EVALIN

A CDC 6600 FORTRAN IV CODE

TO PERFORM AN

EXTENDED VARIATIONAL APPROXIMATION FOR LIGHT

AND INTERMEDIATE NUCLEI

R.Y. CUSSON† & H.C. LEE

ATOMIC ENERGY OF CANADA LIMITED
CHALK RIVER NUCLEAR LABORATORIES
CHALK RIVER, ONTARIO, CANADA

AECL-3801

† *Present Address: Physics Department, Duke University, Durham, North Carolina, 27706, U.S.A.*

EVALIN

A CDC 6600 FORTRAN IV CODE TO PERFORM AN EXTENDED VARIATIONAL APPROXIMATION FOR LIGHT AND INTERMEDIