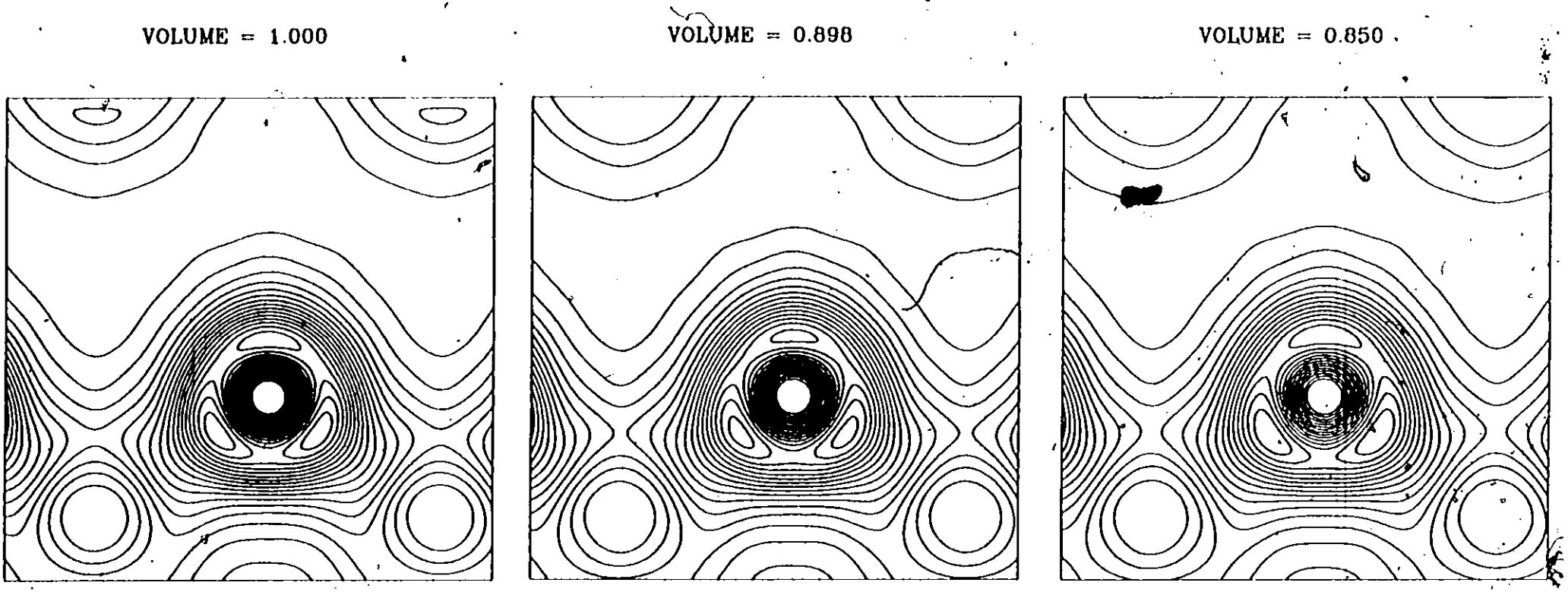


Figure 3-2



VOLUME = 0.935

VOLUME = 1.000

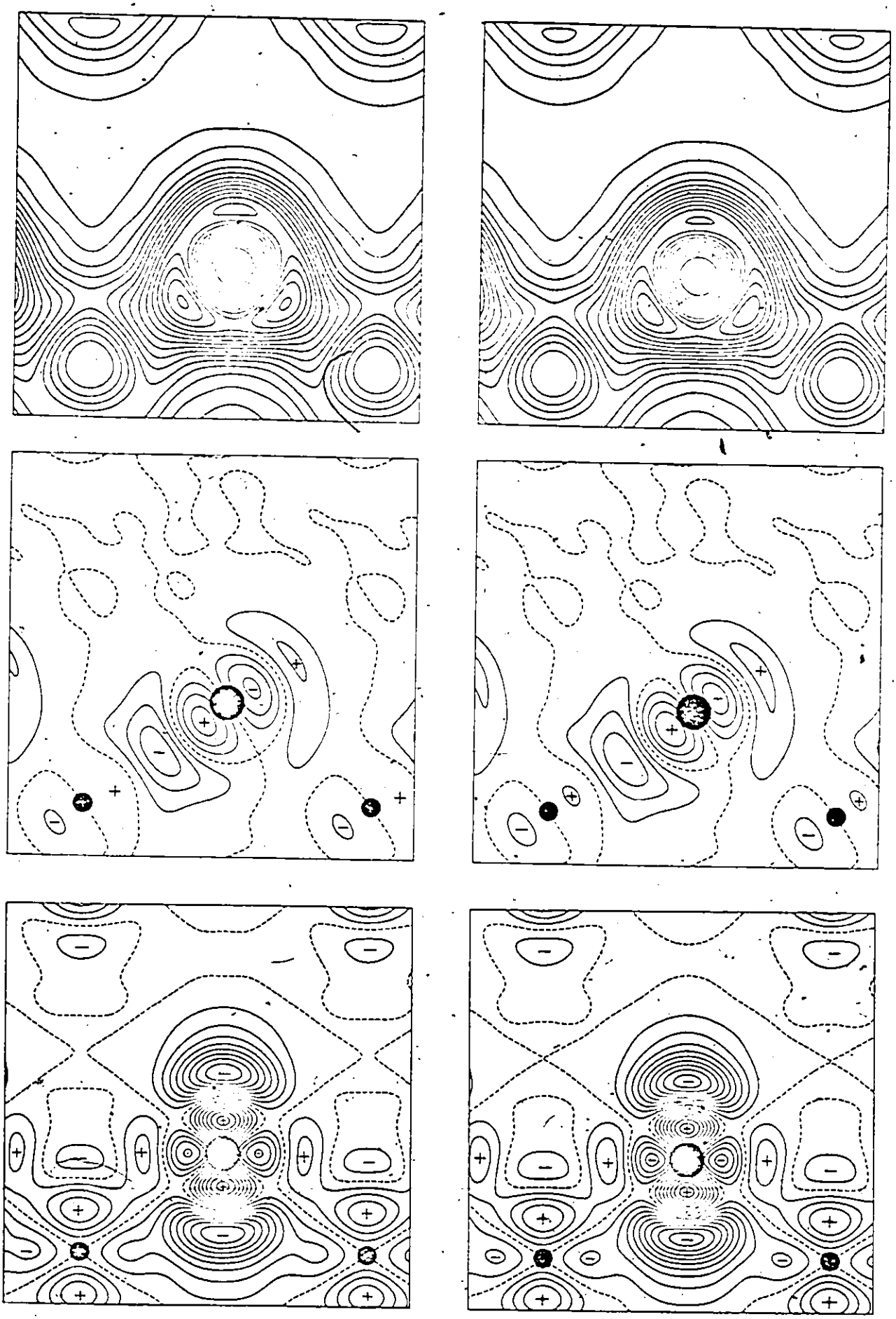


Figure 4-1

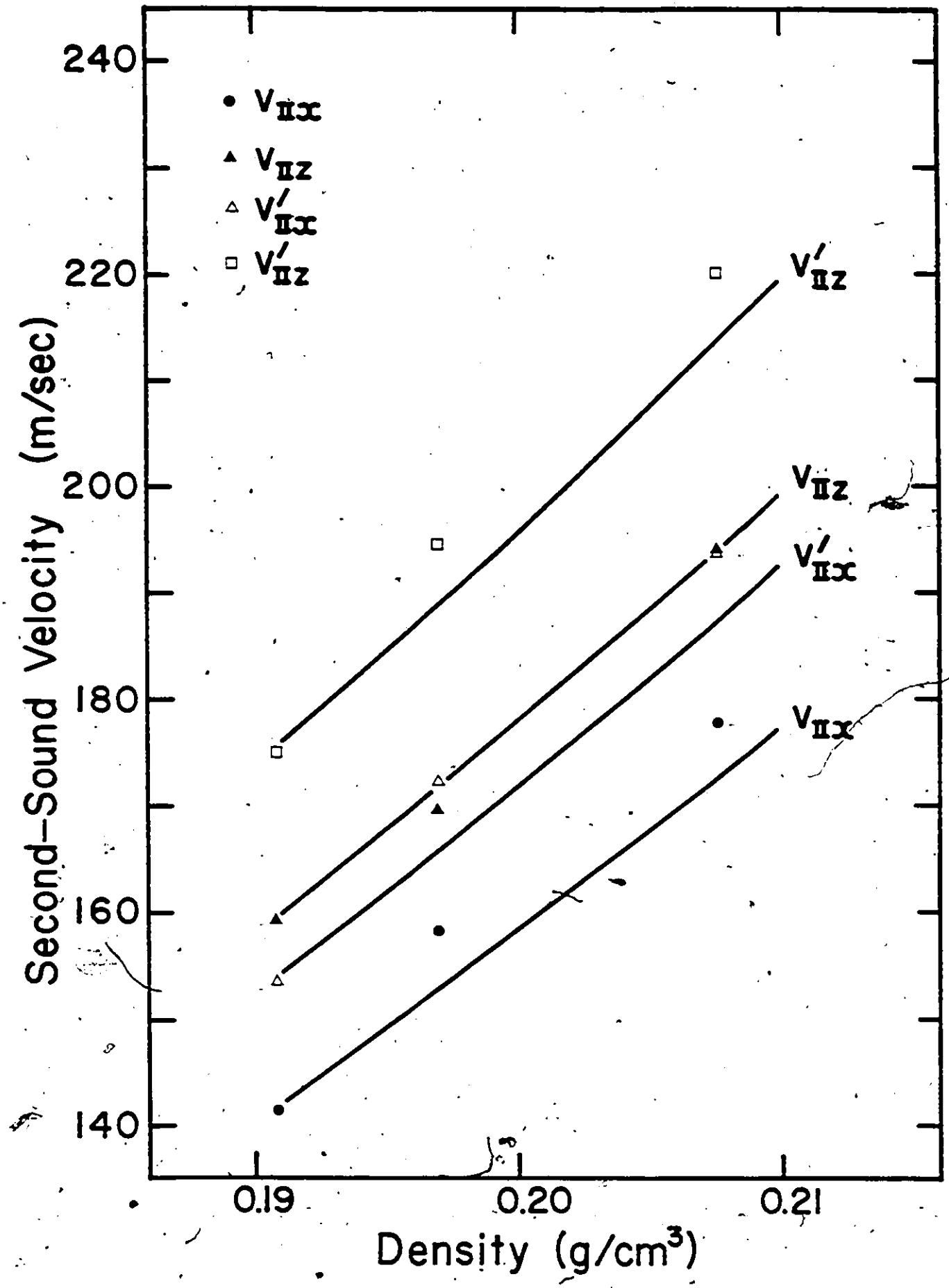

240
220
200
180
160
140
Second-Sound Velocity (m/sec)
0.19
0.20
0.21
Density (g/cm³)
• V_IIx
▲ V_IIz
△ V'_IIx
□ V'_IIz
V'_IIz
V_IIz
V'_IIx
V_IIx

Figure 5-1

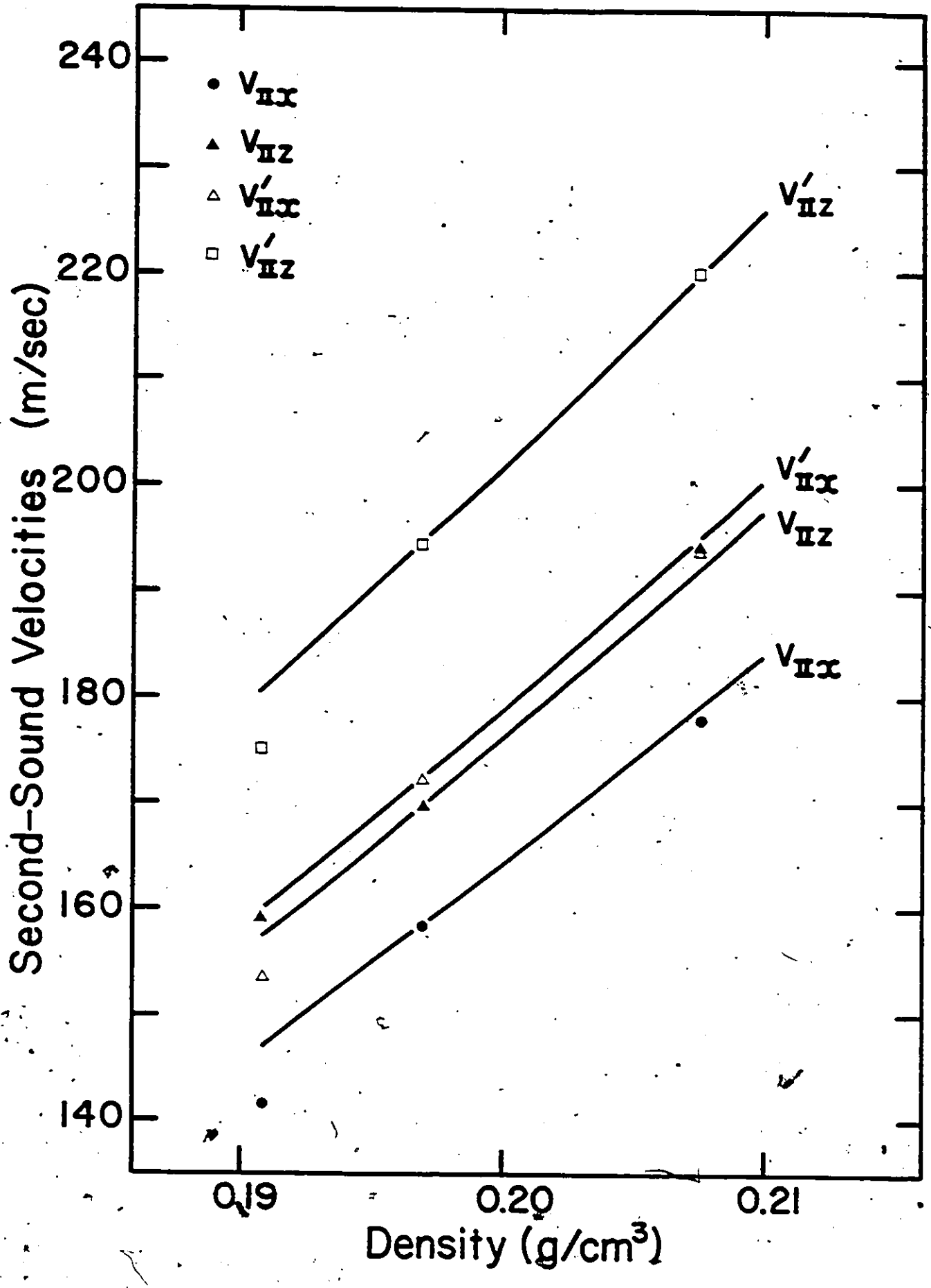

240
220
200
180
160
140
Second-Sound Velocities (m/sec)
0.19
0.20
0.21
Density (g/cm³)
V_IIx
V_IIz
V'_IIx
V'_IIz
V'_IIz
V'_IIx
V_IIz
V_IIx

Figure 5-2

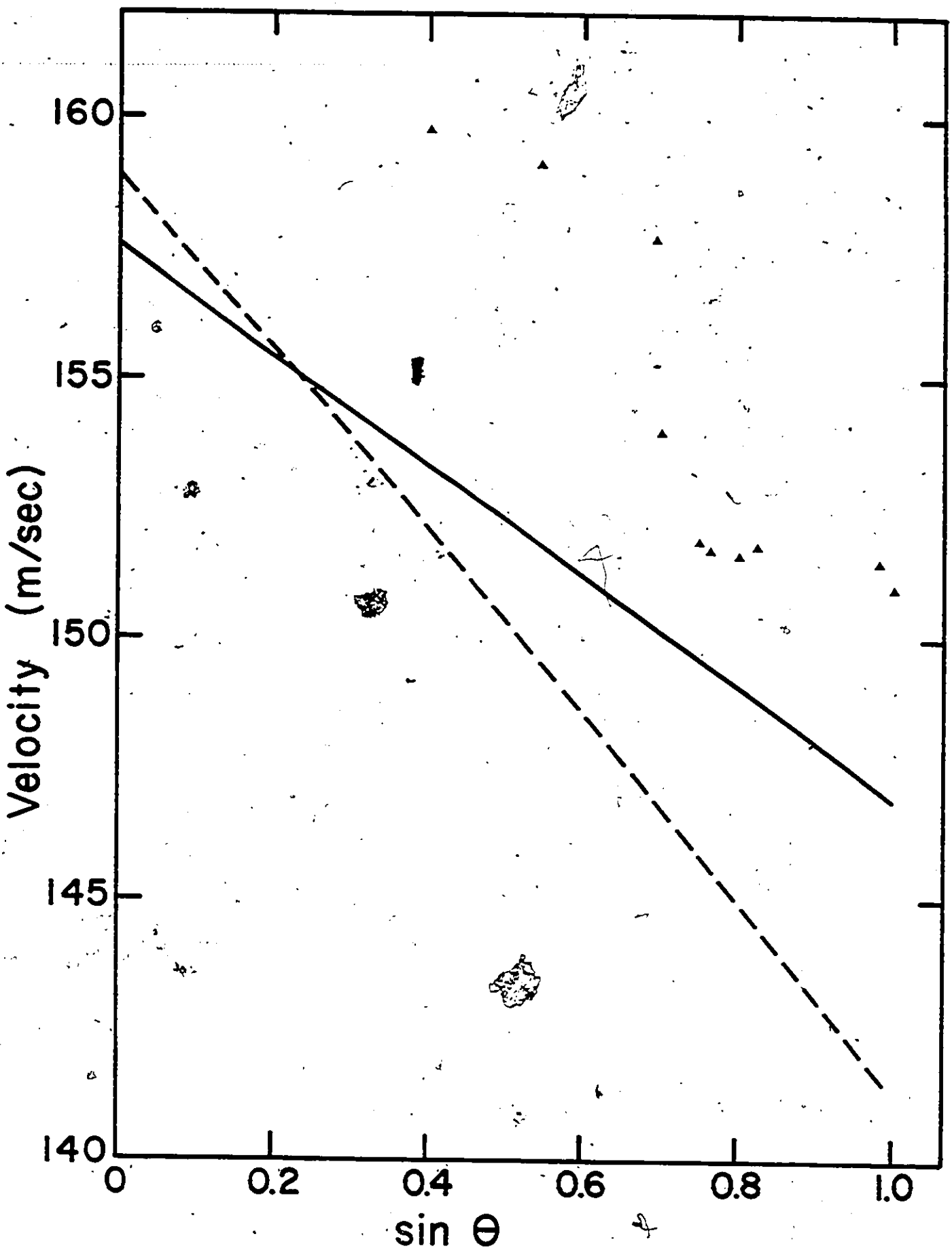


Figure 5-3

PUBLICATIONS

(1) E. Fortin and D. Singh, 'Polarization and intensity fluctuations in He-Ne lasers', Am. J. Phys. 49, 891 (1981)

(2) D. Singh and Y.P. Varshni, 'Debye temperatures for hexagonal crystals', Phys. Rev. B 24, 4340 (1981)

(3) D. Singh and Y.P. Varshni, 'X-ray debye temperature for hexagonal crystals', Acta Cryst. A38, 854 (1982)

(4) D. Singh and Y.P. Varshni, 'Second sound velocities in hexagonal crystals', Phys. Rev. B 25, 7376 (1982)

(5) D. Singh and Y.P. Varshni, 'Accurate eigenvalues and oscillator strengths for the exponential-cosine screened Coulomb potential', Phys. Rev. A 28, 2606 (1983)

(6) D. Singh and Y.P. Varshni, 'Comparative study of the bound states of static screened Coulomb and cut-off Coulomb potentials', Phys. Rev. A 29, 2895 (1983)

(7) D. Singh and Y.P. Varshni, 'Madelung constants for layer compounds', J. Phys. C 17, L321 (1984)

(8) D. Singh and Y.P. Varshni, 'Applicability of the special points method in lattice dynamics', Phys. Rev. B 30, 6917 (1984)

(9) D. Singh and Y.P. Varshni, 'Bound state properties of the Lennard-Jones 8-6 potential', Physica 132C, 235 (1985)

(10) D. Singh and Y.P. Varshni, 'Valence charge density in InSb', Phys. Rev. B 31, 8252 (1985)

(11) Y.P. Varshni and D. Singh, 'Common lines in the rest-frame absorption line spectra of QSOs?', Astrophys. Space Sci. 109, 149 (1985)

(12) D. Singh and Y.P. Varshni, 'Valence charge density in Indium Arsenide', Solid State Commun. 55, 549 (1985)

(13) D. Singh, Y.P. Varshni and R. Dutt, 'Bound eigenstates for two truncated Coulomb potentials', Phys. Rev. A 32, 619 (1985)

(14) D. Singh and Y.P. Varshni, ' Structural properties and polarization charge densities of InSb', Phys. Rev. B (in press 1985)

The Computer Code for the Total Energy Calculations

```
LEVEL 1.4.0 (OCT 1984)          VS FORTRAN          DATE: SEP 17, 1985    TIME: 08:15:15                PAGE:    1

REQUESTED OPTIONS (EXECUTE): NOMAP NOXREF NOOBJECT

OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG,  SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(60)      CHARLEN(500)

                *....*...1.....2.....3.....4.....5.....6.....7.*.....8

                C
                C
                C   TOTAL ENERGY PROGRAM USING
                C    BHS PSEUDOPOTENTIALS AND G=G'
                C    TO ADD MORE CHARGE-DENSITY COMPONENTS USE PROGRAM LDA2
                C
                C   PROGRAM WRITEN BY D. SINGH MAY 1983 - AUGUST 1984
                C
ISN         1           IMPLICIT REAL*8(A-H,O-Z)
ISN         2           DIMENSION AR(533,533),AI(533,533),VR(533,5),VI(533,5),EIG(5)
ISN         3           CALL SCF(AR,AI,VR,VI,EIG)
ISN         4           STOP
ISN         5           END

*STATISTICS*   SOURCE STATEMENTS = 5, PROGRAM SIZE = 4588478 BYTES, PROGRAM NAME = MAIN      PAGE:    1.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**MAIN** END OF COMPILATION 1 ******
```

```
LEVEL 1.4.0 (OCT 1984)          VS FORTRAN          DATE: SEP 17, 1985    TIME: 08:15:16                    PAGE:    2

OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NODUMP AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(64)  .   CHARLEN(500)

                 *....*...1.....2.....3.....4.....5.....6.....7.*.....8

ISN         1.          BLOCK DATA
ISN         2           IMPLICIT REAL*8(A-H,O-Z)
ISN         3           COMMON/DENS/RH(533),RR(750),RRT(750),EX
ISN         4           COMPLEX*16 RRT,RR,RH
ISN         5           DATA RR /750*(0.D0,0.D0)/
ISN         6           DATA RRT/750*(0.D0,0.D0)/
ISN         7           END

*STATISTICS*   SOURCE STATEMENTS = 7, PROGRAM SIZE = 0 BYTES, PROGRAM NAME = BLKDT#     PAGE:    2.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**BLKDT#** END OF COMPILATION 2 ******
```

```
LEVEL 1.4.0 (OCT 1984)          VS FORTRAN          DATE: SEP 17, 1985    TIME: 08:15:16                PAGE:    3

OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  ) LINECOUNT(64)      CHARLEN(500)

                *....*...1....,.....2.....3.....4.....5.....6.....:7.*.....8

ISN         1           FUNCTION BESSI(L,Z)
            C       CALCULATES I(Z) WITH INDEX L+1/2
ISN         2           IMPLICIT REAL*8(A-I,O-Z)
ISN         3           INTEGER*4 IND
ISN         4           DIMENSION TAB(7)
ISN         5           DATA TAB /1.D0,3.D0,15.D0,105.D0,945.D0,10395.D0,135135.D0/
ISN         6           IF(Z .LT. 1.00D0)GOTO 30
ISN         7           PREF=1.0D0/DSQRT(Z*2.0D0*.7853981633974829300)
ISN         8           I0=DSINH(Z)
ISN         9           I1=DCOSH(Z)-DSINH(Z)/Z
ISN        10           IF(L.GT.1)GOTO 10
ISN        11           BESSI=PREF*I0
ISN        12           IF(L.EQ.1)BESSI=PREF*I1
ISN        13           RETURN
ISN        14       10  CONTINUE
ISN        15           DO 20 J=2,L
ISN        16           BESSI=I0-DFLOAT(2*J-1)*I1/Z
ISN        17           I0=I1
ISN        18           I1=BESSI
ISN        19       20  CONTINUE
ISN        20           BESSI=PREF*BESSI
ISN        21           RETURN
ISN        22       30  CONTINUE
ISN        23           PREF=DSQRT(Z/(2.00*.7853981633974829300))*Z**L /TAB(L+1)
ISN        24           IND=0
ISN        25           CUR=1.D0
ISN        26           SUM=1.D0
ISN        27           Z2=0.5D0*Z*Z
ISN        28       40  CONTINUE
ISN        29           IND=IND+1
ISN        30           CUR=CUR*Z2/DFLOAT(IND*(1+2*(L+IND)))
ISN        31           SUM=SUM+CUR
ISN        32           IF(CUR.GT.1.D-17)GOTO 40
ISN        33           BESSI=PREF*SUM
ISN        34           RETURN
ISN        35           END

*STATISTICS*   SOURCE STATEMENTS = 35, PROGRAM SIZE = 1250 BYTES, PROGRAM NAME = BESSI     PAGE:   3.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**BESSI** END OF COMPILATION 3 ******
```

```
LEVEL 1.4.0 (OCT 1984)          VS FORTRAN          DATE: SEP 17, 1985    TIME: 08:15:17                    PAGE:    4

OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(64)      CHARLEN(500)

                *....*...1.........2.........3.........4.........5.........6.........7.*.......8

ISN         1           FUNCTION ERFO(A,K2)
ISN         2           IMPLICIT REAL*8(A-H,K-Z)
ISN         3           DATA FPI /12.5663706143591 7D0/
ISN         4           ERFO=FPI*DEXP(-0.25D0*K2/A)/K2
ISN         5           RETURN
ISN         6           END

*STATISTICS*   SOURCE STATEMENTS = 6, PROGRAM SIZE = 432 BYTES, PROGRAM NAME = ERFO      PAGE:    4.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**ERFO** END OF COMPILATION 4 ******
```

```
LEVEL 1.4.0 (OCT 1984)          VS FORTRAN          DATE: SEP 17, 1985    TIME: 08:15:17                    PAGE:    5

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG  ISRCFLG.
                   NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                     OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(64)      CHARLEN(500)

                  *....*...1.....2.....3.....4.....5.....6.....7.*.....8

ISN         1          FUNCTION FACT(N)
ISN         2          IMPLICIT REAL*8(A-H,O-Z)
ISN         3          DATA IT /0/
ISN         4          DIMENSION FAC(51)
ISN         5          IF(IT.NE.0)GOTO 100
ISN         6          FF=1.0D0
ISN         7          FAC(1)=1.0D0
ISN         8          IT=1
ISN         9          DO 10 J=1,50
ISN        10          FF=FF*DFLOAT(J)
ISN        11          FAC(J+1)=FF
ISN        12   10     CONTINUE
ISN        13   100    FACT=FAC(N+1)
ISN        14          RETURN
ISN        15          END

*STATISTICS*   SOURCE STATEMENTS= 15, PROGRAM SIZE = 920 BYTES, PROGRAM NAME = FACT      PAGE:    5.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**FACT** END OF COMPILATION 5 ******
```

```
LEVEL 1.4.0 (OCT 1984)          VS FORTRAN          DATE: SEP 17, 1985    TIME: 08:15:17                    PAGE:    6

OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(64)      CHARLEN(500)

                  *....*...1.........2.........3.........4.........5.........6.........7.*.....8

ISN         1           FUNCTION GAMII(N)
                C       COMPUTES GAMMA(N/2)
ISN         2           IMPLICIT REAL*8(A-H,O-Z)
ISN         3           DATA IT /0/
ISN         4           DATA FF /1.772453850905500/
ISN         5           DIMENSION FAC(51)
ISN         6           IF(IT.NE.0)GOTO 100
ISN         7           FAC(1)=FF
ISN         8           IT=1
ISN         9           DO 10 J=1,50
ISN        10           FF=FF*(DFLOAT(J)-0.500)
ISN        11           FAC(J+1)=FF
ISN        12       10  CONTINUE
ISN        13       100 IF((2*(N/2)).NE.N)GOTO 20
ISN        14           GAMII=FACT((N-2)/2)
ISN        15           RETURN
ISN        16       20  GAMII=FAC((N+1)/2)
ISN        17           RETURN
ISN        18           END

*STATISTICS*    SOURCE STATEMENTS = 18, PROGRAM SIZE = 1066 BYTES, PROGRAM NAME = GAMII      PAGE:    6.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**GAMII** END OF COMPILATION 6 ******
```

```
LEVEL 1.4.0 (OCT 1984)          VS FORTRAN          DATE: SEP 17, 1985    TIME: 08:15:17                    PAGE:    7

OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                    OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(64)      CHARLEN(500)

                   *....*...1.....2.....3.....4.....5.....6.....7.*.....8

ISN         1          FUNCTION GAUSSO(A,ALPHA,BETA,L)
ISN         2          IMPLICIT REAL*8(A-H,O-Z)
ISN         3          PREF=.78539816339744829300/(A*DSQRT(ALPHA*BETA))
ISN         4          PREF=PREF*DEXP((ALPHA*ALPHA+BETA*BETA)/(-4.D0*A))
ISN         5          GAUSSO=PREF*BESSI(L,ALPHA*BETA/(2.0D0*A))
ISN         6          RETURN
ISN         7          END

*STATISTICS*    SOURCE STATEMENTS = 7, PROGRAM SIZE = 676 BYTES, PROGRAM NAME = GAUSSO    PAGE:    7.

*STATISTICS*     NO DIAGNOSTICS GENERATED.

**GAUSSO** END OF COMPILATION 7 ******
```

```
LEVEL 1.4.0 (OCT 1984)          VS FORTRAN          DATE: SEP 17, 1985    TIME: 08:15:17                    PAGE:    8

OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE  TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(64)      CHARLEN(500)

                 *....*...1.........2.........3.........4.........5.........6.........7.*.......8

ISN         1          FUNCTION GAUSS2(A,ALPHA,BETA,L)
ISN         2          IMPLICIT REAL*8(A-H,O-Z)
ISN         3          REAL*8 NU
ISN         4          INTEGER*4 TNU
ISN         5          NU=DFLOAT(L)+0.5D0
ISN         6          TNU=2*L+1
ISN         7          ALBE=ALPHA*BETA
ISN         8          PREF=0.25D0*GAMH(TNU+4)*ALBE**L/(FACT(TNU)*DSQRT(A)*A**(L+2))
ISN         9          TEM1=0.25D0*(ALPHA**2+BETA**2)/A
ISN        10          TEM2=0.5D0*ALBE/A
ISN        11          PREF=PREF*DEXP(-TEM1)
ISN        12          GAUSS2=PREF*((1.D0-TEM1/(NU+1.D0))*TYP1(L,TEM2)+
                      $          TEM2*TYP2(L,TEM2)/(NU+1.D0))
ISN        13          RETURN
ISN        14          END

*STATISTICS*   SOURCE STATEMENTS = 14, PROGRAM SIZE = 1042 BYTES, PROGRAM NAME = GAUSS2     PAGE:    8.

*STATISTICS*   NO DIAGNOSTICS GENERATED.

**GAUSS2** END OF COMPILATION 8 ******
```

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF GOSTMT NODECK SOURCE TERM NOOBJECT FIXED NOTEST TRMFLG SRCFLG
NOSYM NORENT NOSDUMP AUTODBL(NONE) NOSXM
OPT(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN) LINECOUNT(64) CHARLEN(500)

```
                *....*...1.....2.....3.....4.....5.....6.....7.*.....8
ISN         1         SUBROUTINE GCAL
ISN         2         IMPLICIT REAL*8(A-H,O-Z)
ISN         3         COMMON/INFO/A(3),B(3),C(3),G(750,4),TAU(3),VOL,NUM,NUM1
ISN         4         COMMON/MINUS/INDEXG(533,533)
ISN         5         COMMON/STARS/MAP(533,4),MAPO(4),ISTART(75),ISTOP(75),NSTARS
                C
                C  ELEMENTS OF MAP ARE BARZ, 3, MXY, 2Z (IF BARZ, 2Z IS REDUNDANT)
                C  MAP(I,J)=K MEANS THAT UNDER J, G(I) GOES TO G(K)
                C  ** NOTE ** TIME REVERSAL IS BUILT INTO SUBROUTINE  RHO AND NEED
                C             NOT BE TAKEN INTO ACCOUNT AT THIS LEVEL  (VIA INDEXG)
                C
ISN         6         DIMENSION G1(3),G2(3),G3(3),IORD(533),TEM(533)
ISN         7         EQUIVALENCE(IORD(1),INDEXG(1,1))
ISN         8         MAPO(1)=4
ISN         9         MAPO(2)=3
ISN        10         MAPO(3)=2
ISN        11         MAPO(4)=2
ISN        12         NSTARS=1
ISN        13         ISTART(1)=1
ISN        14         TPI=2.D0*3.1415926535897900
ISN        15    6    NEST=6+DFLOAT(NUM)**(1./3.)
ISN        16         IF((NEST/2)*2.EQ.NEST)NEST=NEST+1
ISN        18         NSUB=NEST/2+1
ISN        19         VOL=A(1)*(B(2)*C(3)-B(3)*C(2))+A(2)*(B(3)*C(1)-B(1)*C(3))
                $     +A(3)*(B(1)*C(2)-B(2)*C(1))
ISN        20         VOL1=TPI/VOL
ISN        21         VOL=DABS(VOL)
ISN        22         WRITE(6,3)VOL
ISN        23         GMAX2=TPI*TPI*(1.6D0*DFLOAT(NUM)/(VOL*TPI))**(2.D0/3.D0)
ISN        24         G1(1)=(B(2)*C(3)-B(3)*C(2))*VOL1
ISN        25         G1(2)=(B(3)*C(1)-B(1)*C(3))*VOL1
ISN        26         G1(3)=(B(1)*C(2)-B(2)*C(1))*VOL1
ISN        27         G2(1)=(C(2)*A(3)-C(3)*A(2))*VOL1
ISN        28         G2(2)=(C(3)*A(1)-C(1)*A(3))*VOL1
ISN        29         G2(3)=(C(1)*A(2)-C(2)*A(1))*VOL1
ISN        30         G3(1)=(A(2)*B(3)-A(3)*B(2))*VOL1
ISN        31         G3(2)=(A(3)*B(1)-A(1)*B(3))*VOL1
ISN        32         G3(3)=(A(1)*B(2)-A(2)*B(1))*VOL1
ISN        33         IN=0
ISN        34         DO 10 I=1,NEST
ISN        35         I1=I-NSUB
ISN        36         DO 10 J=1,NEST
ISN        37         I2=J-NSUB
ISN        38         DO 10 K=1,NEST
ISN        39         I3=K-NSUB
ISN        40         X=I1*G1(1)+I2*G2(1)+I3*G3(1)
ISN        41         Y=I1*G1(2)+I2*G2(2)+I3*G3(2)
ISN        42         Z=I1*G1(3)+I2*G2(3)+I3*G3(3)
ISN        43         GGG=X*X+Y*Y+Z*Z
ISN        44         IF(GGG.GT.GMAX2)GOTO 9
ISN        45         GGG=GGG+2.71D-4*DABS(X*Y*Z)+3.14D-5*(DABS(X)+DABS(Y)+DABS(Z))
                2  -1.D-7*(2.D0*DABS(X)+DABS(Y))-1.D-12*(1.D2*X+1.D1*Y+1.D0*Z)
ISN        46         IN=IN+1
ISN        47         G(IN,1)=X
ISN        48         G(IN,2)=Y
ISN        49         G(IN,3)=Z
```

```
                *....*...1.....2.....3.....4.....5.....6.....7.*.....8
ISN       50            G(IN,4)=GGG
ISN       51    9       CONTINUE
ISN       52    10      CONTINUE
ISN       53            IF(IN .LE. 533)GOTO 5
ISN       54            NUM=NUM-5
ISN       55            GOTO 6
ISN       56    5       NUM=IN
ISN       57            DO 30 I=1,NUM
ISN       58    30      IORD(I)=I
ISN       59    33      CONTINUE
ISN       60            ITES=0
ISN       61            DO 35 I=2,NUM
ISN       62            IF(G(IORD(I),4).GE.G(IORD(I-1),4))GOTO 34
ISN       63            ITES=1
ISN       64            J=IORD(I)
ISN       65            IORD(I)=IORD(I-1)
ISN       66            IORD(I-1)=J
ISN       67    34      CONTINUE
ISN       68    35      CONTINUE
ISN       69            IF(ITES.EQ.1)GOTO 33
ISN       70            DO 40 J=1,4
ISN       71            DO 39 I=1,NUM
ISN       72    39      TEM(I)=G(IORD(I),J)
ISN       73            DO 40 I=1,NUM
ISN       74    40      G(I,J)=TEM(I)
ISN       75            WRITE(6,1)NUM
ISN       76            DO 42 I=2,NUM
ISN       77            IF((G(I,4)-G(I-1,4)).LE.3.D-5)GOTO42
ISN       78            ISTOP(NSTARS)=I-1
ISN       79            NSTARS=NSTARS+1
ISN       80            ISTART(NSTARS)=I
ISN       81    42      CONTINUE
ISN       82            ISTOP(NSTARS)=NUM
ISN       83            MAP(1,1)=1
ISN       84            MAP(1,2)=1
ISN       85            MAP(1,3)=1
ISN       86            MAP(1,4)=1
ISN       87            IN=2
ISN       88            I1=2
ISN       89            I2=ISTOP(2)
ISN       90            DO 45 I=2,NUM
ISN       91            IF(ISTOP(IN).GE.I)GOTO47
ISN       92            IN=IN+1
ISN       93            I1=ISTART(IN)
ISN       94            I2=ISTOP(IN)
ISN       95    47      CONTINUE
                  C     BAR4Z
ISN       96            TEM(1)=-G(I,2)
ISN       97            TEM(2)=G(I,1)
ISN       98            TEM(3)=-G(I,3)
ISN       99            DO 121 J=I1,I2
ISN      100            IF(DABS(TEM(1)-G(J,1)).GT.1.D-7)GOTO121
ISN      101            IF(DABS(TEM(2)-G(J,2)).GT.1.D-7)GOTO121
ISN      102            IF(DABS(TEM(3)-G(J,3)).GT.1.D-7)GOTO121
ISN      103            MAP(1,1)=J
ISN      104            GOTO122
ISN      105    121     CONTINUE
ISN      106    122     CONTINUE
                  C     3
ISN      107            TEM(1)=G(I,2)
```

```
                *....*...1.....2.....3.....4.....5.....6.....7.*.....8
ISN      108          TEM(2)=G(I,3)
ISN      109          TEM(3)=G(I,1)
ISN      110          DO 123 J=I1,I2
ISN      111          IF(DABS(TEM(1)-G(J,1)).GT.1.D-7)GOTO123
ISN      112          IF(DABS(TEM(2)-G(J,2)).GT.1.D-7)GOTO123
ISN      113          IF(DABS(TEM(3)-G(J,3)).GT.1.D-7)GOTO123
ISN      114          MAP(I,2)=J
ISN      115          GOTO124
ISN      116    123   CONTINUE
ISN      117    124   CONTINUE
                C     MXY
ISN      118          TEM(1)=G(I,2)
ISN      119          TEM(2)=G(I,1)
ISN      120          TEM(3)=G(I,3)
ISN      121          DO 125 J=I1,I2
ISN      122          IF(DABS(TEM(1)-G(J,1)).GT.1.D-7)GOTO125
ISN      123          IF(DABS(TEM(2)-G(J,2)).GT.1.D-7)GOTO125
ISN      124          IF(DABS(TEM(3)-G(J,3)).GT.1.D-7)GOTO125
ISN      125          MAP(I,3)=J
ISN      126          GOTO126
ISN      127    125   CONTINUE
ISN      128    126   CONTINUE
                C     2Z
ISN      129          TEM(1)=-G(I,1)
ISN      130          TEM(2)=-G(I,2)
ISN      131          TEM(3)= G(I,3)
ISN      132          DO 127 J=I1,I2
ISN      133          IF(DABS(TEM(1)-G(J,1)).GT.1.D-7)GOTO127
ISN      134          IF(DABS(TEM(2)-G(J,2)).GT.1.D-7)GOTO127
ISN      135          IF(DABS(TEM(3)-G(J,3)).GT.1.D-7)GOTO127
ISN      136          MAP(I,4)=J
ISN      137          GOTO128
ISN      138    127   CONTINUE
ISN      139    128   CONTINUE
ISN      140    45    CONTINUE
ISN      141          DO 41 I=1,NUM
ISN      142    41    G(I,4)=G(I,1)**2+G(I,2)**2+G(I,3)**2
ISN      143          WRITE(6,43)
ISN      144          DO 20 I=1,NUM
ISN      145    20    WRITE(6,2)I,G(I,1),G(I,2),G(I,3),G(I,4)
ISN      146          NUM1=NUM
ISN      147          DO 50 I1=1,NUM
ISN      148          DO 50 I2=1,NUM
ISN      149          G1(1)=G(I1,1)-G(I2,1)
ISN      150          G1(2)=G(I1,2)-G(I2,2)
ISN      151          G1(3)=G(I1,3)-G(I2,3)
ISN      152          GGG=G1(1)**2+G1(2)**2+G1(3)**2
ISN      153          DO 55 I=1,NUM
ISN      154          IF(DABS(G1(1)-G(I,1)).GT.1.D-5)GOTO54
ISN      155          IF(DABS(G1(2)-G(I,2)).GT.1.D-5)GOTO54
ISN      156          IF(DABS(G1(3)-G(I,3)).GT.1.D-5)GOTO54
ISN      157          INDEXG(I1,I2)=I
ISN      158          GOTO 52
ISN      159    54    CONTINUE
ISN      160    55    CONTINUE
ISN      161          INDEXG(I1,I2)=NUM+10
ISN      162    52    CONTINUE
ISN      163    50    CONTINUE
ISN      164          WRITE(6,101)
ISN      165          RETURN
```

```
                *....*...1.........2.........3.........4.........5.........6.........7.*.......8
ISN       166   1       FORMAT(/,' SYMMETRICAL NUM = ',I3)
ISN       167   2       FORMAT(' G(',I3,') = ',3F11.6,'     G*G=',F12.6)
ISN       168   3       FORMAT(' VOL=',F11.6,/)
ISN       169   43      FORMAT(' MAP FOLLOWS BAR4Z,3,MXY,2Z')
ISN       170   101     FORMAT(' SUPPLEMENTARY G VECTORS NOT USED')
ISN       171   102     FORMAT(' G(',I4,')= ',3F11.6,'     G*G=',F12.6)
ISN       172           END
```

STATISTICS SOURCE STATEMENTS = 171, PROGRAM SIZE = 10972 BYTES, PROGRAM NAME = GCAL PAGE: 9.

STATISTICS NO DIAGNOSTICS GENERATED.

GCAL END OF COMPILATION 9 ******

```
LEVEL 1.4.0 (OCT 1984)          VS FORTRAN          DATE: SEP 17, 1985    TIME: 08:15:18                    PAGE:    13

OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(64)      CHARLEN(500)

                *....*...1.........2.........3.........4.........5.........6.........7.*.......8

ISN         1           SUBROUTINE MATRIX(K,LMAX,AR,AI)
ISN         2           IMPLICIT REAL*8 (A-H,O-Z)
ISN         3           COMMON/INFO/B(9),G(750,4),TAU(3),VOL,NUM,NUM1
ISN         4           COMMON/DENS/RH(533),RR(750),RRT(750),EX
ISN         5           COMMON/MINUS/INDEXG(533,533)
ISN         6           DIMENSION K(3),G1(3),G2(3),AR(NUM,NUM),AI(NUM,NUM)
ISN         7           COMPLEX*16 R,RT,TEM,RH,RR,RRT,VCOUL,STRUCT,ADD
ISN         8           REAL*8 K
ISN         9           DO 10 I1=1,NUM
ISN        10           DO 10 I2=I1,NUM
ISN        11           IND=INDEXG(I1,I2)
ISN        12           R=RR(IND)
ISN        13           RT=RRT(IND)
ISN        14           IF(IND.EQ.1)RT=(0.D0,0.D0)
ISN        16           DO 20 I=1,3
ISN        17           G1(I)=G(I1,I)
ISN        18           G2(I)=G(I2,I)
ISN        19   20      CONTINUE
ISN        20           GS=0.D0
ISN        21           IF(IND.LE.NUM1)GS=G(IND,4)
ISN        23           ADD=STRUCT(K,G1,G2,LMAX)+VCOUL(R,GS)+2.D0*RT
ISN        24           AR(I1,I2)=DREAL(ADD)
ISN        25           AI(I1,I2)=DIMAG(ADD)
ISN        26           AR(I2,I1)=AR(I1,I2)
ISN        27           AI(I2,I1)=-AI(I1,I2)
                C       WRITE(4,99)G1(1),G1(2),G1(3),G2(1),G2(2),G2(3),ADD
                C 99    FORMAT(1X,3F8.4,2X,3F8.4,2X,2F11.5)
ISN        28   9       CONTINUE
ISN        29   10      CONTINUE
ISN        30           DO 30 I=1,NUM
ISN        31           GK=(G(I,1)+K(1))**2+(G(I,2)+K(2))**2+(G(I,3)+K(3))**2
ISN        32   30      AR(I,I)=AR(I,I)+GK
ISN        33           RETURN
ISN        34           END

*STATISTICS*   SOURCE STATEMENTS = 32, PROGRAM SIZE = 2076 BYTES, PROGRAM NAME = MATRIX     PAGE:    13.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**MATRIX** END OF COMPILATION 10 ******
```

```
LEVEL 1.4.0 (OCT 1984)          VS FORTRAN          DATE: SEP 17, 1985    TIME: 08:15:18                PAGE:    14

OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                     OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN   )  LINECOUNT(64)      CHARLEN(500)

                *....*...1.........2.........3.........4.........5.........6.........7.*.......8

ISN         1           FUNCTION PLEG(L,X)
ISN         2           IMPLICIT REAL*8(A-H,O-Z)
ISN         3           P0=1.0D0
ISN         4           P1=X
ISN         5           IF(L.GT.1)GOTO 10
ISN         6           PLEG=P0
ISN         7           IF(L.EQ.1)PLEG=P1
ISN         9           RETURN
ISN        10    10     CONTINUE
ISN        11           DO 20 I=2,L
ISN        12           PLEG=(DFLOAT(2*I-1)*X*P1-DFLOAT(I-1)*P0)/DFLOAT(I)
ISN        13           P0=P1
ISN        14           P1=PLEG
ISN        15    20     CONTINUE
ISN        16           RETURN
ISN        17           END

*STATISTICS*   SOURCE STATEMENTS = 16, PROGRAM SIZE = 682 BYTES, PROGRAM NAME = PLEG      PAGE:    14.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**PLEG** END OF COMPILATION 11 ******
```

```
OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(64)      CHARLEN(500)

                *....*...1.....2.....3.....4.....5.....6.....7.*.....8

ISN         1           SUBROUTINE READIN(IG)
ISN         2           IMPLICIT REAL*8(A-H,O-Z)
ISN         3           COMMON/INFO/A(3),B(3),C(3),G(750,4),TAU(3),VOL,NUM,NUM1
ISN         4           COMMON/CONST1/ZV1,AC1(2),CC1(2),AO1(3),CO1(6),A11(3),C11(6),
                       $          A21(3),C21(6)
ISN         5           COMMON/CONST2/ZV2,AC2(2),CC2(2),AO2(3),CO2(6),A12(3),C12(6),
                       $          A22(3),C22(6)
ISN         6    1      FORMAT(2X,I3,I5)
ISN         7    2      FORMAT(3F10.0)
ISN         8    3      FORMAT(6F10.0)
ISN         9    4      FORMAT(1X,/,3(2X,3F11.6))
ISN        10    5      FORMAT(1X,3F12.7)
ISN        11    6      FORMAT(1X,6F12.7)
ISN        12    7      FORMAT(/,/,' ATOM1',/)
ISN        13    8      FORMAT(/,/,' ATOM2',/)
ISN        14    9      FORMAT(' TAU=',3F13.8)
ISN        15   10      FORMAT(1H1)
ISN        16   11      FORMAT('  NUM=',I3,/)
ISN        17           WRITE(6,10)
ISN        18           READ(5,1)NUM,IG
ISN        19           WRITE(6,11)NUM
ISN        20           READ(5,2)(A(I),I=1,3)
ISN        21           READ(5,2)(B(I),I=1,3)
ISN        22           READ(5,2)(C(I),I=1,3)
ISN        23           READ(5,2)(TAU(I),I=1,3)
ISN        24           WRITE(6,4)A(1),A(2),A(3),B(1),B(2),B(3),C(1),C(2),C(3)
ISN        25           WRITE(6,9)(TAU(I),I=1,3)
ISN        26           WRITE(6,7)
ISN        27           READ(5,2)ZV1,AC1(1),AC1(2)
ISN        28           READ(5,2)CC1(1),CC1(2)
ISN        29           READ(5,2)(AO1(I),I=1,3)
ISN        30           READ(5,3)(CO1(I),I=1,6)
ISN        31           READ(5,2)(A11(I),I=1,3)
ISN        32           READ(5,3)(C11(I),I=1,6)
ISN        33           READ(5,2)(A21(I),I=1,3)
ISN        34           READ(5,3)(C21(I),I=1,6)
ISN        35           WRITE(6,5)ZV1,AC1(1),AC1(2)
ISN        36           WRITE(6,5)CC1(1),CC1(2)
ISN        37           WRITE(6,5)(AO1(I),I=1,3)
ISN        38           WRITE(6,6)(CO1(I),I=1,6)
ISN        39           WRITE(6,5)(A11(I),I=1,3)
ISN        40           WRITE(6,6)(C11(I),I=1,6)
ISN        41           WRITE(6,5)(A21(I),I=1,3)
ISN        42           WRITE(6,6)(C21(I),I=1,6)
ISN        43           WRITE(6,8)
ISN        44           READ(5,2)ZV2,AC2(1),AC2(2)
ISN        45           READ(5,2)CC2(1),CC2(2)
ISN        46           READ(5,2)(AO2(I),I=1,3)
ISN        47           READ(5,3)(CO2(I),I=1,6)
ISN        48           READ(5,2)(A12(I),I=1,3)
ISN        49           READ(5,3)(C12(I),I=1,6)
ISN        50           READ(5,2)(A22(I),I=1,3)
ISN        51           READ(5,3)(C22(I),I=1,6)
ISN        52           WRITE(6,5)ZV2,AC2(1),AC2(2)
ISN        53           WRITE(6,5)CC2(1),CC2(2)
ISN        54           WRITE(6,5)(AO2(I),I=1,3)
```

```
                *....*...1.....2.....3.....4.....5.....6.....7.*.....8

ISN       55          WRITE(6,6)(C02(I),I=1,6)
ISN       56          WRITE(6,5)(A12(I),I=1,3)
ISN       57          WRITE(6,6)(C12(I),I=1,6)
ISN       58          WRITE(6,5)(A22(I),I=1,3)
ISN       59          WRITE(6,6)(C22(I),I=1,6)
ISN       60          CALL TRANS(A01(1),A01(2),A01(3),
                     $           C01(1),C01(2),C01(3),C01(4),C01(5),C01(6))
ISN       61          CALL TRANS(A11(1),A11(2),A11(3),
                     $           C11(1),C11(2),C11(3),C11(4),C11(5),C11(6))
ISN       62          CALL TRANS(A21(1),A21(2),A21(3),
                     $           C21(1),C21(2),C21(3),C21(4),C21(5),C21(6))
ISN       63          CALL TRANS(A02(1),A02(2),A02(3),
                     $           C02(1),C02(2),C02(3),C02(4),C02(5),C02(6))
ISN       64          CALL TRANS(A12(1),A12(2),A12(3),
                     $           C12(1),C12(2),C12(3),C12(4),C12(5),C12(6))
ISN       65          CALL TRANS(A22(1),A22(2),A22(3),
                     $           C22(1),C22(2),C22(3),C22(4),C22(5),C22(6))
ISN       66          RETURN
ISN       67          END
```

STATISTICS SOURCE STATEMENTS = 67, PROGRAM SIZE = 4368 BYTES, PROGRAM NAME = READIN PAGE: 15.

STATISTICS NO DIAGNOSTICS GENERATED.

READIN END OF COMPILATION 12 ******

```
OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN   )  LINECOUNT(64)      CHARLEN(500)

                *....*...1.........2.........3.........4.........5.........6.........7.*.......8

ISN       1           SUBROUTINE READK(K,WEIGHT,NUMK)
ISN       2           IMPLICIT REAL*8(A-H,O-Z)
ISN       3           REAL*8 K
ISN       4           DIMENSION K(100,3),WEIGHT(100)
ISN       5     1     FORMAT(2X,I3)
ISN       6     2     FORMAT(4F10.0)
ISN       7     3     FORMAT('1 NUMK=',I3,/)
ISN       8     4     FORMAT(' K,WEIGHT  ',4F11.6)
ISN       9           READ(5,1)NUMK
ISN      10           WRITE(6,3)NUMK
ISN      11           DO 10 I=1,NUMK
ISN      12           READ(5,2)K(I,1),K(I,2),K(I,3),WEIGHT(I)
ISN      13     10    WRITE(6,4)K(I,1),K(I,2),K(I,3),WEIGHT(I)
ISN      14           RETURN
ISN      15           END

*STATISTICS*   SOURCE STATEMENTS = 15, PROGRAM SIZE = 982 BYTES, PROGRAM NAME = READK      PAGE:   17.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**READK** END OF COMPILATION 13 ******
```

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF GOSTMT NODECK SOURCE TERM NOOBJECT FIXED NOTEST TRMFLG SRCFLG
NOSYM NORENT NOSDUMP AUTODBL(NONE) NOSXM
OPT(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN) LINECOUNT(64) CHARLEN(500)

```
                *....*...1.....2.....3.....4.....5.....6.....7.*.....8
ISN         1         SUBROUTINE RHO
ISN         2         IMPLICIT REAL*8(A-H,O-Z)
ISN         3         COMPLEX*16 RRT,RR,RH,VCOUL
ISN         4         COMPLEX*16 TEM1,TEM2,TEM3,RCON,RCON1(48)
ISN         5         LOGICAL LGDOSM,LGF,LGDONE
ISN         6         COMMON/INFO/A(3),B(3),C(3),G(750,4),TAU(3),VOL,NUM,NUM1
ISN         7         COMMON/DENS/RH(533),RR(750),RRT(750),EX
ISN         8         COMMON/MINUS/INDEXG(533,533)
ISN         9         COMMON/STARS/MAP(533,4),MAPO(4),ISTART(75),ISTOP(75),NSTARS
ISN        10         COMMON/SPACE/RCON(750),WORK(750,8)
ISN        11         DIMENSION DELA(3),DELB(3),DELC(3),LGDOSM(4),LGF(48)
ISN        12         DIMENSION IEQV(48)
ISN        13         DATA DAMP/0.1500D0/
                C  ELEMENTS OF MAP ARE BAR4Z, 3, MXY, 2Z (IF BAR4Z, 2Z IS REDUNDANT)
ISN        14         DATA LGDOSM/.FALSE.,.TRUE.,.TRUE.,.FALSE./
ISN        15         DAMM=(1.D0-DAMP)/DFLOAT(12167)
ISN        16         DO 60 I=1,NUM1
ISN        17   60    RCON(I)=DCMPLX(0.D0,0.D0)
ISN        18         DO 70 IND1=1,NUM
ISN        19         IND2=INDEXG(1,IND1)
ISN        20         RCON(IND1)=RCON(IND1)+0.5D0*RH(IND1)
ISN        21         RCON(IND2)=RCON(IND2)+0.5D0*DCONJG(RH(IND1))
ISN        22   70    CONTINUE
ISN        23         RCON(1)=RCON(1)/VOL
ISN        24         DO 80 NST=2,NSTARS
ISN        25         ISHIFT=ISTART(NST)-1
ISN        26         NS=ISTOP(NST)+1-ISTART(NST)
ISN        27         DO 79 I1=1,NS
ISN        28         DO 78 I2=1,NS
ISN        29   78    LGF(I2)=.FALSE.
ISN        30         LGF(I1)=.TRUE.
ISN        31         NEQ=1
ISN        32         RCON1(I1)=RCON(I1+ISHIFT)
ISN        33         IEQV(1)=I1+ISHIFT
ISN        34   119   LGDONE=.TRUE.
ISN        35         IF(.NOT.LGDOSM(1))GOTO81
ISN        36         NEQ1=NEQ
ISN        37         DO 83 J=1,NEQ1
ISN        38         JT=MAP(IEQV(J),1)
ISN        39         IF(LGF(JT-ISHIFT))GOTO83
ISN        40         LGF(JT-ISHIFT)=.TRUE.
ISN        41         LGDONE=.FALSE.
ISN        42         NEQ=NEQ+1
ISN        43         IEQV(NEQ)=JT
ISN        44         RCON1(I1)=RCON1(I1)+RCON(JT)
ISN        45   83    CONTINUE
ISN        46   81    CONTINUE
ISN        47         IF(.NOT.LGDOSM(2))GOTO91
ISN        48         NEQ1=NEQ
ISN        49         DO 93 J=1,NEQ1
ISN        50         JT=MAP(IEQV(J),2)
ISN        51         IF(LGF(JT-ISHIFT))GOTO93
ISN        52         LGF(JT-ISHIFT)=.TRUE.
ISN        53         LGDONE=.FALSE.
ISN        54         NEQ=NEQ+1
ISN        55         IEQV(NEQ)=JT
```

```
                *....*...1.....2.....3.....4.....5.....6.....7.*.....8
ISN       56          RCON1(I1)=RCON1(I1)+RCON(JT)
ISN       57    93    CONTINUE
ISN       58    91    CONTINUE
ISN       59          IF(.NOT.LGDOSM(3))GOTO71
ISN       60          NEQ1=NEQ
ISN       61          DO 73 J=1,NEQ1
ISN       62          JT=MAP(IEQV(J),3)
ISN       63          IF(LGF(JT-ISHIFT))GOTO73
ISN       64          LGF(JT-ISHIFT)=.TRUE.
ISN       65          LGDONE=.FALSE.
ISN       66          NEQ=NEQ+1
ISN       67          IEQV(NEQ)=JT
ISN       68          RCON1(I1)=RCON1(I1)+RCON(JT)
ISN       69    73    CONTINUE
ISN       70    71    CONTINUE
ISN       71          IF(.NOT.LGDOSM(4))GOTO61
ISN       72          NEQ1=NEQ
ISN       73          DO 63 J=1,NEQ1
ISN       74          JT=MAP(IEQV(J),4)
ISN       75          IF(LGF(JT-ISHIFT))GOTO63
ISN       76          LGF(JT-ISHIFT)=.TRUE.
ISN       77          LGDONE=.FALSE.
ISN       78          NEQ=NEQ+1
ISN       79          IEQV(NEQ)=JT
ISN       80          RCON1(I1)=RCON1(I1)+RCON(JT)
ISN       81    63    CONTINUE
ISN       82    61    CONTINUE
ISN       83          IF(.NOT.LGDONE)GOTO119
ISN       84          RCON1(I1)=RCON1(I1)/(DFLOAT(NEQ)*VOL)
ISN       85    79    CONTINUE
ISN       86          DO 80 J=1,NS
ISN       87          RCON(J+ISHIFT)=RCON1(J)
ISN       88    80    CONTINUE
ISN       89          DO 5 I=1,3
ISN       90          DELA(I)=A(I)/23.D0
ISN       91          DELB(I)=B(I)/23.D0
ISN       92          DELC(I)=C(I)/23.D0
ISN       93    5     CONTINUE
ISN       94          DO 40 I=1,NUM
ISN       95          RRT(I)=DAMP*RRT(I)
ISN       96          RR(I)=DAMP*RR(I)+(1.D0-DAMP)*RCON(I)
ISN       97    40    CONTINUE
ISN       98          EX=0.D0
ISN       99          DO 120 I1=1,23
ISN      100          DO 120 I2=1,23
ISN      101          DO 120 I3=1,23
ISN      102          X=DFLOAT(I1)*DELA(1)+DFLOAT(I2)*DELB(1)+DFLOAT(I3)*DELC(1)
ISN      103          Y=DFLOAT(I1)*DELA(2)+DFLOAT(I2)*DELB(2)+DFLOAT(I3)*DELC(2)
ISN      104          Z=DFLOAT(I1)*DELA(3)+DFLOAT(I2)*DELB(3)+DFLOAT(I3)*DELC(3)
ISN      105          TEM2=DCMPLX(0.D0,0.D0)
ISN      106          DO 130 IG1=1,NUM
ISN      107          RG=X*G(IG1,1)+Y*G(IG1,2)+Z*G(IG1,3)
ISN      108          TEM2=TEM2+RCON(IG1)*DCMPLX(DCOS(RG),DSIN(RG))
ISN      109   130    CONTINUE
ISN      110          RHI=TABS(DREAL(TEM2))
ISN      111          TT=TFORM(RHI)
ISN      112          EX=EX+RHI*EXC(RHI)
ISN      113          RHT=DAMM*TT
ISN      114          DO 120 I=1,NUM
ISN      115          RG=G(I,1)*X+G(I,2)*Y+G(I,3)*Z
```

```
                *....*...1.....2.....3.....4.....5.....6.....7.*.....8
ISN      116          RRT(I)=RRT(I)+DCMPLX(RHT*DCOS(RG),-RHT*DSIN(RG))
ISN      117   120    CONTINUE
ISN      118          EX=EX*VOL/DFLOAT(12167)
ISN      119          REWIND 1
ISN      120          LGDONE=.FALSE.
ISN      121          READ(1,2)TEM1,TEM2
ISN      122          DO 125 I=2,NUM
ISN      123          IF(LGDONE)GOTO10
ISN      124          READ(1,2,END=10)TEM1,TEM2
ISN      125          GOTO11
ISN      126   10     TEM1=(0.D0,0.D0)
ISN      127          TEM2=(0.D0,0.D0)
ISN      128          LGDONE=.TRUE.
ISN      129   11     CONTINUE
ISN      130          TEM3=(RR(I)-DAMP*TEM1)/(1.D0-DAMP)
ISN      131          EX=EX+VOL*DREAL(DCONJG(TEM3)*((0.25D0,0.D0)*
                     1  VCOUL(TEM3,G(I,4))-(0.5D0,0.D0)*VCOUL(TEM1,G(I,4))-TEM2))
ISN      132   125    CONTINUE
ISN      133          REWIND 1
ISN      134          WRITE(6,4)EX
ISN      135          DO 50 I=1,NUM
ISN      136          WRITE(6,1)RCON(I),RR(I),RRT(I)
ISN      137          WRITE(1,2)RR(I),RRT(I)
ISN      138   50     CONTINUE
ISN      139          REWIND 1
ISN      140          RETURN
ISN      141   4      FORMAT(' EX ',F13.7)
ISN      142   2      FORMAT(4F20.15)
ISN      143   1      FORMAT(' RCON,RR,RRT ',3(3X,2F12.7))
ISN      144          END

*STATISTICS*  SOURCE STATEMENTS = 144, PROGRAM SIZE = 7818 BYTES, PROGRAM NAME = RHO  PAGE: 18.

*STATISTICS*  NO DIAGNOSTICS GENERATED.

**RHO** END OF COMPILATION 14 ******
```

```
OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  ) LINECOUNT(64)      CHARLEN(500)

                *....*...1.....2.....3.....4.....5.....6.....7.*.....8

ISN         1           SUBROUTINE RSETUP(ISET,WEIGHT,VECR,VECI,EIG)
ISN         2           IMPLICIT REAL*8(A-H,O-Z)
ISN         3           COMMON/INFO/G(3009),TAU(4),NUM,NUM1
ISN         4           COMMON/DENS/RH(533),RR(750),RRT(750),EX
ISN         5           COMMON/MINUS/INDEXG(533,533)
ISN         6           COMMON/TRACE/TR
ISN         7           DIMENSION VECR(NUM,5),VECI(NUM,5),EIG(5)
ISN         8           COMPLEX*16 RH,TEM,RR,RRT
ISN         9           IF(ISET.NE.0)TR=0.D0
ISN        11           TR=TR+2.D0*WEIGHT*(EIG(1)+EIG(2)+EIG(3)+EIG(4))
ISN        12           DO 5 I1=1,4
ISN        13           TEST=0.D0
ISN        14           DO 6 I2=1,NUM
ISN        15     6     TEST=TEST+VECR(I2,I1)*VECR(I2,I1)+VECI(I2,I1)*VECI(I2,I1)
ISN        16           TEST=DSQRT(TEST)
ISN        17           DO 5 I2=1,NUM
ISN        18           VECR(I2,I1)=VECR(I2,I1)/TEST
ISN        19     5     VECI(I2,I1)=VECI(I2,I1)/TEST
ISN        20           IF(ISET.EQ.0)GOTO8
ISN        21           DO 7 I1=1,NUM
ISN        22     7     RH(I1)=(0.D0,0.D0)
ISN        23     8     CONTINUE
ISN        24           DO 10 I1=1,NUM
ISN        25           DO 10 I2=1,NUM
ISN        26           IND=INDEXG(I1,I2)
ISN        27           IF(IND.GT.NUM)GOTO10
ISN        28           TEM=(0.D0,0.D0)
ISN        29           DO 20 I=1,4
ISN        30           TEM=TEM+
                       1 DCMPLX(VECR(I2,I),-VECI(I2,I))*DCMPLX(VECR(I1,I),VECI(I1,I))
ISN        31    20     CONTINUE
ISN        32           RH(IND)=RH(IND)+TEM*DCMPLX(2.D0*WEIGHT,0.D0)
ISN        33    10     CONTINUE
ISN        34           RETURN
ISN        35           END

*STATISTICS*   SOURCE STATEMENTS = 34, PROGRAM SIZE = 2218 BYTES, PROGRAM NAME = RSETUP    PAGE:   21.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**RSETUP** END OF COMPILATION 15 *****
```

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF GOSTMT NODECK SOURCE TERM NOOBJECT FIXED NOTEST TRMFLG SRCFLG
NOSYM NORENT NOSDUMP AUTODBL(NONE) NOSXM
OPT(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN) LINECOUNT(64) CHARLEN(500)

```
                *....*...1.........2.........3.........4.........5.........6.........7.*.......8
ISN         1           SUBROUTINE SCF(AR,AI,VECR,VECI,EIG)
ISN         2           IMPLICIT REAL*8(A-H,O-Z)
ISN         3           REAL*8 K(100,3)
ISN         4           COMMON/INFO/G(3012),VOL,NUM,NUM1
ISN         5           COMMON/DENS/HH(533),RR(750),RRT(750),EX
ISN         6           COMMON/TRACE/TR
ISN         7           COMMON/SPACE/WORK(7500)
ISN         8           COMPLEX*16 RH,RR,RRT
ISN         9           DIMENSION WEIGHT(100),AR(1,1),AI(1,1),VECR(1,1),VECI(1,1)
ISN        10           DIMENSION EIG(1),TK(3)
ISN        11           DATA TK /3*5.D-3/
ISN        12           DATA INUM / 1/
ISN        13           DATA ETES/0.D0/
ISN        14    1      FORMAT(/,'  TRACE=',F11.6,5X,'ENERGY=',F12.6,/)
ISN        15    2      FORMAT(1H1)
ISN        16    4      FORMAT('1 CONVERGED NOW DO K=0 TO GET GAP',/)
ISN        17           CALL READIN(IG)
ISN        18           CALL READK(K,WEIGHT,NUMK)
ISN        19           CALL GCAL
ISN        20           IF(IG.EQ.0)CALL GUESS
ISN        22           IF(IG .NE. 0)CALL LAST
ISN        24   10      CALL ITER(K,WEIGHT,NUMK,2,AR,AI,VECR,VECI,EIG)
ISN        25           INUM=INUM-1
ISN        26           CALL ENERGY(EN)
ISN        27           WRITE(6,1)TR,EN
ISN        28           WRITE(4,1)TR,EN
ISN        29           IF(DABS(TR-ETES) .LT. 1.D-4)GOTO 20
ISN        30           IF(INUM .LE. 0)GOTO 21
ISN        31           REWIND 1
ISN        32           ETES=TR
ISN        33           WRITE(6,2)
ISN        34           GOTO 10
ISN        35   20      WRITE(6,4)
ISN        36   21      CONTINUE
                C       CALL MATRIX(TK,2,AR,AI)
                C       RLB=0.30
                C       RUB=1.18D0
                C       CALL EISCH3(NUM,NUM,AR,AI,RLB,RUB,4,NFOUND,
                C      1        EIG,VECR,VECI,ICOND,WORK)
                C       NU=NUM
                C       IF(NU.GT.16)NU=16
                C       WRITE(6,3)(EIG(I),I=1,NU)
                C     3 FORMAT('1  FOR K=(0,0,0) EIG=',8F11.6,/,21X,8F11.6)
                C    24 READ(5,22,END=23)TK(1),TK(2),TK(3)
                C       CALL BAND(TK,2,AR,AI,VECR,VECI,EIG)
                C       GOTO 24
                C    23 CONTINUE
ISN        37           RETURN
ISN        38   22      FORMAT(3F10.0)
ISN        39           END
```

STATISTICS SOURCE STATEMENTS = 37, PROGRAM SIZE = 4380 BYTES, PROGRAM NAME = SCF PAGE: 22.

STATISTICS NO DIAGNOSTICS GENERATED.

SCF END OF COMPILATION 16 ******

```
OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77),NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(64)      CHARLEN(500)

                *....*...1.....2.....3.....4.....5.....6.....7.*.....8

ISN         1           SUBROUTINE LAST
ISN         2           IMPLICIT REAL*8(A-H,O-Z)
ISN         3           COMPLEX*16 RRT,RR,RH
ISN         4           COMMON/INFO/G(3013),NUM,NUM1
ISN         5           COMMON/DENS/RH(533),RR(750),RRT(750),EX
ISN         6    1      FORMAT(4F20.15)
ISN         7    2      FORMAT(' USE RHO FROM LAST ITERATION')
ISN         8    3      FORMAT(' R,RT',2F11.6,3X,2F11.6,7X,I4)
ISN         9           WRITE(6,2)
ISN        10           DO 10 I=1,NUM1
ISN        11           READ(3,1,END=99)RR(I),RRT(I)
ISN        12           WRITE(1,1)RR(I),RRT(I)
ISN        13   10      WRITE(6,3)RR(I),RRT(I),I
ISN        14   99      REWIND 1
ISN        15           RETURN
ISN        16           END

*STATISTICS*   SOURCE STATEMENTS = 16, PROGRAM SIZE = 886 BYTES, PROGRAM NAME = LAST      PAGE:   23.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**LAST** END OF COMPILATION 17 ******
```

```
OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK-  SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NOREENT NOSDUMP AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(64)      CHARLEN(500)

                *....*...1.....2.....3.....4.....5.....6.....7.*.....8

ISN         1       FUNCTION STRUCT(K,G,GP,LMAX)
ISN         2       IMPLICIT REAL*8(A-H,O-Z)
ISN         3       REAL*8 K
ISN         4       COMMON/INFO/D(3009),TAU(3),VOL,NUM,NUM1
ISN         5       COMPLEX*16 STRUCT
ISN         6       DIMENSION K(3),G(3),GP(3),GG(3)
ISN         7       GG(1)=G(1)-GP(1)
ISN         8       GG(2)=G(2)-GP(2)
ISN         9       GG(3)=G(3)-GP(3)
                C
                C
                C     NEXT TWO STATEMENTS FIND V(G) FOR ATOMS 1 AND 2
                C     IF THE SAME S2=S1
                C     ELSE EVALUATE BOTH   USEING SUML1 AND SUML2
                C
                C
ISN        10       S1=SUML1(K,G,GP,LMAX)
                C     S2=S1
ISN        11       S2=SUML2(K,G,GP,LMAX)
ISN        12       QT=2.D0*(GG(1)*TAU(1)+GG(2)*TAU(2)+GG(3)*TAU(3))
ISN        13       SS=2.D0*(S1+DCOS(QT)*S2)/VOL
ISN        14       SA=2.D0*DSIN(QT)*S2/VOL
ISN        15       STRUCT=DCMPLX(SS,SA)
ISN        16       RETURN
ISN        17       END

*STATISTICS*   SOURCE STATEMENTS = 17, PROGRAM SIZE = 894 BYTES, PROGRAM NAME = STRUCT    PAGE:   24.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**STRUCT** END OF COMPILATION 18 ******
```

```
OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                     OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(64)      CHARLEN(500)

                *....*...1.........2.........3.........4.........5.........6.....:...7.*.......8

ISN         1           FUNCTION ERF1(A,K2)
ISN         2           IMPLICIT REAL*8(A-H,K-Z)
ISN         3           DATA PI /3.1415926535897900/
ISN         4           ERF1=(PI/A)**1.500*DEXP(-0.2500*K2/A)
ISN         5           RETURN
ISN         6           END

*STATISTICS*   SOURCE STATEMENTS = 6, PROGRAM SIZE = 494 BYTES, PROGRAM NAME = ERF1      PAGE:   25.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**ERF1** END OF COMPILATION 19 ******
```

```
OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF  GOSTMT NODECK  SOURCE  TERM NOOBJECT FIXED  NOTEST  TRMFLG  SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)  NOSXM
                      OPT(0) LANGLVL(77) NOFIPS  FLAG(I)  NAME(MAIN  )  LINECOUNT(64)  CHARLEN(500)

                *....*...1.....2.....3.....4.....5.....6.....7.*.....8

ISN         1          FUNCTION ERF2(A,K2)
ISN         2          IMPLICIT REAL*8(A-H,K-Z)
ISN         3          DATA PI /3.1415926535897900/
ISN         4          ERF2=(PI/A)**1.5D0*(6.D0*A-K2)*DEXP(-0.25D0*K2/A)/(4.D0*A*A)
ISN         5          RETURN
ISN         6          END

*STATISTICS*   SOURCE STATEMENTS = 6, PROGRAM SIZE = 558 BYTES, PROGRAM NAME = ERF2      PAGE:  26.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**ERF2** END OF COMPILATION 20 ******
```

```
OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(64)      CHARLEN(500)

                *....*...1.....2.....3.....4.....5.....6.....7.*.....8

ISN         1           FUNCTION SUML1(K,G,GP,LMAX)
ISN         2           IMPLICIT REAL*8(A-H,O-Z)
ISN         3           REAL*8 K
ISN         4           COMMON/CONST1/ZV,A1C,A2C,C1C,C2C,A10,A20,A30,C10,C20,C30,C40,
                   $      C50,C60,A11,A21,A31,C11,C21,C31,C41,C51,C61,A12,A22,A32,
                   $           C12,C22,C32,C42,C52,C62
ISN         5           DIMENSION K(3),G(3),GP(3)
ISN         6           DATA FPI /12.566370614359917D0/
ISN         7           ALPHA=DSQRT((K(1)+G(1))**2+(K(2)+G(2))**2+(K(3)+G(3))**2)
ISN         8           BETA=DSQRT((K(1)+GP(1))**2+(K(2)+GP(2))**2+(K(3)+GP(3))**2)
ISN         9           ALBE=ALPHA*BETA
ISN        10           COSGAM=1.D0
ISN        11           IF(ALBE.GT.1.D-6)COSGAM=((K(1)+G(1))*(K(1)+GP(1))+
                   $  (K(2)+G(2))*(K(2)+GP(2))+(K(3)+G(3))*(K(3)+GP(3)))/ALBE
ISN        13           L=0
ISN        14           TEMP=C10*GAUSS0(A10,ALPHA,BETA,L)+C40*GAUSS2(A10,ALPHA,BETA,L)
                   $     +C20*GAUSS0(A20,ALPHA,BETA,L)+C50*GAUSS2(A20,ALPHA,BETA,L)
                   $     +C30*GAUSS0(A30,ALPHA,BETA,L)+C60*GAUSS2(A30,ALPHA,BETA,L)
                   $     -C12*GAUSS0(A12,ALPHA,BETA,L)-C42*GAUSS2(A12,ALPHA,BETA,L)
                   $     -C22*GAUSS0(A22,ALPHA,BETA,L)-C52*GAUSS2(A22,ALPHA,BETA,L)
                   $     -C32*GAUSS0(A32,ALPHA,BETA,L)-C62*GAUSS2(A32,ALPHA,BETA,L)
ISN        15           SUML1=FPI*DFLOAT(2*L+1)*TEMP*PLEG(L,COSGAM)
ISN        16           L=1
ISN        17           TEMP=C11*GAUSS0(A11,ALPHA,BETA,L)+C41*GAUSS2(A11,ALPHA,BETA,L)
                   $     +C21*GAUSS0(A21,ALPHA,BETA,L)+C51*GAUSS2(A21,ALPHA,BETA,L)
                   $     +C31*GAUSS0(A31,ALPHA,BETA,L)+C61*GAUSS2(A31,ALPHA,BETA,L)
                   $     -C12*GAUSS0(A12,ALPHA,BETA,L)-C42*GAUSS2(A12,ALPHA,BETA,L)
                   $     -C22*GAUSS0(A22,ALPHA,BETA,L)-C52*GAUSS2(A22,ALPHA,BETA,L)
                   $     -C32*GAUSS0(A32,ALPHA,BETA,L)-C62*GAUSS2(A32,ALPHA,BETA,L)
ISN        18           SUML1=SUML1+FPI*DFLOAT(2*L+1)*TEMP*PLEG(L,COSGAM)
ISN        19     10    CONTINUE
ISN        20           TEMP=(G(1)-GP(1))**2+(G(2)-GP(2))**2+(G(3)-GP(3))**2
ISN        21           SUML1=SUML1+C12*ERF1(A12,TEMP)+C22*ERF1(A22,TEMP)+
                   $ C32*ERF1(A32,TEMP)+C42*ERF2(A12,TEMP)+C52*ERF2(A22,TEMP)
                   $ +C62*ERF2(A32,TEMP)
ISN        22           IF(TEMP .LT. 1.D-5)RETURN
ISN        24           SUML1=SUML1-ZV*(C1C*ERF0(A1C,TEMP)+C2C*ERF0(A2C,TEMP))
ISN        25           RETURN
ISN        26           END

*STATISTICS*   SOURCE STATEMENTS = 24, PROGRAM SIZE = 2678 BYTES, PROGRAM NAME = SUML1      PAGE:   27.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**SUML1** END OF COMPILATION 21 ******
```

```
OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(60)      CHARLEN(500)

                 *....*...1.....2.....3.....4.....5.....6.....7.*.....8

ISN         1          FUNCTION SUML2(K,G,GP,LMAX)
ISN         2          IMPLICIT REAL*8(A-H,O-Z)
ISN         3          REAL*8 K
ISN         4          COMMON/CONST2/ZV,A1C,A2C,C1C,C2C,A10,A20,A30,C10,C20,C30,C40,
                      $     C50,C60,A11,A21,A31,C11,C21,C31,C41,C51,C61,A12,A22,A32,
                      $           C12,C22,C32,C42,C52,C62
ISN         5          DIMENSION K(3),G(3),GP(3)
ISN         6          DATA FPI /12.566370614359170D0/
ISN         7          ALPHA=DSQRT((K(1)+G(1))**2+(K(2)+G(2))**2+(K(3)+G(3))**2)
ISN         8          BETA=DSQRT((K(1)+GP(1))**2+(K(2)+GP(2))**2+(K(3)+GP(3))**2)
ISN         9          ALBE=ALPHA*BETA
ISN        10          COSGAM=1.D0
ISN        11          IF(ALBE.GT.1.D-6)COSGAM=((K(1)+G(1))*(K(1)+GP(1))+
                      $  (K(2)+G(2))*(K(2)+GP(2))+(K(3)+G(3))*(K(3)+GP(3)))/ALBE
ISN        13          L=0
ISN        14          TEMP=C10*GAUSS0(A10,ALPHA,BETA,L)+C40*GAUSS2(A10,ALPHA,BETA,L)
                      $     +C20*GAUSS0(A20,ALPHA,BETA,L)+C50*GAUSS2(A20,ALPHA,BETA,L)
                      $     +C30*GAUSS0(A30,ALPHA,BETA,L)+C60*GAUSS2(A30,ALPHA,BETA,L)
                      $     -C12*GAUSS0(A12,ALPHA,BETA,L)-C42*GAUSS2(A12,ALPHA,BETA,L)
                      $     -C22*GAUSS0(A22,ALPHA,BETA,L)-C52*GAUSS2(A22,ALPHA,BETA,L)
                      $     -C32*GAUSS0(A32,ALPHA,BETA,L)-C62*GAUSS2(A32,ALPHA,BETA,L)
ISN        15          SUML2=FPI*DFLOAT(2*L+1)*TEMP*PLEG(L,COSGAM)
ISN        16          L=1
ISN        17          TEMP=C11*GAUSS0(A11,ALPHA,BETA,L)+C41*GAUSS2(A11,ALPHA,BETA,L)
                      $     +C21*GAUSS0(A21,ALPHA,BETA,L)+C51*GAUSS2(A21,ALPHA,BETA,L)
                      $     +C31*GAUSS0(A31,ALPHA,BETA,L)+C61*GAUSS2(A31,ALPHA,BETA,L)
                      $     -C12*GAUSS0(A12,ALPHA,BETA,L)-C42*GAUSS2(A12,ALPHA,BETA,L)
                      $     -C22*GAUSS0(A22,ALPHA,BETA,L)-C52*GAUSS2(A22,ALPHA,BETA,L)
                      $     -C32*GAUSS0(A32,ALPHA,BETA,L)-C62*GAUSS2(A32,ALPHA,BETA,L)
ISN        18          SUML2=SUML2+FPI*DFLOAT(2*L+1)*TEMP*PLEG(L,COSGAM)
ISN        19    10    CONTINUE
ISN        20          TEMP=(G(1)-GP(1))**2+(G(2)-GP(2))**2+(G(3)-GP(3))**2
ISN        21          SUML2=SUML2+C12*ERF1(A12,TEMP)+C22*ERF1(A22,TEMP)+
                      $ C32*ERF1(A32,TEMP)+C42*ERF2(A12,TEMP)+C52*ERF2(A22,TEMP)
                      $ +C62*ERF2(A32,TEMP)
ISN        22          IF(TEMP .LT. 1.D-5)RETURN
ISN        24          SUML2=SUML2-ZV*(C1C*ERF0(A1C,TEMP)+C2C*ERF0(A2C,TEMP))
ISN        25          RETURN
ISN        26          END

*STATISTICS*   SOURCE STATEMENTS = 24, PROGRAM SIZE = 2678 BYTES, PROGRAM NAME = SUML2      PAGE:   28.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**SUML2** END OF COMPILATION 22 ******
```

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF GOSTMT NODECK SOURCE TERM NOOBJECT FIXED NOTEST TRMFLG SRCFLG
NOSYM NORENT NOSDUMP AUTODBL(NONE) NOSXM
OPT(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN) LINECOUNT(64) CHARLEN(500)

```
                *....*...1.....2.....3.....4.....5.....6.....7.*.....8
ISN         1           FUNCTION SUM1(M,A)
ISN         2           IMPLICIT REAL*8(A-H,O-Z)
ISN         3           IF(DABS(A) .LT. 1.00D0)GOTO 20
ISN         4           F1=1.0D0
ISN         5           F2=1.0D0
ISN         6           S1=1.0D0
ISN         7           S2=1.0D0
ISN         8           SIGN=1.D0
ISN         9           IF(2*(M/2).EQ.M)SIGN=-1.D0
ISN        11           IF(M.EQ.0)GOTO 5
ISN        12           DO 10 I=1,M
ISN        13           F1=DFLOAT(M+1-I)*F1/A
ISN        14           F2=-F2
ISN        15           S1=S1+F1
ISN        16           S2=S2+F1*F2
ISN        17    10     CONTINUE
ISN        18    5      CONTINUE
ISN        19           SUM1=(S2*DEXP(A)+SIGN*S1*DEXP(-A))/A
ISN        20           RETURN
ISN        21    20     CONTINUE
ISN        22           IF(2*(M/2) .NE. M)GOTO 30
ISN        23           INDM=0
ISN        24           A2=A*A
ISN        25           CUR=2.D0
ISN        26           SUM1=CUR/DFLOAT(M+1)
ISN        27    21     CONTINUE
ISN        28           INDM=INDM+2
ISN        29           CUR=CUR*A2/DFLOAT(INDM*(INDM-1))
ISN        30           SUM1=SUM1+CUR/DFLOAT(INDM+M+1)
ISN        31           IF(CUR.GT.1.D-17)GOTO 21
ISN        32           RETURN
ISN        33    30     CONTINUE
ISN        34           INDM=1
ISN        35           A2=A*A
ISN        36           CUR=2.D0*A
ISN        37           SUM1=CUR/DFLOAT(M+2)
ISN        38    22     CONTINUE
ISN        39           INDM=INDM+2
ISN        40           CUR=CUR*A2/DFLOAT(INDM*(INDM-1))
ISN        41           SUM1=SUM1+CUR/DFLOAT(INDM+M+1)
ISN        42           IF(DABS(CUR).GT.1.D-17)GOTO 22
ISN        43           RETURN
ISN        44           END
```

STATISTICS SOURCE STATEMENTS = 43, PROGRAM SIZE = 1434 BYTES, PROGRAM NAME = SUM1 PAGE: 29.

STATISTICS NO DIAGNOSTICS GENERATED.

SUM1 END OF COMPILATION 23 ******

```
LEVEL 1.4.0 (OCT 1984)          VS FORTRAN          DATE: SEP 17, 1985    TIME: 08:15:20                    PAGE:    30

OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN   )  LINECOUNT(64)      CHARLEN(500)

                *....*...1.........2.........3.........4.........5.........6.........7.*.......8

ISN         1           FUNCTION TABS(A)
ISN         2           IMPLICIT REAL*8(A-H,O-Z)
ISN         3           TABS=A
ISN         4           IF(A.GT.0.D0)RETURN
ISN         6           TABS=0.D0
ISN         7           WRITE(6,1)A
ISN         8           RETURN
ISN         9     1     FORMAT('  A LESS THAN ZERO, A =',F13.5,5X,'*************')
ISN        10           END

*STATISTICS*   SOURCE STATEMENTS = 9, PROGRAM SIZE = 496 BYTES, PROGRAM NAME = TABS      PAGE:  30.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**TABS** END OF COMPILATION 24 ******
```

```
OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(64)      CHARLEN(500)

                *....*...1.....2.....3.....4.....5.....6.....7.*.....8

ISN         1           FUNCTION TFORM(RR)
ISN         2           IMPLICIT REAL*8(A-H,O-Z)
ISN         3           REAL*8 LN
ISN         4           DATA CON1 /0.2387324100/
                      C     CON1=3/(4*PI)
ISN         5           DATA OT  /0.333333333333333300/
ISN         6           IF(RR.EQ.0.00)GOTO20
ISN         7           R=(CON1/RR)**OT
ISN         8           TFORM=-0.61093/R
ISN         9           IF(R .LT. 1.00)GOTO 10
ISN        10           SR=DSQRT(R)
ISN        11           TFORM=TFORM-(0.143200+0.175900*SR+0.063660*R)/
                      $         (1.00+1.052900*SR+0.333400*R)**2
ISN        12           RETURN
ISN        13     10    LN=DLOG(R)
ISN        14           TFORM=TFORM-0.058370+0.031100*LN-0.008400*R+0.001300*LN*R
ISN        15           RETURN
ISN        16     20    TFORM=0.00
ISN        17           RETURN
ISN        18           END

*STATISTICS*   SOURCE STATEMENTS = 18, PROGRAM SIZE = 834 BYTES, PROGRAM NAME = TFORM     PAGE:   31.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**TFORM** END OF COMPILATION 25 ******
```

```
LEVEL 1.4.0 (OCT 1984)          VS FORTRAN          DATE: SEP 17, 1985    TIME: 08:15:20                PAGE:    32

OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP.AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(64)      CHARLEN(500)

                  *....*...1.........2.........3.........4.........5.........6.........7.*.......8

ISN         1           FUNCTION EXC(RR)
ISN         2           IMPLICIT REAL*8(A-H,O-Z)
ISN         3           DATA CON1 /0.2387324100/
ISN         4           DATA OT /0.3333333333333333D0/
ISN         5           IF(RR.EQ.0.D0)GOTO20
ISN         6           R=(CON1/RR)**OT
ISN         7           EXC=-0.4582D0/R
ISN         8           IF(R.LT.1.D0)GOTO10
ISN         9           EXC=EXC-0.1432D0/(1.D0+1.0529D0*DSQRT(R)+0.3334D0*R)
ISN        10           RETURN
ISN        11   10      RL=DLOG(R)
ISN        12           EXC=EXC-0.0480D0+0.0311D0*RL-0.0116D0*R+0.0020D0*R*RL
ISN        13           RETURN
ISN        14   20      EXC=0.0D0
ISN        15           RETURN
ISN        16           END

*STATISTICS*   SOURCE STATEMENTS = 16, PROGRAM SIZE = 756 BYTES, PROGRAM NAME = EXC       PAGE:    32.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**EXC** END OF COMPILATION 26 ******
```

```
OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF  GOSTMT NODECK  SOURCE  TERM NOOBJECT FIXED  NOTEST  TRMFLG  SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)  NOSXM
                      OPT(0) LANGLVL(77) NOFIPS  FLAG(I)  NAME(MAIN  )  LINECOUNT(64)  CHARLEN(500)

                *....*...1.....2.....3.....4.....5.....6.....7.*.....8

ISN         1           FUNCTION TYP1(L,A)
ISN         2           IMPLICIT REAL*8(A-H,O-Z)
ISN         3           SIGN=-1.0D0
ISN         4           IF(2*(L/2).NE.L)SIGN= 1.0D0
ISN         6           FACTL=FACT(L)
ISN         7           SUM=0.0D0
ISN         8           L1=L+1
ISN         9           DO 10 K1=1,L1
ISN        10           K=L1-K1
ISN        11           SIGN=-SIGN
ISN        12           COEFF=FACTL/(FACT(K)*FACT(L-K))
ISN        13           SUM=SUM+SIGN*COEFF*SUM1(2*K,A)
ISN        14    10     CONTINUE
ISN        15           TYP1=SUM
ISN        16           RETURN
ISN        17           END

*STATISTICS*   SOURCE STATEMENTS = 16, PROGRAM SIZE = 804 BYTES, PROGRAM NAME = TYP1      PAGE:   33.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**TYP1** END OF COMPILATION 27 ******
```

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF GOSTMT NODECK SOURCE TERM NOOBJECT FIXED NOTEST TRMFLG SRCFLG
NOSYM NORENT NOSDUMP AUTODBL(NONE) NOSXM
OPT(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN) LINECOUNT(64) CHARLEN(500)

```
                *....*...1.........2.........3.........4.........5.........6.........7.*.......8

ISN         1           FUNCTION TYP2(L,A)
ISN         2           IMPLICIT REAL*8(A-H,O-Z)
ISN         3           SIGN=-1.0D0
ISN         4           IF(2*(L/2).NE.L)SIGN= 1.0D0
ISN         6           FACTL=FACT(L)
ISN         7           SUM=0.0D0
ISN         8           L1=L+1
ISN         9           DO 10 K1=1,L1
ISN        10           K=L1-K1
ISN        11           SIGN=-SIGN
ISN        12           COEFF=FACTL/(FACT(K)*FACT(L-K))
ISN        13           SUM=SUM+SIGN*COEFF*SUM1(2*K+1,A)
ISN        14     10    CONTINUE
ISN        15           TYP2=SUM
ISN        16           RETURN
ISN        17           END
```

STATISTICS SOURCE STATEMENTS = 16, PROGRAM SIZE = 808 BYTES, PROGRAM NAME = TYP2 PAGE: 34.

STATISTICS NO DIAGNOSTICS GENERATED.

TYP2 END OF COMPILATION 28 ******

```
LEVEL 1.4.0 (OCT 1984)          VS FORTRAN          DATE: SEP 17, 1985    TIME: 08:15:20                    PAGE:    35

OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(60)      CHARLEN(500)

                 *....*...1.........2.........3.........4.........5.........6.........7.*.......8

ISN         1           FUNCTION VCOUL(R,GS)
ISN         2           IMPLICIT REAL*8(A-H,O-Z)
ISN         3           COMPLEX*16 VCOUL,R
ISN         4           DATA EPI /25.132741200/
ISN         5           IF(GS.GT.1.D-9)GOTO 10
ISN         6           VCOUL=DCMPLX(0.D0,0.D0)
ISN         7           RETURN
ISN         8     10    VCOUL=R*DCMPLX(EPI/GS,0.D0)
ISN         9           RETURN
ISN        10           END

*STATISTICS*   SOURCE STATEMENTS = 10, PROGRAM SIZE = 516 BYTES, PROGRAM NAME = VCOUL      PAGE:    35:

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**VCOUL** END OF COMPILATION 29 ******
```

```
OPTIONS IN EFFECT:  NOLIST NOMAP,NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST . TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(64)      CHARLEN(500)

                *....*...1.........2.........3.........4.........5.........6.........7.*.......8

ISN         1           SUBROUTINE ITER(K,WEIGHT,NUMK,LMAX,AR,AI,VECR,VECI,EIG)
ISN         2           IMPLICIT REAL*8(A-H,O-Z)
ISN         3           REAL*8 K
ISN         4           COMMON/INFO/G(3013),NUM,NUM1
ISN         5           COMMON/SPACE/WORK(7500)
ISN         6           DIMENSION K(100,3),WEIGHT(100),AR(NUM,NUM),AI(NUM,NUM)
ISN         7           DIMENSION EIG(1),VECR(NUM,5),VECI(NUM,5),TK(3)
ISN         8           REAL*8 RLB(20)/0.40D0,0.48D0,0.40D0,0.48D0,0.48D0,15*0.D0/
ISN         9           REAL*8 RUB(20)/1.15D0,1.06D0,1.15D0,1.06D0,1.06D0,15*0.D0/
ISN        10    1      FORMAT(/,' SOLVE FOR K =',3F11.6)
ISN        11    2      FORMAT('      SET UP MATRIX')
ISN        12    3      FORMAT('      DIAGONALIZE')
ISN        13    4      FORMAT(' EIG =',9F11.6)
ISN        14    5      FORMAT(' EIG =',5F11.6,/,6X,4F11.6)
ISN        15           ISET=1
ISN        16           DO 10 I=1,NUMK
ISN        17           TK(1)=K(I,1)
ISN        18           TK(2)=K(I,2)
ISN        19           TK(3)=K(I,3)
ISN        20           WRITE(6,1)TK(1),TK(2),TK(3)
ISN        21           WRITE(6,2)
ISN        22           CALL MATRIX(TK,LMAX,AR,AI)
ISN        23           WRITE(6,3)
                   C
                   C     CALL EISPACK DIAGONALIZATION
                   C
ISN        24           CALL EISCH3(NUM,NUM,AR,AI,RLB(I),RUB(I),4,NFOUND,
                   1          EIG,VECR,VECI,ICOND,WORK)
ISN        25           WRITE(6,4)(EIG(J),J=1,4)
ISN        26           WRITE(4,5)(EIG(J),J=1,4)
ISN        27           IF(ICOND.NE.0)STOP'ERROR IN EISCH3'
ISN        29           IF(NFOUND.LT.4)STOP'NOT ENOUGH EIGENVALUES FOUND'
ISN        31           CALL RSETUP(ISET,WEIGHT(I),VECR,VECI,EIG)
ISN        32           ISET=0
ISN        33    10     CONTINUE
ISN        34           CALL RHO
ISN        35           RETURN
ISN        36           END

*STATISTICS*   SOURCE STATEMENTS = 34, PROGRAM SIZE = 2344 BYTES, PROGRAM NAME = ITER      PAGE:   36.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**ITER** END OF COMPILATION 30 ******
```

```
OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   OSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(64)      CHARLEN(500)

                *....*...1.........2.........3.........4.........5.........6.........7.*.......8

ISN         1           SUBROUTINE GUESS
ISN         2           IMPLICIT REAL*8(A-H,O-Z)
ISN         3           COMPLEX*16 RRT,RR,RH
ISN         4           COMPLEX*16 TEM2
ISN         5           COMMON/INFO/A(3),B(3),C(3),G(750,4),TAU(4),NUM,NUM1
ISN         6           COMMON/DENS/RH(533),RR(750),RRT(750),EX
ISN         7           DIMENSION DELA(3),DELB(3),DELC(3)
ISN         8           DO 5 I=1,3
ISN         9           DELA(I)=A(I)/13.DO
ISN        10           DELB(I)=B(I)/13.DO
ISN        11           DELC(I)=C(I)/13.DO
ISN        12    5      CONTINUE
ISN        13           DO 40 I=1,NUM
ISN        14           RR(I)=DCMPLX(0.DO,0.DO)
ISN        15    40     RRT(I)=DCMPLX(0.DO,0.DO)
ISN        16           DO 120 I1=1,13
ISN        17           DO 120 I2=1,13
ISN        18           DO 120 I3=1,13
ISN        19           X=DFLOAT(I1)*DELA(1)+DFLOAT(I2)*DELB(1)+DFLOAT(I3)*DELC(1)
ISN        20           Y=DFLOAT(I1)*DELA(2)+DFLOAT(I2)*DELB(2)+DFLOAT(I3)*DELC(2)
ISN        21           Z=DFLOAT(I1)*DELA(3)+DFLOAT(I2)*DELB(3)+DFLOAT(I3)*DELC(3)
ISN        22           RHI=RINIT(X,Y,Z)
ISN        23           RHT=TFORM(DABS(RHI))
ISN        24           DO 120 I=1,NUM
ISN        25           RG=G(I,1)*X+G(I,2)*Y+G(I,3)*Z
ISN        26           RRT(I)=RRT(I)+DCMPLX(RHT*DCOS(RG),-RHT*DSIN(RG))
ISN        27           RR(I)=RR(I)+DCMPLX(RHI*DCOS(RG),-RHI*DSIN(RG))
ISN        28   120     CONTINUE
ISN        29           TEM2=DCMPLX(1.DO/DFLOAT(2197),0.DO)
ISN        30           DO 50 I=1,NUM
ISN        31           RR(I)=RR(I)*TEM2
ISN        32           RRT(I)=RRT(I)*TEM2
ISN        33           WRITE(6,1)RR(I),RRT(I)
ISN        34           WRITE(1,2)RR(I),RRT(I)
ISN        35    50     CONTINUE
ISN        36           REWIND 1
ISN        37           RETURN
ISN        38    1      FORMAT(' RR,RRT ',2(3X,2F12.7))
ISN        39    2      FORMAT(4F20.15)
ISN        40           END

*STATISTICS*   SOURCE STATEMENTS = 40, PROGRAM SIZE = 2520 BYTES, PROGRAM NAME = GUESS     PAGE:   37.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**GUESS** END OF COMPILATION 31 ******
```

```
LEVEL 1.4.0 (OCT 1984)          VS FORTRAN          DATE: SEP 17, 1985    TIME: 08:15:23                    PAGE:    38

OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(64)      CHARLEN(500)

                 *....*...1.........2.........3.........4.........5.........6.........7.*.......8

ISN         1            FUNCTION RINIT(X,Y,Z)
ISN         2            IMPLICIT REAL*8(A-H,O-Z)
ISN         3            COMMON/INFO/A(3),B(3),C(3),G(750,4),TAU(3),VOL,NUM,NUM1
ISN         4            COMMON/CONST1/ZV1,AC1(31)
ISN         5            COMMON/CONST2/ZV2,AC2(31)
ISN         6            RINIT=8.D0/VOL
ISN         7            RETURN
ISN         8            END

*STATISTICS*  SOURCE STATEMENTS = 8, PROGRAM SIZE = 384 BYTES, PROGRAM NAME = RINIT      PAGE:    38.

*STATISTICS*     NO DIAGNOSTICS GENERATED.

**RINIT** END OF COMPILATION 32 ******
```

```
OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(64)      CHARLEN(500)

                *....*...1.....2.....3.....4.....5.....6.....7.*.....8

ISN         1           SUBROUTINE BAND(KM,LMAX,AR,AI,VECR,VECI,EIG)
ISN         2           IMPLICIT REAL*8(A-H,O-Z)
ISN         3           REAL*8 KM
ISN         4           COMMON/INFO/G(3013),NUM,NUM1
ISN         5           COMMON/SPACE/WORK(7500)
ISN         6           DIMENSION KM(3),AR(NUM,NUM),AI(NUM,NUM)
ISN         7           DIMENSION VECR(NUM,5),VECI(NUM,5),EIG(1)
ISN         8    1      FORMAT(/,' SOLVE FOR K =',3F11.6)
ISN         9    4      FORMAT(' EIG =',9F11.6)
ISN        10    5      FORMAT(' EIG =',5F13.6,/,6X,4F13.6)
ISN        11           WRITE(6,1)KM(1),KM(2),KM(3)
ISN        12           CALL MATRIX(KM,LMAX,AR,AI)
ISN        13           RLB=0.30
ISN        14           RUB=1.1800
ISN        15           CALL EISCH3(NUM,NUM,AR,AI,RLB,RUB,4,NFOUND,
                       1        EIG,VECR,VECI,ICOND,WORK)
ISN        16           IF(ICOND.NE.0)STOP'ERROR IN EISCH3'
ISN        18           IF(NFOUND.LT.4)STOP'NOT ENOUGH EIGENVALUES FOUND'
ISN        20           WRITE(6,4)(EIG(J),J=1,4)
ISN        21           WRITE(4,5)(EIG(J),J=1,4)
ISN        22           RETURN
ISN        23           END

*STATISTICS*   SOURCE STATEMENTS = 21, PROGRAM SIZE = 1620 BYTES, PROGRAM NAME = BAND      PAGE:   39.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**BAND** END OF COMPILATION 33 ******
```

```
LEVEL 1.4.0 (OCT 1984)          VS FORTRAN          DATE: SEP 17, 1985    TIME: 08:15:23          PAGE:    40

OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF  GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(64)      CHARLEN(500)

                   *....*...1.....2.....3.....4.....5.....6.....7.*.....8

ISN         1          SUBROUTINE ENERGY(EN)
ISN         2          IMPLICIT REAL*8(A-H,O-Z)
ISN         3          COMMON/TRACE/TR
ISN         4          COMMON/DENS/RH(533),RR(750),RRT(750),EX
ISN         5          COMMON/INFO/B(9),G(750,4),TAU(3),VOL,NUM,NUM1
ISN         6          COMPLEX*16 RH,RR,RRT,VCOUL
ISN         7    1     FORMAT(''ALPHA,GEWALD',2F12.6)
ISN         8          CALL ALPHAZ(AZ)
ISN         9          CALL GEWALD(GE)
ISN        10          WRITE(6,1)AZ,GE
ISN        11          EN=TR+2.D0*EX+AZ+GE
ISN        12          RETURN
ISN        13          END

*STATISTICS*   SOURCE STATEMENTS = 13, PROGRAM SIZE = 552 BYTES, PROGRAM NAME = ENERGY     PAGE:    40.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**ENERGY** END OF COMPILATION 34 ******
```

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF GOSTMT NODECK SOURCE TERM NOOBJECT FIXED NOTEST TRMFLG SRCFLG
NOSYM NORENT NOSDUMP AUTODBL(NONE) NOSXM
OPT(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN) LINECOUNT(64) CHARLEN(500)

```
                *....*...1.........2.........3.........4.........5.........6.........7.*.......8

ISN         1           SUBROUTINE GEWALD(GAMMA)
ISN         2           IMPLICIT REAL*8(A-H,O-Z)
ISN         3           COMMON/INFO/A(3),B(3),C(3),G(750,4),TAU(3),V,NUM,NUM1
ISN         4           COMMON/CONST1/Z1,D1(31)
ISN         5           COMMON/CONST2/Z2,D2(31)
ISN         6           DATA PI /3.14159265358979D0/
ISN         7           T1=2.D0*TAU(1)
ISN         8           T2=2.D0*TAU(2)
ISN         9           T3=2.D0*TAU(3)
ISN        10           CHI2=2.3*(T1*T1+T2*T2+T3*T3)
ISN        11           CHI=DSQRT(CHI2)
ISN        12           V1=-2.D0/(CHI*DSQRT(PI))
ISN        13           V2=0.D0
ISN        14           V3=PI*CHI2
ISN        15           DO 10 I=2,NUM
ISN        16           S=DCOS(T1*G(I,1)+T2*G(I,2)+T3*G(I,3))
ISN        17           GG=0.25D0*CHI2*G(I,4)
ISN        18           IF(GG.GT.35.D0)GOTO10
ISN        19           TEM=PI*CHI2*DEXP(-GG)/(GG*V)
ISN        20           V1=V1+TEM
ISN        21           V2=V2+TEM*S
ISN        22       10  CONTINUE
ISN        23           DO 20 I11=1,9
ISN        24           I1=I11-5
ISN        25           DO 20 I12=1,9
ISN        26           I2=I12-5
ISN        27           DO 20 I13=1,9
ISN        28           I3=I13-5
ISN        29           X=DFLOAT(I1)*A(1)+DFLOAT(I2)*B(1)+DFLOAT(I3)*C(1)
ISN        30           Y=DFLOAT(I1)*A(2)+DFLOAT(I2)*B(2)+DFLOAT(I3)*C(2)
ISN        31           Z=DFLOAT(I1)*A(3)+DFLOAT(I2)*B(3)+DFLOAT(I3)*C(3)
ISN        32           R12=X*X+Y*Y+Z*Z
ISN        33           R22=(X+T1)**2+(Y+T2)**2+(Z+T3)**2
ISN        34           R1=DSQRT(R12)
ISN        35           R2=DSQRT(R22)
ISN        36           IF(R1.GT.1.D-5)V1=V1+DERFC(R1/CHI)/R1
ISN        38           V2=V2+DERFC(R2/CHI)/R2
ISN        39       20  CONTINUE
ISN        40           GAMMA=(Z1*Z1+Z2*Z2)*V1+2.D0*Z1*Z2*V2-(Z1+Z2)**2*V3/V
ISN        41           RETURN
ISN        42           END
```

STATISTICS SOURCE STATEMENTS = 41, PROGRAM SIZE = 2106 BYTES, PROGRAM NAME = GEWALD PAGE: 41.

STATISTICS NO DIAGNOSTICS GENERATED.

GEWALD END OF COMPILATION 35 ******

```
LEVEL 1.4.0 (OCT 1984)          VS FORTRAN          DATE: SEP 17, 1985    TIME: 08:15:23                    PAGE:   42

OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST · TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(64)      CHARLEN(500)

                *....*...1.....2.....3.....4.....5.....6.....7.*.....8

ISN         1           SUBROUTINE ALPHAZ(ZALPHA)
ISN         2           IMPLICIT REAL*8(A-H,O-Z)
ISN         3           COMMON/INFO/G(3012),V,NUM,NUM1
ISN         4           COMMON/CONST1/Z1,A1(2),C1(2),D1(27)
ISN         5           COMMON/CONST2/Z2,A2(2),C2(2),D2(27)
ISN         6           ZALPHA=(8.D0*0.785398816D0/V)*(Z1*(C1(1)/A1(1)+C1(2)/A1(2))+
                   $        Z2*(C2(1)/A2(1)+C2(2)/A2(2)))*(Z1+Z2)
ISN         7           RETURN
ISN         8           END

*STATISTICS*   SOURCE STATEMENTS = 8, PROGRAM SIZE = 512 BYTES, PROGRAM NAME = ALPHAZ    PAGE:   42.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**ALPHAZ** END OF COMPILATION 36 ******
```

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF GOSTMT NODECK SOURCE TERM NOOBJECT FIXED NOTEST TRMFLG SRCFLG
NOSYM NORENT NOSDUMP AUTODBL(NONE) NOSXM
OPT(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN) LINECOUNT(64) CHARLEN(500)

```
                *....*...1.....2.....3.....4.....5.....6.....7.*.....8
ISN       1           SUBROUTINE TRANS(A1,A2,A3,C1,C2,C3,C4,C5,C6)
ISN       2           IMPLICIT REAL*8(A-H,O-Z)
ISN       3           REAL*16 S,Q,Y,TEMP,B2,B4,B6
ISN       4           DIMENSION S(6,6),Q(6,6),Y(6,6)
ISN       5           B2=0.5Q0*QSQRT(QATAN(1.Q0))
ISN       6           B4=1.5Q0*B2
ISN       7           B6=2.5Q0*B4
                C
                C     SET UP S
                C
ISN       8           S(1,1)=B2/((A1+A1)**1.5Q0)
ISN       9           S(1,2)=B2/((A1+A2)**1.5Q0)
ISN      10           S(1,3)=B2/((A1+A3)**1.5Q0)
ISN      11           S(2,2)=B2/((A2+A2)**1.5Q0)
ISN      12           S(2,3)=B2/((A2+A3)**1.5Q0)
ISN      13           S(3,3)=B2/((A3+A3)**1.5Q0)
ISN      14           S(1,4)=B4/((A1+A1)**2.5Q0)
ISN      15           S(1,5)=B4/((A1+A2)**2.5Q0)
ISN      16           S(1,6)=B4/((A1+A3)**2.5Q0)
ISN      17           S(2,5)=B4/((A2+A2)**2.5Q0)
ISN      18           S(2,6)=B4/((A2+A3)**2.5Q0)
ISN      19           S(3,6)=B4/((A3+A3)**2.5Q0)
ISN      20           S(2,4)=S(1,5)
ISN      21           S(3,4)=S(1,6)
ISN      22           S(3,5)=S(2,6)
ISN      23           S(4,4)=B6/((A1+A1)**3.5Q0)
ISN      24           S(4,5)=B6/((A1+A2)**3.5Q0)
ISN      25           S(4,6)=B6/((A1+A3)**3.5Q0)
ISN      26           S(5,5)=B6/((A2+A2)**3.5Q0)
ISN      27           S(5,6)=B6/((A2+A3)**3.5Q0)
ISN      28           S(6,6)=B6/((A3+A3)**3.5Q0)
                C
                C     NOW SET UP Q
                C
ISN      29           Q(1,1)=QSQRT(S(1,1))
ISN      30           DO 10 L=2,6
ISN      31           L1=L-1
ISN      32           DO 30 I=1,L1
ISN      33           TEMP=0.Q0
ISN      34           IF(I.EQ.1)GOTO 39
ISN      35           I1=I-1
ISN      36           DO 40 K=1,I1
ISN      37      40   TEMP=TEMP-Q(K,I)*Q(K,L)
ISN      38      39   Q(I,L)=(TEMP+S(I,L))/Q(I,I)
ISN      39      30   CONTINUE
ISN      40           TEMP=0.Q0
ISN      41           DO 20 K=1,L1
ISN      42      20   TEMP=TEMP-Q(K,L)*Q(K,L)
ISN      43           Q(L,L)=QSQRT(TEMP+S(L,L))
ISN      44      10   CONTINUE
                C
                C     NOW FIND -Q INVERSE AND PUT IN Y
                C
ISN      45           Y(1,1)=-1.Q0/Q(1,1)
ISN      46           DO 50 I=2,6
ISN      47           Y(I,I)=-1.Q0/Q(I,I)
```

```
                *....*...1.....2.....3.....4....5.....6.....7.*.....8
ISN       48            I1=I-1
ISN       49            DO 60 JJ=1,I1
ISN       50            J=I-JJ
ISN       51            TEMP=0.QO
ISN       52            J1=J+1
ISN       53            DO 70 K=J1,I
ISN       54      70    TEMP=TEMP-Q(J,K)*Y(K,I)
ISN       55      60    Y(J,I)=TEMP/Q(J,J)
ISN       56      50    CONTINUE
                C
                C     NOW FIND A'S AND STORE IN SAME PLACE AS C'S
                C
ISN       57            C1=C1*Y(1,1)+C2*Y(1,2)+C3*Y(1,3)+C4*Y(1,4)+C5*Y(1,5)+C6*Y(1,6)
ISN       58            C2=C2*Y(2,2)+C3*Y(2,3)+C4*Y(2,4)+C5*Y(2,5)+C6*Y(2,6)
ISN       59            C3=C3*Y(3,3)+C4*Y(3,4)+C5*Y(3,5)+C6*Y(3,6)
ISN       60            C4=C4*Y(4,4)+C5*Y(4,5)+C6*Y(4,6)
ISN       61            C5=C5*Y(5,5)+C6*Y(5,6)
ISN       62            C6=C6*Y(6,6)
ISN       63            RETURN
ISN       64            END
```

STATISTICS SOURCE STATEMENTS = 64, PROGRAM SIZE = 6626 BYTES, PROGRAM NAME = TRANS PAGE: 43.

STATISTICS NO DIAGNOSTICS GENERATED.

TRANS END OF COMPILATION 37 ******

Sample Input Data for the Total Energy Program

```
580  10          INAS
  5.50000    5.50000    0.00000
  5.50000    0.00000    5.50000              INAS INPUT DATA AT A HIGHLY
  0.00000    5.50000    5.50000              COMPRESSED VOLUME
  1.37500    1.37500    1.37500              LATTICE PARAMETER = 11 A.U.
  3.00000    1.79000    0.71000  IN
  6.72510   -5.72510
  1.09000    1.66000    2.96000
 -6.35770   -0.39020    0.26860   0.30240    0.00960   -0.02180
  0.99000    1.24000    1.53000
 -5.11500   -0.07270   -0.02210   0.15520    0.07910    0.01390
  0.64000    0.72000    0.91000
 -5.29750   -1.15210   -0.34800   0.04970   -0.04930    0.04480
  5.00000    2.60000    1.03000  AS
  2.62180   -1.62180
  2.41000    2.77000    3.52000
 -4.71620    0.79520    0.11460   0.13260   -0.01520   -0.02690
  1.74000    1.92000    2.42000
 -3.71410    0.18770    0.09870   0.08300   -0.01710   -0.01060
  1.67000    1.93000    2.22000
 -3.38450    0.09480   -0.00200   -0.04780  -0.07890   -0.02270
    2
  .142800    .142800    .142800 .25000000    TWO SPECIAL POINTS FOR FCC
  .428399    .142800    .142800 .75000000
```

The Computer Code for the SCH Approximation

```
LEVEL 1.4.0 (OCT 1984)          VS FORTRAN          DATE: SEP 17, 1985    TIME: 08:16:04                PAGE:    1

REQUESTED OPTIONS (EXECUTE): NOMAP NOXREF NOOBJECT

OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(64)      CHARLEN(500)

                *....*...1.........2.........3.........4.........5.........6.........7.*.......8

                C
                C     SELF CONSISTENT HARMONIC APPROXIMATION FOR FCC RARE GAS SOLIDS
                C
                C     PROGRAM WRITTEN BY D. SINGH (1984)
                C
                C
                C
ISN         1         IMPLICIT REAL*8(A-H,O-Z)
ISN         2         DATA WPREV/0.D0/
ISN         3         DATA PREC /5.D-6/
ISN         4         DATA ITERM/75/
ISN         5         CALL READIN
ISN         6         ITER=0
ISN         7   10    CALL DYNAM1(WAV,FS)
ISN         8         IF(DABS(WAV-WPREV).LE.PREC)GOTO20
ISN         9         WPREV=WAV
ISN        10         ITER=ITER+1
                C     WRITE(4,2)ITER,WAV,FS
ISN        11         IF(ITER.LE.ITERM)GOTO10
ISN        12         WRITE(6,1)ITER
ISN        13         STOP
ISN        14   20    CONTINUE
ISN        15         CALL BRANCH
ISN        16         FREE=FS+0.5D0*AVEO(0)
ISN        17         WRITE(6,3)FREE
ISN        18         STOP
ISN        19   1     FORMAT(/,/,' NOT CONVERGED AFTER',I3,' ITERATIONS',/,
                      1 ' ABANDONING CALCULATION **')
ISN        20   2     FORMAT(2X,I4,3X,F15.8,D17.8)
ISN        21   3     FORMAT('1 FREE ENERGY FOLLOWS',/,/,' FREE =',F13.2)
ISN        22         END

*STATISTICS*   SOURCE STATEMENTS = 22, PROGRAM SIZE = 806 BYTES, PROGRAM NAME = MAIN      PAGE:    1.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**MAIN** END OF COMPILATION 1 ******
```

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF GOSTMT NODECK SOURCE TERM NOOBJECT FIXED NOTEST TRMFLG SRCFLG
NOSYM NORENT NOSDUMP AUTODBL(NONE) NOSXM
OPT(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN) LINECOUNT(64) CHARLEN(500)

```
                *....*...1.........2.........3.........4.........5.........6.........7.*.......8

ISN         1           SUBROUTINE CALCS
ISN         2           IMPLICIT REAL*8(A-H,O-Z)
ISN         3           COMMON/ATOMS/W(2400,4),ISH(2400),RSHELL(80),DD(80,7),IS,N
ISN         4           COMMON/WIDTH/A(80,3),RM(80,3)
ISN         5           COMMON/WAVEFN/D(80,3,3),S(80,3,3),RL(80,3)
ISN         6           DIMENSION ST(3,3),X(3),DTEM(3,3),WORK(10)
ISN         7           DO 1000 ISHELL=2,IS
ISN         8           DO 6 I=1,3
ISN         9           DO 6 J=1,3
ISN        10     6     DTEM(I,J)=D(ISHELL,I,J)
                       C
                       C  CALL IMSL DIAGONALIZATION ROUTINE
                       C
ISN        11           CALL EIGRS(DTEM,3,12,X,ST,3,WORK,IER)
ISN        12           IF(DABS(X(1)).LT.1.D-20)X(1)=1.D-20
ISN        14           IF(DABS(X(2)).LT.1.D-20)X(2)=1.D-20
ISN        16           IF(DABS(X(3)).LT.1.D-20)X(3)=1.D-20
ISN        18           A(ISHELL,1)=1.D0/X(1)
ISN        19           A(ISHELL,2)=1.D0/X(2)
ISN        20           A(ISHELL,3)=1.D0/X(3)
ISN        21           DO 10 I=1,3
ISN        22           RM(ISHELL,I)=0.D0
ISN        23           DO 10 J=1,3
ISN        24           RM(ISHELL,I)=RM(ISHELL,I)+ST(J,I)*RL(ISHELL,J)
ISN        25     10    S(ISHELL,I,J)=ST(J,I)
ISN        26     1000  CONTINUE
ISN        27           RETURN
ISN        28           END
```

STATISTICS SOURCE STATEMENTS = 25, PROGRAM SIZE = 1556 BYTES, PROGRAM NAME = CALCS PAGE: 2.

STATISTICS NO DIAGNOSTICS GENERATED.

CALCS END OF COMPILATION 2 ******

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF GOSTMT NODECK SOURCE TERM NOOBJECT FIXED NOTEST TRMFLG SRCFLG
NOSYM NORENT NOSDUMP AUTODBL(NONE) NOSXM
OPT(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN) LINECOUNT(64) CHARLEN(500)

```
                *....*...1.........2.........3.........4.........5.........6.........7.*.......8
ISN        1            SUBROUTINE FORCE1(IGRID)
                C
                C       CALCULATES THE SEVEN AVERAGES NEEDED FOR THE FORCE CONSTANTS
                C
ISN        2            IMPLICIT REAL*8(A-H,O-Z)
ISN        3            COMMON/SPHERE/R(20000,3),IH(20000),N1,N2,N3
ISN        4            COMMON/ATOMS/W(2400,4),ISH(2400),RSHELL(80),DD(80,7),IS,N
ISN        5            COMMON/WIDTH/A(80,3),RM(80,3)
ISN        6            REAL*4 R
ISN        7            DATA ITEST/0/
ISN        8            DATA EPS/2.D-4/
ISN        9            DATA PI/3.14159265358979323D0/
ISN       10            IF(ITEST.EQ.0)CALL GENS(IGRID)
ISN       12            ITEST=1
ISN       13            WRITE(6,2)N1,N2,N3,EPS
ISN       14            DO 1000 ISHELL=2,IS
ISN       15            X=RM(ISHELL,1)
ISN       16            Y=RM(ISHELL,2)
ISN       17            Z=RM(ISHELL,3)
ISN       18            A1=A(ISHELL,1)
ISN       19            A2=A(ISHELL,2)
ISN       20            A3=A(ISHELL,3)
ISN       21            RM1=DSQRT(-2.D0*DLOG(EPS)/A1)
ISN       22            RM2=DSQRT(-2,D0*DLOG(EPS)/A2)
ISN       23            RM3=DSQRT(-2.D0*DLOG(EPS)/A3)
ISN       24            S11=0.D0
ISN       25            S12=0.D0
ISN       26            S13=0.D0
ISN       27            S14=0.D0
ISN       28            S15=0.D0
ISN       29            S16=0.D0
ISN       30            S17=0.D0
ISN       31            S21=0.D0
ISN       32            S22=0.D0
ISN       33            S23=0.D0
ISN       34            S24=0.D0
ISN       35            S25=0.D0
ISN       36            S26=0.D0
ISN       37            S27=0.D0
ISN       38            S31=0.D0
ISN       39            S32=0.D0
ISN       40            S33=0.D0
ISN       41            S34=0.D0
ISN       42            S35=0.D0
ISN       43            S36=0.D0
ISN       44            S37=0.D0
                C  (I,J,S)  (1,1,1),(1,2,2),(1,3,3),(2,2,4),(2,3,5),(3,3,6)
                C    ALPHA=BETA   ( , ,7)
ISN       45            DO 10 I=1,N3
ISN       46            R1=RM1*R(I,1)
ISN       47            R2=RM2*R(I,2)
ISN       48            R3=RM3*R(I,3)
ISN       49            IND=IH(I)
ISN       50            C1=R1+X
ISN       51            C2=R2+Y
ISN       52            C3=R3+Z
```

```
                *....*...1.....2.....3.....4.....5.....6.....7.*.....8
ISN        53           CC=C1*C1+C2*C2+C3*C3
ISN        54           DIST=DSQRT(CC)
ISN        55           CALL DDAZIZ(DIST,01,02)
ISN        56           PREF=DEXP(-0.5D0*(A1*R1*R1+A2*R2*R2+A3*R3*R3))
ISN        57           SS1=C1*C1*02*PREF/CC
ISN        58           SS2=C1*C2*02*PREF/CC
ISN        59           SS3=C1*C3*02*PREF/CC
ISN        60           SS4=C2*C2*02*PREF/CC
ISN        61           SS5=C2*C3*02*PREF/CC
ISN        62           SS6=C3*C3*02*PREF/CC
ISN        63           SS7=01*PREF
ISN        64           S31=S31+SS1
ISN        65           S32=S32+SS2
ISN        66           S33=S33+SS3
ISN        67           S34=S34+SS4
ISN        68           S35=S35+SS5
ISN        69           S36=S36+SS6
ISN        70           S37=S37+SS7
ISN        71           IF(IND.EQ.3)GOTO10
ISN        72           S21=S21+SS1
ISN        73           S22=S22+SS2
ISN        74           S23=S23+SS3
ISN        75           S24=S24+SS4
ISN        76           S25=S25+SS5
ISN        77           S26=S26+SS6
ISN        78           S27=S27+SS7
ISN        79           IF(IND.EQ.2)GOTO10
ISN        80           S11=S11+SS1
ISN        81           S12=S12+SS2
ISN        82           S13=S13+SS3
ISN        83           S14=S14+SS4
ISN        84           S15=S15+SS5
ISN        85           S16=S16+SS6
ISN        86           S17=S17+SS7
ISN        87     10    CONTINUE
ISN        88           PREF=-DSQRT(A1*A2*A3/(2.D0*PI)**3)
ISN        89           VOL=(4.D0*PI/3.D0)*RM1*RM2*RM3
ISN        90           P1=PREF*VOL/DFLOAT(45*N1)
ISN        91           P2=-20.D0*PREF*VOL/DFLOAT(45*N2)
ISN        92           P3=64.D0*PREF*VOL/DFLOAT(45*N3)
ISN        93           DD(ISHELL,1)=P1*S11+P2*S21+P3*S31
ISN        94           DD(ISHELL,2)=P1*S12+P2*S22+P3*S32
ISN        95           DD(ISHELL,3)=P1*S13+P2*S23+P3*S33
ISN        96           DD(ISHELL,4)=P1*S14+P2*S24+P3*S34
ISN        97           DD(ISHELL,5)=P1*S15+P2*S25+P3*S35
ISN        98           DD(ISHELL,6)=P1*S16+P2*S26+P3*S36
ISN        99           DD(ISHELL,7)=P1*S17+P2*S27+P3*S37
                  C    5 FORMAT(/,7(/,1X,3D20.10))
                  C      S11=S11/DFLOAT(N1)
                  C      S21=S21/DFLOAT(N2)
                  C      S31=S31/DFLOAT(N3)
                  C      S12=S12/DFLOAT(N1)
                  C      S22=S22/DFLOAT(N2)
                  C      S32=S32/DFLOAT(N3)
                  C      S13=S13/DFLOAT(N1)
                  C      S23=S23/DFLOAT(N2)
                  C      S33=S33/DFLOAT(N3)
                  C      S14=S14/DFLOAT(N1)
                  C      S24=S24/DFLOAT(N2)
                  C      S34=S34/DFLOAT(N3)
```

```
                *....*...1.........2.........3.........4.........5.........6.........7.*.......8

                C      S15=S15/DFLOAT(N1)
                C      S25=S25/DFLOAT(N2)
                C      S35=S35/DFLOAT(N3)
                C      S16=S16/DFLOAT(N1)
                C      S26=S26/DFLOAT(N2)
                C      S36=S36/DFLOAT(N3)
                C      S17=S17/DFLOAT(N1)
                C      S27=S27/DFLOAT(N2)
                C      S37=S37/DFLOAT(N3)
                C      WRITE(7,5)S11,S21,S31,S12,S22,S32,S13,S23,S33,S14,S24,S34,
                C     1          S15,S25,S35,S16,S26,S36,S17,S27,S37
ISN       100          WRITE(6,1)ISHELL,(DD(ISHELL,II),II=1,7)
ISN       101          WRITE(6,3)A(ISHELL,1),A(ISHELL,2),A(ISHELL,3)
ISN       102          WRITE(6,4)X,Y,Z
ISN       103     1000 CONTINUE
ISN       104          RETURN
ISN       105     1    FORMAT(' ISHELL,DD ',I2,3X,7D13.5)
ISN       106     2    FORMAT('1 AVERAGES USE N1,N2,N3',3I6,5X,'EPS =',D8.1,/,/)
ISN       107     3    FORMAT('      A1,A2,A3   ',3F13.7)
ISN       108     4    FORMAT('      R1,R2,R3   ',3F13.7,/)
ISN       109          END
```

STATISTICS SOURCE STATEMENTS = 108, PROGRAM SIZE = 3692 BYTES, PROGRAM NAME = FORCE1 PAGE: 3.

STATISTICS NO DIAGNOSTICS GENERATED.

FORCE1 END OF COMPILATION 3 ******

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF GOSTMT NODECK SOURCE TERM NOOBJECT FIXED NOTEST TRMFLG SRCFLG
NOSYM NORENT NOSDUMP AUTODBL(NONE) NOSXM
OPT(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN) LINECOUNT(64) CHARLEN(500)

```
                *....*...1.....2.....3.....4.....5.....6.....7.*.....8
ISN         1           SUBROUTINE GENS(NUM)
ISN         2           IMPLICIT REAL*4(A-H,O-Z)
ISN         3           COMMON/SPHERE/R(20000,3),IH(20000),N1,N2,N3
ISN         4           N1=0
ISN         5           N2=0
ISN         6           N3=0
ISN         7           LIM=2*NUM+1
ISN         8           DEL=1.0/FLOAT(NUM)
ISN         9           DO 10 I1=1,LIM
ISN        10           J1=I1-NUM-1
ISN        11           X=DEL*FLOAT(J1)
ISN        12           DO 10 I2=1,LIM
ISN        13           J2=I2-NUM-1
ISN        14           Y=DEL*FLOAT(J2)
ISN        15           DO 10 I3=1,LIM
ISN        16           J3=I3-NUM-1
ISN        17           Z=DEL*FLOAT(J3)
ISN        18           R2=X*X+Y*Y+Z*Z
ISN        19           IF(R2.GE.1.0001)GOTO10
ISN        20           I=3
ISN        21           N3=N3+1
ISN        22           IF(N3.GT.20000)STOP00320
ISN        24           R(N3,1)=X
ISN        25           R(N3,2)=Y
ISN        26           R(N3,3)=Z
ISN        27           IF(2*(J1/2).NE.J1)GOTO5
ISN        28           IF(2*(J2/2).NE.J2)GOTO5
ISN        29           IF(2*(J3/2).NE.J3)GOTO5
ISN        30           I=2
ISN        31           N2=N2+1
ISN        32           IF(4*(J1/4).NE.J1)GOTO5
ISN        33           IF(4*(J2/4).NE.J2)GOTO5
ISN        34           IF(4*(J3/4).NE.J3)GOTO5
ISN        35           I=1
ISN        36           N1=N1+1
ISN        37     5     IH(N3)=I
ISN        38    10     CONTINUE
ISN        39           RETURN
ISN        40           END
```

STATISTICS SOURCE STATEMENTS = 39, PROGRAM SIZE = 1354 BYTES, PROGRAM NAME = GENS PAGE: 6.

STATISTICS NO DIAGNOSTICS GENERATED.

GENS END OF COMPILATION 4 ******

```
OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  ) LINECOUNT(64)      CHARLEN(500)

                 *....*...1....+.....2.....3.....4.....5.....6.....7.*.....8

ISN        1           SUBROUTINE DAZIZ(R,O1,O2)
                   C   AZIZ CHEM POTENTIAL FOR ARGON   HFD-C
                   C   RETURNS O1,O2
                   C   J. CHEM. PHYS. 67,5719 (1977)
                   C   UNITS R IN ANGSTROMS , ENERGIES AZIZ IN DEGREES K
ISN        2           IMPLICIT REAL*8(A-H,O-Z)
                   C   DATA A/0.950720D7/
                   C   DATA ALPHA/16.34565D0/
                   C   DATA C6 /1.0914254D0/
                   C   DATA C8 /0.600259D0/
                   C   DATA C10/0.3700113D0/
                   C   DATA D/1.4D0/
                   C   DATA E/143.224D0/
                   C   DATA GAMMA/2.D0/
                   C   DATA RM/3.759D0/
                   C   KRYPTON  -- MOL. PHYS. 38,177 (1979)
ISN        3           DATA A/0.121531D8/
ISN        4           DATA ALPHA/16.49676D0/
ISN        5           DATA C6 /1.156173D0/
ISN        6           DATA C8 /0.541492D0/
ISN        7           DATA C10/0.2839735D0/
ISN        8           DATA D/1.28D0/
ISN        9           DATA E/199.90D0/
ISN       10           DATA GAMMA/2.4D0/
ISN       11           DATA RM/4.012D0/
ISN       12           X=R/RM
ISN       13           O1=0.D0
ISN       14           O2=0.D0
ISN       15           AX=ALPHA*X
ISN       16           IF(AX.GT.100.D0)GOTO5
ISN       17           PREF=A*X**(GAMMA-2.D0)*DEXP(-AX)
ISN       18           O1=PREF*(GAMMA-AX)
ISN       19           O2=PREF*(GAMMA*(GAMMA-1.D0)-AX*2.D0*GAMMA+AX*AX)
ISN       20      5    CONTINUE
ISN       21           X6=1.D0/X**6
ISN       22           X7=X6/X
ISN       23           X8=X7/X
ISN       24           X9=X8/X
ISN       25           X10=X9/X
ISN       26           X11=X10/X
ISN       27           X12=X11/X
ISN       28           D0=-C6*X6-C8*X8-C10*X10
ISN       29           D1=6.D0*C6*X7+8.D0*C8*X9+10.D0*C10*X11
ISN       30           D2=-42.D0*C6*X8-72.D0*C8*X10-110.D0*C10*X12
ISN       31           E0=1.D0
ISN       32           E1=0.D0
ISN       33           E2=0.D0
ISN       34           IF(X.GE.D)GOTO10
ISN       35           DX=D/X
ISN       36           E0=0.D0
ISN       37           IF(DX.GT.11)GOTO10
ISN       38           E0=DEXP(-(DX-1.D0)**2)
ISN       39           E1=2.D0*DX*(DX-1.D0)*E0/X
ISN       40           E2=2.D0*DX*(DX-1.D0)*E1/X+2.D0*DX*(2.D0-3.D0*DX)*E0/(X*X)
ISN       41      10   CONTINUE
ISN       42           O1=(O1+(D0*E1+E0*D1)/X)*E/(RM*RM)
```

```
                *....*...1.....2.....3.....4.....5.....6.....7.*.....8
ISN       43          O2=(O2+DO*E2+2.DO*D1*E1+D2*EO)*E/(RM*RM)-O1
ISN       44          RETURN
ISN       45          END
```

STATISTICS SOURCE STATEMENTS = 45, PROGRAM SIZE = 1594 BYTES, PROGRAM NAME = DAZIZ PAGE: 7.

STATISTICS NO DIAGNOSTICS GENERATED.

DAZIZ END OF COMPILATION 5 ******

```
OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  ) LINECOUNT(64)      CHARLEN(500)

                *....*...1.........2.........3.........4.........5.........6.........7.*.......8

ISN         1           SUBROUTINE GENECC(A,RAD)
ISN         2           IMPLICIT REAL*8(A-H,O-Z)
ISN         3           COMMON/ATOMS/R(2400,4),ISHELL(2400),RSHELL(80),DD(80,7),IS,N
ISN         4           DIMENSION INDEX(2400),TEMP(2400)
ISN         5           N=0
ISN         6           A2=0.5D0*A
ISN         7           IMAX=RAD/A2
ISN         8           IM=2*IMAX+1
ISN         9           DO 10 I11=1,IM
ISN        10           I1=I11-IMAX-1
ISN        11           DO 10 I12=1,IM
ISN        12           I2=I12-IMAX-1
ISN        13           DO 10 I13=1,IM
ISN        14           I3=I13-IMAX-1
ISN        15           X=DFLOAT(I2+I3)*A2
ISN        16           Y=DFLOAT(I1+I3)*A2
ISN        17           Z=DFLOAT(I1+I2)*A2
ISN        18           RR=DSQRT(X*X+Y*Y+Z*Z)
ISN        19           IF(RR.GT.RAD)GOTO10
ISN        20           N=N+1
ISN        21           IF(N.GT.2399)STOP00812
ISN        23           INDEX(N)=N
ISN        24           R(N,1)=X
ISN        25           R(N,2)=Y
ISN        26           R(N,3)=Z
ISN        27           R(N,4)=RR+DABS(X*Y*Z)*1.11D-5+1.3D-8*(DABS(X)+DABS(Y)+DABS(Z))
ISN        28     10    CONTINUE
                  C     HEAPIFY
ISN        29           MAX1=N/2
ISN        30           DO 20 K=1,MAX1
ISN        31           I=MAX1+1-K
ISN        32           J=2*I
ISN        33           ITEM=INDEX(I)
ISN        34           TEM=R(ITEM,4)
ISN        35     15    IF(J-N)17,18,16
ISN        36     17    IF(R(INDEX(J),4).LT.R(INDEX(J+1),4))J=J+1
ISN        38     18    IF(TEM.GE.R(INDEX(J),4))GOTO16
ISN        39           INDEX(J/2)=INDEX(J)
ISN        40           J=2*J
ISN        41           GOTO15
ISN        42     16    CONTINUE
ISN        43           INDEX(J/2)=ITEM
ISN        44     20    CONTINUE
                  C     SORT HEAP
ISN        45           N1=N-1
ISN        46           DO 30 I=1,N1
ISN        47           J=N-I+1
ISN        48           II=INDEX(J)
ISN        49           INDEX(J)=INDEX(1)
ISN        50           INDEX(1)=II
ISN        51           J=J-1
                  C     RESTORE HEAP
ISN        52           II=1
ISN        53     25    IL=2*II
ISN        54           IF(IL-J)31,35,30
ISN        55     31    IF(R(INDEX(IL),4).LT.R(INDEX(IL+1),4))IL=IL+1
```

```
                *....*...1.....2.....3.....4.....5.....6.....7.*.....8
ISN       57    35    IF(R(INDEX(II),4).GE.R(INDEX(IL),4))GOTO30
ISN       58          III=INDEX(II)
ISN       59          INDEX(II)=INDEX(IL)
ISN       60          INDEX(IL)=III
ISN       61          II=IL
ISN       62          GOTO25
ISN       63    30    CONTINUE
ISN       64          DO 40 I=1,4
ISN       65          DO 45 J=1,N
ISN       66    45    TEMP(J)=R(J,I)
ISN       67          DO 40 J=1,N
ISN       68          R(J,I)=TEMP(INDEX(J))
ISN       69    40    CONTINUE
ISN       70          IS=1
ISN       71          RR=R(1,4)
ISN       72          RSHELL(1)=0.D0
ISN       73          WRITE(6,2)IS,RSHELL(IS)
ISN       74          DO 50 I=2,N
ISN       75          ISHELL(I-1)=IS
ISN       76          IF((R(I,4)-RR).LT.1.D-9)GOTO50
ISN       77          IS=IS+1
ISN       78          IF(IS.GT.80)$TOP00800
ISN       80          RR=R(I,4)
ISN       81          RSHELL(IS)=DSQRT(R(I,1)**2+R(I,2)**2+R(I,3)**2)
ISN       82          WRITE(6,1)IS,RSHELL(IS)
ISN       83    50    CONTINUE
ISN       84          ISHELL(N)=IS
ISN       85          DO 60 I=1,N
ISN       86    60    R(I,4)=RSHELL(ISHELL(I))
ISN       87          ISHELL(N)=IS
ISN       88          RETURN
ISN       89    1     FORMAT(1X,I2,2X,F13.7)
ISN       90    2     FORMAT(/,/,' DISTANCES OF SHELLS USED',/,1X,I2,2X,F13.7)
ISN       91          END

*STATISTICS*   SOURCE STATEMENTS = 87, PROGRAM SIZE = 32186 BYTES, PROGRAM NAME = GENFCC    PAGE:    9.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**GENFCC** END OF COMPILATION 6 ******
```

```
OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF  GOSTMT NODECK  SOURCE  TERM NOOBJECT FIXED  NOTEST  TRMFLG  SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)  NOSXM
                      OPT(0) LANGLVL(77) NOFIPS  FLAG(I)  NAME(MAIN  )  LINECOUNT(64)  CHARLEN(500)

                *....*...1.....2.....3.....4.....5.....6.....7.*.....8

ISN         1           SUBROUTINE PHIAB(IGRID)
ISN         2           IMPLICIT REAL*8(A-H,O-Z)
ISN         3           COMMON/ATOMS/R(2400,4),ISHELL(2400),RSHELL(80),DD(80,7),IS,N
ISN         4           COMMON/WAVEFN/D(80,3,3),S(80,3,3),RL(80,3)
ISN         5           COMMON/DYNAM/PHI(80,3,3),INDEX(80)
ISN         6           DATA ITES/0/
ISN         7           IF(ITES.NE.0)GOTO5
ISN         8           ITES=1
ISN         9           DO 20 I=2,IS
ISN        10           DO 20 J=1,3
ISN        11   20      RL(I,J)=R(INDEX(I),J)
ISN        12   5       CONTINUE
ISN        13           CALL CALCS
ISN        14           CALL FORCE1(IGRID)
ISN        15           WRITE(6,1)
ISN        16           DO 30 I=2,IS
ISN        17           DO 25 J=1,3
ISN        18           DO 25 K=J,3
ISN        19           PHI(I,J,K)=S(I,1,J)*S(I,1,K)*DD(I,1)+S(I,2,J)*S(I,2,K)*DD(I,4)+
                       1 S(I,3,J)*S(I,3,K)*DD(I,6)+
                       2 (S(I,1,J)*S(I,2,K)+S(I,2,J)*S(I,1,K))*DD(I,2)+
                       3 (S(I,1,J)*S(I,3,K)+S(I,3,J)*S(I,1,K))*DD(I,3)+
                       4 (S(I,2,J)*S(I,3,K)+S(I,3,J)*S(I,2,K))*DD(I,5)
ISN        20           IF(J.EQ.K)PHI(I,J,K)=PHI(I,J,K)+DD(I,7)
ISN        22           IF(J.NE.K)PHI(I,K,J)=PHI(I,J,K)
ISN        24   25      CONTINUE
ISN        25           WRITE(6,2)I
ISN        26           WRITE(6,3)((PHI(I,J,K),K=1,3),J=1,3)
ISN        27   30      CONTINUE
ISN        28           RETURN
ISN        29   1       FORMAT('1  FORCE CONSTANTS FOLLOW FOR REFERENCE VECTORS',/)
ISN        30   2       FORMAT(/,/,/,'     SHELL ',I2,/)
ISN        31   3       FORMAT(3(5X,3F15.7,/))
ISN        32           END

*STATISTICS*   SOURCE STATEMENTS = 30, PROGRAM SIZE = 2398 BYTES, PROGRAM NAME = PHIAB      PAGE:   11.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**PHIAB** END OF COMPILATION 7 ******
```

```
OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF  GOSTMT NODECK  SOURCE  TERM NOOBJECT FIXED  NOTEST  TRMFLG  SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)  NOSXM
                      OPT(0) LANGLVL(77) NOFIPS  FLAG(I)  NAME(MAIN  )  LINECOUNT(60)  CHARLEN(500)

                *....*...1.....2.....3.....4.....5.....6.....7.*.....8

ISN       1          SUBROUTINE SYMPHI(PHIP,PHI,R1,R2)
ISN       2          IMPLICIT REAL*8(A-H,O-Z)
ISN       3          DIMENSION PHIP(3,3),PHI(3,3),R2(3),R1(3)
ISN       4          INTEGER P(3),S(3)
                C    SET UP P
                C    INDEX 1
ISN       5          IF(DABS(DABS(R2(1))-DABS(R1(1))).GT.1.D-8)GOTO5
ISN       6          P(1)=1
ISN       7          S(1)=1
ISN       8          IF(R2(1)*R1(1).LT.0.D0)S(1)=-1
ISN      10          GOTO100
ISN      11     5    IF(DABS(DABS(R2(1))-DABS(R1(2))).GT.1.D-8)GOTO10
ISN      12          P(1)=2
ISN      13          S(1)=1
ISN      14          IF(R2(1)*R1(2).LT.0.D0)S(1)=-1
ISN      16          GOTO100
ISN      17    10    IF(DABS(DABS(R2(1))-DABS(R1(3))).GT.1.D-8)GOTO15
ISN      18          P(1)=3
ISN      19          S(1)=1
ISN      20          IF(R2(1)*R1(3).LT.0.D0)S(1)=-1
ISN      22          GOTO100
ISN      23    15    CONTINUE
ISN      24          WRITE(6,1)(R1(I),I=1,3),(R2(I),I=1,3)
ISN      25     1    FORMAT(' CAN NOT DO SYMMETRY TRANSFORMATION'-- ABANDONING',/
                    1 ,' R1,R2',/,1X,3F13.5,/,1X,3F13.5)
ISN      26          STOP00514
ISN      27    100   CONTINUE
                C    INDEX 2
ISN      28          IF(DABS(DABS(R2(2))-DABS(R1(1))).GT.1.D-8)GOTO20
ISN      29          IF(P(1).EQ.1)GOTO20
ISN      30          P(2)=1
ISN      31          S(2)=1
ISN      32          IF(R2(2)*R1(1).LT.0.D0)S(2)=-1
ISN      34          GOTO200
ISN      35    20    IF(DABS(DABS(R2(2))-DABS(R1(2))).GT.1.D-8)GOTO25
ISN      36          IF(P(1).EQ.2)GOTO25
ISN      37          P(2)=2
ISN      38          S(2)=1
ISN      39          IF(R2(2)*R1(2).LT.0.D0)S(2)=-1
ISN      41          GOTO200
ISN      42    25    IF(DABS(DABS(R2(2))-DABS(R1(3))).GT.1.D-8)GOTO15
ISN      43          IF(P(1).EQ.3)GOTO15
ISN      44          P(2)=3
ISN      45          S(2)=1
ISN      46          IF(R2(2)*R1(3).LT.0.D0)S(2)=-1
ISN      48    200   CONTINUE
                C    INDEX 3
ISN      49          IF(DABS(DABS(R2(3))-DABS(R1(1))).GT.1.D-8)GOTO30
ISN      50          IF((P(1).EQ.1).OR.(P(2).EQ.1))GOTO30
ISN      51          P(3)=1
ISN      52          S(3)=1
ISN      53          IF(R2(3)*R1(1).LT.0.D0)S(3)=-1
ISN      55          GOTO300
ISN      56    30    IF(DABS(DABS(R2(3))-DABS(R1(2))).GT.1.D-8)GOTO35
ISN      57          IF((P(1).EQ.2).OR.(P(2).EQ.2))GOTO35
ISN      58          P(3)=2
```

```
                *....*...1.....2.....3.....4.....5.....6.....7.*.....8

ISN       59            S(3)=1
ISN       60            IF(R2(3)*R1(2).LT.0.D0)S(3)=-1
ISN       62            GOTO300
ISN       63    35      IF(DABS(DABS(R2(3))-DABS(R1(3))).GT.1.D-8)GOTO15
ISN       64            IF((P(1).EQ.3).OR.(P(2).EQ.3))GOTO15
ISN       65            P(3)=3
ISN       66            S(3)=1
ISN       67            IF(R2(3)*R1(3).LT.0.D0)S(3)=-1
ISN       69    300     CONTINUE
                C       SET.UP PHIP
ISN       70            DO 50 I=1,3
ISN       71            DO 50 J=1,3
ISN       72            PHIP(I,J)=DFLOAT(S(I)*S(J))*PHI(P(I),P(J))
ISN       73    50      CONTINUE
ISN       74            RETURN
ISN       75            END
```

STATISTICS SOURCE STATEMENTS = 66, PROGRAM SIZE = 2708 BYTES, PROGRAM NAME = SYMPHI PAGE: 12.

STATISTICS NO DIAGNOSTICS GENERATED.

SYMPHI END OF COMPILATION 8 ******

```
OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF  GOSTMT NODECK  SOURCE  TERM NOOBJECT FIXED  NOTEST  TRMFLG  SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)  NOSXM
                      OPT(0) LANGLVL(77) NOFIPS  FLAG(I)  NAME(MAIN  )  LINECOUNT(64)  CHARLEN(500)

                *....*...1.....2.....3.....4.....5.....6.....7.*.....8

ISN         1          SUBROUTINE READQ
ISN         2          IMPLICIT REAL*8(A-H,O-Z)
ISN         3          COMMON/VECTOR/Q(2000,3),WEIGHT(2000),NQ
ISN         4          COMMON/PARMS/XX(3),A0,RMAX,IGRID
ISN         5          DATA PI/3.14159265358979323D0/
ISN         6    1     FORMAT(4F10.0)
ISN         7    2     FORMAT(1X,I4,3F12.7,3X,F13.6)
ISN         8    3     FORMAT('1 Q-VECTORS AND WEIGHTS - UNITS 2*PI/A0 =',F9.6,/,/)
ISN         9          PREF=2.D0*PI/A0
ISN        10          NQ=0
ISN        11          WRITE(6,3)PREF
ISN        12    10    READ(5,1,END=20)(Q(NQ+1,I),I=1,3),WEIGHT(NQ+1)
ISN        13          NQ=NQ+1
ISN        14          WRITE(6,2)NQ,(Q(NQ,I),I=1,3),WEIGHT(NQ)
ISN        15          Q(NQ,1)=PREF*Q(NQ,1)
ISN        16          Q(NQ,2)=PREF*Q(NQ,2)
ISN        17          Q(NQ,3)=PREF*Q(NQ,3)
ISN        18          GOTO10
ISN        19    20    RETURN
ISN        20          END

*STATISTICS*  SOURCE STATEMENTS = 20, PROGRAM SIZE = 1186 BYTES, PROGRAM NAME = READQ  PAGE: 14.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**READQ** END OF COMPILATION 9 ******.
```

```
OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)  NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  ) LINECOUNT(64)      CHARLEN(500)

                *....*...1.....2.....3.....4.....5.....6.....7.*.....8

ISN         1          SUBROUTINE DYNAM1(WAV,FS)
ISN         2          IMPLICIT REAL*8(A-H,O-Z)
ISN         3          COMMON/VECTOR/Q(2000,3),WEIGHT(2000),NQ
ISN         4          COMMON/ATOMS/R(2400,4),ISHELL(2400),RSHELL(80),DD(80,7),IS,N
ISN         5          COMMON/WAVEFN/D(80,3,3),S(80,3,3),RL(80,3)
ISN         6          COMMON/DYNAM/PHI(80,3,3),INDEX(80)
ISN         7          COMMON/PARMS/BETA,HB,MASS,A0,RMAX,IGRID
ISN         8          REAL*8 MASS
ISN         9          DIMENSION DYN(3,3),VEC(3,3),PP(3,3),W2(3),IPER(6,3),QT(3)
ISN        10          DIMENSION ISG(3),W(3),WORK(10)
ISN        11          DATA ITES/0/
ISN        12          DATA DAMP/0.300D0/
ISN        13          IF(ITES.NE.0)GOTO7
ISN        14          IPER(1,1)=1
ISN        15          IPER(1,2)=2
ISN        16          IPER(1,3)=3
ISN        17          IPER(2,1)=1
ISN        18          IPER(2,2)=3
ISN        19          IPER(2,3)=2
ISN        20          IPER(3,1)=2
ISN        21          IPER(3,2)=1
ISN        22          IPER(3,3)=3
ISN        23          IPER(4,1)=2
ISN        24          IPER(4,2)=3
ISN        25          IPER(4,3)=1
ISN        26          IPER(5,1)=3
ISN        27          IPER(5,2)=1
ISN        28          IPER(5,3)=2
ISN        29          IPER(6,1)=3
ISN        30          IPER(6,2)=2
ISN        31          IPER(6,3)=1
ISN        32          CALL GENFCC(A0,RMAX)
ISN        33          CALL FORCE0
ISN        34          CALL READQ
ISN        35          ITES=1
ISN        36    7     CONTINUE
ISN        37          WAV=0.D0
ISN        38          FS=0.D0
ISN        39          IF(ITES.NE.1)CALL PHIAB(IGRID)
ISN        41          ITES=ITES+1
ISN        42          DO 5 I=2,IS
ISN        43          DO 5 J=1,3
ISN        44          DO 5 K=1,3
ISN        45    5     D(I,J,K)=DAMP*D(I,J,K)
ISN        46          WRITE(6,4)
ISN        47          DO 1000 IQ=1,NQ
ISN        48          DO 9 I=1,3
ISN        49          DO 9 J=1,3
ISN        50    9     DYN(I,J)=0.D0
ISN        51          DO 20 K=2,N
ISN        52          NSHELL=ISHELL(K)
ISN        53          IREF=INDEX(NSHELL)
ISN        54          DO 11 I=1,3
ISN        55          W2(I)=R(IREF,I)
ISN        56          W(I)=R(K,I)
ISN        57          DO 11 J=1,3
```

```
                 *....*...1.....2.....3.....4.....5.....6.....7.*.....8
ISN        58    11     VEC(I,J)=PHI(NSHELL,I,J)
ISN        59           CALL SYMPHI(PP,VEC,W2,W)
ISN        60           DO 15 I=1,3
ISN        61           DO 15 J=1,3
ISN        62           DYN(I,J)=DYN(I,J)+(DCOS(Q(IQ,1)*R(K,1)+Q(IQ,2)*R(K,2)+
                       1  Q(IQ,3)*R(K,3))-1.D0)*PP(I,J)/MASS
ISN        63    15     CONTINUE
ISN        64           DO 20 I=1,3
ISN        65           DO 20 J=1,3
ISN        66           IF(I.NE.J)DYN(J,I)=DYN(I,J)
ISN        68    20     CONTINUE
ISN        69           CALL EIGRS(DYN,3,11,W2,VEC,3,WORK,IER)
ISN        70           DO 22 I=1,3
ISN        71           IF(W2(I).GE.0.D0)GOTO22
ISN        72           WRITE(6,3)
ISN        73           WRITE(6,1)(W2(J),J=1,3)
ISN        74           STOP
ISN        75    22     W(I)=DSQRT(W2(I))
ISN        76           DO 25 II=1,6
ISN        77           DO 25 IS1=1,2
ISN        78           ISG(1)=2*IS1-3
ISN        79           DO 25 IS2=1,2
ISN        80           ISG(2)=2*IS2-3
ISN        81           DO 25 IS3=1,2
ISN        82           ISG(3)=2*IS3-3
ISN        83           DO 30 I=1,3
ISN        84           QT(I)=DFLOAT(ISG(I))*Q(IQ,IPER(II,I))
ISN        85           DO 30 J=1,3
ISN        86    30     DYN(I,J)=DFLOAT(ISG(J))*VEC(IPER(II,J),I)
ISN        87           DO 40 NSHELL=2,IS
ISN        88           IREF=INDEX(NSHELL)
ISN        89           PREF=HB*(1.D0-DCOS(QT(1)*R(IREF,1)+QT(2)*R(IREF,2)+
                       1 QT(3)*R(IREF,3)))*WEIGHT(IQ)*(1.D0-DAMP)/(48.D0*MASS)
ISN        90           DO 40 IBR=1,3
ISN        91           BHW=0.5D0*BETA*HB*W(IBR)
ISN        92           PREF2=DCOTH(BHW)/W(IBR)
ISN        93           PREF3=DLOG(2.D0*DSINH(BHW))/BETA-0.25D0*HB*W(IBR)*DCOTH(BHW)
ISN        94           FS=FS+PREF3*WEIGHT(IQ)
ISN        95           DO 40 I=1,3
ISN        96           DO 40 J=1,3
ISN        97           D(NSHELL,I,J)=D(NSHELL,I,J)+PREF*PREF2*DYN(IBR,I)*DYN(IBR,J)
ISN        98    40     CONTINUE
ISN        99    25     CONTINUE
ISN       100           WAV=WAV+(W(1)+W(2)+W(3))*WEIGHT(IQ)
ISN       101    1000   WRITE(6,2)(Q(IQ,I),I=1,3),(W(I),I=1,3)
ISN       102           WAV=WAV/DFLOAT(3)
ISN       103           WRITE(6,6)WAV,FS
ISN       104           RETURN
ISN       105    1      FORMAT(' W2',3F13.6)
ISN       106    2      FORMAT(1X,3F12.7,4X,3F12.7)
ISN       107    3      FORMAT('  FREQUENCY IMAGINARY .. ABANDONING CALCULATION  **')
ISN       108    4      FORMAT('1 Q-VECTORS AND FREQUENCIES FOLLOW - 3 BRANCHES',/)
ISN       109    6      FORMAT(/,/,' W-AVERAGE =',F11.8,5X,'FS =',F13.5)
ISN       110           END

*STATISTICS*   SOURCE STATEMENTS = 108, PROGRAM SIZE = 5180 BYTES, PROGRAM NAME = DYNAM1    PAGE:  15.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**DYNAM1** END OF COMPILATION 10 ******
```

```
OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(60)      CHARLEN(500)

                 *....*...1.....2.....3.....4.....5.....6.....7.*.....8

ISN         1          BLOCK DATA
ISN         2          IMPLICIT REAL*8(A-H,O-Z)
ISN         3          COMMON/WAVEFN/D(80,3,3),S(80,3,3),RL(80,3)
ISN         4          DATA D/720*0.D0/
ISN         5          END

*STATISTICS*   SOURCE STATEMENTS = 5, PROGRAM SIZE = 0 BYTES, PROGRAM NAME = BLKDT#     PAGE:   17.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**BLKDT#** END OF COMPILATION 11 ******
```

```
LEVEL 1.4.0 (OCT 1984)          VS FORTRAN          DATE: SEP 17, 1985    TIME: 08:16:09                    PAGE:    18

OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN   )  LINECOUNT(64)      CHARLEN(500)

                *....*...1.........2.........3.........4.........5.........6.........7.*.......8

ISN         1           SUBROUTINE READIN
ISN         2           IMPLICIT REAL*8(A-H,O-Z)
ISN         3           COMMON/PARMS/BETA,HBAR,MASS,A0,RMAX,IGRID
ISN         4           REAL*8 MASS
ISN         5           HBAR=7.6382829D0
ISN         6           BOLTK=1.D0
ISN         7           READ(5,1)MASS,A0,T,RMAX,IGRID
ISN         8           WRITE(6,2)
ISN         9           WRITE(6,3)MASS,A0,T,RMAX,IGRID
ISN        10           MASS=MASS*1.2027313D0
ISN        11           BETA=1.D0/(T*BOLTK)
ISN        12           WRITE(6,4)
ISN        13           RETURN
ISN        14    1      FORMAT(4E10.0,I5)
ISN        15    2      FORMAT(1H1)
ISN        16    3      FORMAT(' MASS =',F9.5,' A.M.U.',/,' A0   =',F9.5,' ANGSTROMS'
                     1      ,/,' TEMP =',F9.5,' DEGREES K',/,/,' RMAX,IGRID',F9.4,I4)
ISN        17    4      FORMAT(/,/,' ALL FREQUENCIES IN THZ  (10E12 HZ)')
ISN        18           END

*STATISTICS*    SOURCE STATEMENTS = 18, PROGRAM SIZE = 800 BYTES, PROGRAM NAME = READIN     PAGE:    18.

*STATISTICS*     NO DIAGNOSTICS GENERATED.

**READIN** END OF COMPILATION 12 ******
```

```
OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(64)      CHARLEN(500)

                *....*...1.....2.....3.....4.....5.....6.....7.*.....8

ISN         1           FUNCTION DCOTH(X)
ISN         2           IMPLICIT REAL*8(A-H,O-Z)
ISN         3           IF(X.GT.50.D0)GOTO5
ISN         4           IF(X.LT.-50.D0)GOTO10
ISN         5           E1=DEXP(2.D0*X)
ISN         6           DCOTH=(E1+1.D0)/(E1-1.D0)
ISN         7           RETURN
ISN         8     5     DCOTH=1.D0
ISN         9           RETURN
ISN        10    10     DCOTH=-1.D0
ISN        11           RETURN
ISN        12           END

*STATISTICS*   SOURCE STATEMENTS = 12, PROGRAM SIZE = 552 BYTES, PROGRAM NAME = DCOTH.    PAGE:   19.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**DCOTH** END OF COMPILATION 13 ******
```

```
OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE  TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(60)      CHARLEN(500)

                *....*...1.....2.....3.....4.....5.....6.....7.....8

ISN         1           SUBROUTINE DYNAM2
ISN         2           IMPLICIT REAL*8(A-H,O-Z)
ISN         3           COMMON/VECTOR/Q(2000,3),WEIGHT(2000),NQ
ISN         4           COMMON/ATOMS/R(2400,4),ISHELL(2400),RSHELL(80),DD(80,7),IS,N
ISN         5           COMMON/WAVEFN/D(80,3,3),S(80,3,3),RL(80,3)
ISN         6           COMMON/DYNAM/PHI(80,3,3),INDEX(80)
ISN         7           COMMON/PARMS/BETA,HB,MASS,AO,RMAX,IGRID
ISN         8           REAL*8 MASS
ISN         9           DIMENSION DYN(3,3),VEC(3,3),PP(3,3),W2(3),QT(3),WORK(10)
ISN        10           DIMENSION ISG(3),W(3)
ISN        11           DATA TWOPI/6.28318530800/
ISN        12           DATA CONV/4.1356700/
ISN        13           DO 1000 IQ=1,NQ
ISN        14           DO 9 I=1,3
ISN        15           DO 9 J=1,3
ISN        16     9     DYN(I,J)=0.DO
ISN        17           DO 20 K=2,N
ISN        18           NSHELL=ISHELL(K)
ISN        19           IREF=INDEX(NSHELL)
ISN        20           DO 11 I=1,3
ISN        21           W2(I)=R(IREF,I)
ISN        22           W(I)=R(K,I)
ISN        23           DO 11 J=1,3
ISN        24     11    VEC(I,J)=PHI(NSHELL,I,J)
ISN        25           CALL SYMPHI(PP,VEC,W2,W)
ISN        26           DO 15 I=1,3
ISN        27           DO 15 J=I,3
ISN        28           DYN(I,J)=DYN(I,J)+(DCOS(Q(IQ,1)*R(K,1)+Q(IQ,2)*R(K,2)+
                       1  Q(IQ,3)*R(K,3))-1.DO)*PP(I,J)/MASS
ISN        29     15    CONTINUE
ISN        30           DO 20 I=1,3
ISN        31           DO 20 J=I,3
ISN        32           IF(I.NE.J)DYN(J,I)=DYN(I,J)
ISN        34     20    CONTINUE
ISN        35           CALL EIGRS(DYN,3,11,W2,VEC,3,WORK,IER)
ISN        36           DO 22 I=1,3
ISN        37           IF(W2(I).GE.0.DO)GOTO22
ISN        38           WRITE(6,3)
ISN        39           WRITE(6,1)(W2(J),J=1,3)
ISN        40           STOP
ISN        41     22    W(I)=DSQRT(W2(I))/TWOPI
ISN        42           WRITE(6,2)(Q(IQ,I),I=1,3),(W(I),I=1,3)
ISN        43           W(1)=W(1)*CONV
ISN        44           W(2)=W(2)*CONV
ISN        45           W(3)=W(3)*CONV
ISN        46     1000  WRITE(6,4)W(1),W(2),W(3)
ISN        47           RETURN
ISN        48     1     FORMAT(' W2',3F13.6)
ISN        49     2     FORMAT(1X,3F12.7,4X,3F12.7)
ISN        50     3     FORMAT('  FREQUENCY IMAGINARY .. ABANDONING CALCULATION  **')
ISN        51     4     FORMAT(41X,3F12.7,/)
ISN        52           END

*STATISTICS*   SOURCE STATEMENTS = 51, PROGRAM SIZE = 2810 BYTES, PROGRAM NAME = DYNAM2     PAGE:   20.

*STATISTICS*    NO DIAGNOSTICS GENERATED.
```

DYNAM2 END OF COMPILATION 14 ******

```
OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK  SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                    *OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(64)      CHARLEN(500)

                *....*...1.....2.....3.....4.....5.....6.....7.*.....8

ISN         1           SUBROUTINE BRANCH
ISN         2           IMPLICIT REAL*8(A-H,O-Z)
ISN         3           COMMON/VECTOR/Q(2000,3),WEIGHT(2000),NQ
ISN         4           COMMON/PARMS/XX(3),A0,RMAX,IGRID
ISN         5           DATA PI/3.14159265358979323D0/
ISN         6           PREF=2.D0*PI/A0
ISN         7           NQ=20
ISN         8           WRITE(6,1)
ISN         9           WRITE(6,2)
ISN        10           DO 10 I=1,20
ISN        11           Q(I,1)=PREF*DFLOAT(I)/20.D0
ISN        12           Q(I,2)=0.D0
ISN        13    10     Q(I,3)=0.D0
ISN        14           CALL DYNAM2
ISN        15           WRITE(6,1)
ISN        16           WRITE(6,2)
ISN        17           DO 20 I=1,20
ISN        18           Q(I,1)=PREF*DFLOAT(I)/40.D0
ISN        19           Q(I,2)=PREF*DFLOAT(I)/40.D0
ISN        20    20     Q(I,3)=PREF*DFLOAT(I)/40.D0
ISN        21           CALL DYNAM2
ISN        22           WRITE(6,1)
ISN        23           WRITE(6,2)
ISN        24           DO 30 I=1,20
ISN        25           Q(I,1)=4.D0*PREF*DFLOAT(I)/80.D0
ISN        26           Q(I,2)=4.D0*PREF*DFLOAT(I)/80.D0
ISN        27    30     Q(I,3)=0.D0
ISN        28           CALL DYNAM2
ISN        29           RETURN
ISN        30    1      FORMAT('1 DOING SPECIAL DIRECTION',/)
ISN        31    2      FORMAT(19X,'Q',26X,'FREQUENCY(THZ) /  ENERGY(MEV)',/,/)
ISN        32           END

*STATISTICS*   SOURCE STATEMENTS = 32, PROGRAM SIZE = 1270 BYTES, PROGRAM NAME = BRANCH     PAGE:  22.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**BRANCH** END OF COMPILATION 15 ******
```

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF GOSTMT NODECK SOURCE TERM NOOBJECT FIXED NOTEST TRMFLG SRCFLG
NOSYM NORENT NOSDUMP AUTODBL(NONE) NOSXM
OPT(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN) LINECOUNT(64) CHARLEN(500)

```
                *....*...1.....2.....3.....4.....5.....6.....7.*.....8
ISN     1               FUNCTION AZIZ(R)
                C       AZIZ CHEN POTENTIAL FOR ARGON   HFD-C
                C       J. CHEM. PHYS. 67,5719 (1977)
                C       UNITS R IN ANGSTROMS , ENERGIES AZIZ IN DEGREES K
ISN     2               IMPLICIT REAL*8(A-H,O-Z)
                C       DATA A/0.9502720D7/
                C       DATA ALPHA/16.34565D0/
                C       DATA C6 /1.091425400/
                C       DATA C8 /0.600259500/
                C       DATA C10/0.370011300/
                C       DATA D/1.400/
                C       DATA E/143.22400/
                C       DATA GAMMA/2.00/
                C       DATA RM/3.75900/
                C       KRYPTON  -- MOL. PHYS. 38,177 (1979)
ISN     3               DATA A/0.121531208/
ISN     4               DATA ALPHA/16.496763D0/
ISN     5               DATA C6 /1.156173900/
ISN     6               DATA C8 /0.541492300/
ISN     7               DATA C10/0.283973500/
ISN     8               DATA D/1.2800/
ISN     9               DATA E/199.900D0/
ISN    10               DATA GAMMA/2.400/
ISN    11               DATA RM/4.01200/
ISN    12               X=R/RM
ISN    13               AZIZ=-(C6/X**6+C8/X**8+C10/X**10)
ISN    14               DX=D/X
ISN    15               IF((DX.GT.1.D0).AND.(DX.LT.11.D0))AZIZ=AZIZ*DEXP(-(DX-1.D0)**2)
ISN    17               IF(DX.GE.11.D0)AZIZ=0.D0
ISN    19               AX=ALPHA*X
ISN    20               IF(AX.GT.125.D0)GOTO10
ISN    21               AZIZ=AZIZ+A*DEXP(-AX)*X**GAMMA
ISN    22    10         CONTINUE
ISN    23               AZIZ=E*AZIZ
ISN    24               RETURN
ISN    25               END
```

STATISTICS SOURCE STATEMENTS = 23, PROGRAM SIZE = 904 BYTES, PROGRAM NAME = AZIZ PAGE: 23.

STATISTICS NO DIAGNOSTICS GENERATED.

AZIZ END OF COMPILATION 16 ******

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF GOSTMT NODECK SOURCE TERM NOOBJECT FIXED NOTEST TRMFLG SRCFLG
NOSYM NORENT NOSDUMP AUTODBL(NONE) NOSXM
OPT(2) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN) LINECOUNT(64) CHARLEN(500)

```
                *....*...1.....2.....3.....4.....5.....6.....7.*.....8
ISN         1           FUNCTION AVEO(IDUMMY)
ISN         2           IMPLICIT REAL*8(A-H,O-Z)
ISN         3           COMMON/SPHERE/R(20000,3),TH(20000),N1,N2,N3
ISN         4           COMMON/ATOMS/W(2400,4),ISH(2400),RSHELL(80),DD(80,7),IS,N
ISN         5           COMMON/WIDTH/A(80,3),RM(80,3)
ISN         6           REAL*4 R
ISN         7           DATA EPS/2.D-4/
ISN         8           DATA PI/3.1415926535897932300/
ISN         9           WRITE(6,2)
ISN        10           AVEO=0.DO
ISN        11           DO 1000 ISHELL=2,IS
ISN        12           X=RM(ISHELL,1)
ISN        13           Y=RM(ISHELL,2)
ISN        14           Z=RM(ISHELL,3)
ISN        15           A1=A(ISHELL,1)
ISN        16           A2=A(ISHELL,2)
ISN        17           A3=A(ISHELL,3)
ISN        18           RM1=DSQRT(-2.DO*DLOG(EPS)/A1)
ISN        19           RM2=DSQRT(-2.DO*DLOG(EPS)/A2)
ISN        20           RM3=DSQRT(-2.DO*DLOG(EPS)/A3)
ISN        21           S11=0.DO
ISN        22           S21=0.DO
ISN        23           S31=0.DO
ISN        24           DO 10 I=1,N3
ISN        25           R1=RM1*R(I,1)
ISN        26           R2=RM2*R(I,2)
ISN        27           R3=RM3*R(I,3)
ISN        28           C1=R1+X
ISN        29           C2=R2+Y
ISN        30           C3=R3+Z
ISN        31           IND=IH(I)
ISN        32           DIST=DSQRT(C1*C1+C2*C2+C3*C3)
ISN        33           PREF=DEXP(-0.500*(A1*R1*R1+A2*R2*R2+A3*R3*R3))
ISN        34           SS=AZIZ(DIST)*PREF
ISN        35           S31=S31+SS
ISN        36           IF(IND.EQ.3)GOTO10
ISN        37           S21=S21+SS
ISN        38           IF(IND.EQ.2)GOTO10
ISN        39           S11=S11+SS
ISN        40        10 CONTINUE
ISN        41           PREF=-DSQRT(A1*A2*A3/(2.DO*PI)**3)
ISN        42           VOL=(4.DO*PI/3.DO)*RM1*RM2*RM3
ISN        43           P1=PREF*VOL/DFLOAT(45*N1)
ISN        44           P2=-20.DO*PREF*VOL/DFLOAT(45*N2)
ISN        45           P3=64.DO*PREF*VOL/DFLOAT(45*N3)
ISN        46           AVERAG=P1*S11+P2*S21+P3*S31
ISN        47           WRITE(6,1)ISHELL,AVERAG
ISN        48           WRITE(6,3)A(ISHELL,1),A(ISHELL,2),A(ISHELL,3)
ISN        49           WRITE(6,4)X,Y,Z
ISN        50           AVEO=AVEO+AVERAG
ISN        51      1000 CONTINUE
ISN        52           WRITE(6,5)AVEO
ISN        53           RETURN
ISN        54         1 FORMAT(' ISHELL,AVERAGE ',I2,3X,D13.5)
ISN        55         2 FORMAT('1  COMPUTE AVERAGE OF AZIZ-CHEN POTENTIAL',/,/)
ISN        56         3 FORMAT('      A1,A2,A3  ',3F13.7)
```

```
                *....*...1.....2.....3.....4.....5.....6.....7.*.....8
ISN       57    4      FORMAT('      R1,R2,R3  ',3F13.7,/)
ISN       58    5      FORMAT(/,/,/,' RET AVEO =',F14.7)
ISN       59           END
```

STATISTICS SOURCE STATEMENTS = 59, PROGRAM SIZE = 2530 BYTES, PROGRAM NAME = AVEO PAGE: 24.

STATISTICS NO DIAGNOSTICS GENERATED.

AVEO END OF COMPILATION 17 ******

```
OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(60)      CHARLEN(500)

                *....*...1.....2.....3.....4.....5.....6.....7.*.....8

ISN         1           SUBROUTINE DDAZIZ(R,O1,O2)
            C  ,        AZIZ-CHEN POTENTIAL BY INTERPOLATION
ISN         2           IMPLICIT REAL*8(A-H,O-Z),LOGICAL(S)
ISN         3           DIMENSION X1(10000),X2(10000)
ISN         4           DATA SFIRST/.FALSE./
ISN         5           COMMON/ATOMS/W(9600),ISHELL(2400),RSHELL(80),DD(80,7),IS,N
ISN         6           IF(SFIRST)GOTO50
ISN         7           SFIRST=.TRUE.
ISN         8           RMAX=6.D0+RSHELL(IS)
ISN         9           DELTA=RMAX/DFLOAT(10000)
ISN        10           DO 10 I=1,10000
ISN        11           CALL DAZIZ(DELTA*DFLOAT(I),O1,O2)
ISN        12           X1(I)=O1
ISN        13  10       X2(I)=O2
ISN        14  50       CONTINUE
ISN        15           I=R/DELTA
ISN        16           IF(I.LE.1)GOTO60
ISN        17           IF(I-10000)30,20,40
ISN        18  20       I=I-1
ISN        19  30       RO=DELTA*DFLOAT(I)
ISN        20           DEL=R-RO
ISN        21           O1=X1(I)+DEL*(0.5D0*(X1(I+1)-X1(I-1))+DEL*(X1(I+1)+X1(I-1)-
                       1    2.D0*X1(I))/DELTA)/DELTA
ISN        22           O2=X2(I)+DEL*(0.5D0*(X2(I+1)-X2(I-1))+DEL*(X2(I+1)+X2(I-1)-
                       1    2.D0*X2(I))/DELTA)/DELTA
ISN        23           RETURN
ISN        24  40       O1=0.D0
ISN        25           O2=0.D0
ISN        26           RETURN
ISN        27  60       O1=X1(1)
ISN        28           O2=X2(1)
ISN        29           RETURN
ISN        30           END

*STATISTICS*   SOURCE STATEMENTS = 30, PROGRAM SIZE = 161286 BYTES, PROGRAM NAME = DDAZIZ    PAGE:   26.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**DDAZIZ** END OF COMPILATION 18 ******
```

```
OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE   TERM NOOBJECT FIXED  NOTEST   TRMFLG   SRCFLG
                    NOSYM NORENT NOSDUMP AUTODBL(NONE)   NOSXM
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(64)      CHARLEN(500)

                *....*...1.........2.........3.........4.........5.........6.........7.*.......8

ISN         1           SUBROUTINE FORCEO
ISN         2           IMPLICIT REAL*8(A-H,O-Z)
ISN         3           COMMON/ATOMS/R(2400,4),ISH(2400),RSHELL(80),DD(80,7),IS,N
ISN         4           COMMON/DYNAM/PHI(80,3,3),INDEX(80)
ISN         5           DIMENSION FOR(3,3)
ISN         6    1      FORMAT('1  NON-AVERAGED FORCE CONSTANTS - UNITS DEG K/A**2')
ISN         7    2      FORMAT(/,/,' SHELL',I3,/)
ISN         8    3      FORMAT(3(1X,3F15.7,/))
ISN         9    4      FORMAT('1 REFERENCE VECTORS FOR SHELLS 2 TO IS',/)
ISN        10    5      FORMAT(1X,I2,2X,3F13.7,F16.7)
ISN        11    6      FORMAT(/,/,/,' ***  NOW DO HARMONIC DISPERSION CURVES  ***')
ISN        12           INDEX(1)=1
ISN        13           J=1
ISN        14           DO 10 I=2,N
ISN        15           IF(ISH(I-1).EQ.ISH(I))GOTO10
ISN        16           J=J+1
ISN        17           INDEX(J)=I
ISN        18   10      CONTINUE
ISN        19           IF(J.NE.IS)STOP00700
ISN        21           WRITE(6,4)
ISN        22           DO 9 I=2,IS
ISN        23    9      WRITE(6,5)I,(R(INDEX(I),J),J=1,3),RSHELL(I)
ISN        24           WRITE(6,1)
ISN        25           DO 20 I=2,IS
ISN        26           RR=RSHELL(I)
ISN        27           WRITE(6,2)I
ISN        28           CALL DAZIX(RR,01,02)
ISN        29           DO 15 J=1,3
ISN        30           DO 15 K=1,3
ISN        31   15      FOR(J,K)=-R(INDEX(I),J)*R(INDEX(I),K)*02/RR**2
ISN        32           DO 16 J=1,3
ISN        33   16      FOR(J,J)=FOR(J,J)-01
ISN        34           DO 17 J=1,3
ISN        35           DO 17 K=1,3
ISN        36   17      PHI(I,J,K)=FOR(J,K)
ISN        37   20      WRITE(6,3)((FOR(J,K),K=1,3),J=1,3)
ISN        38           WRITE(6,6)
ISN        39           CALL BRANCH
ISN        40           RETURN
ISN        41           END

*STATISTICS*   SOURCE STATEMENTS = 40, PROGRAM SIZE = 2156 BYTES, PROGRAM NAME = FORCEO    PAGE: 27.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

**FORCEO** END OF COMPILATION 19 ******
```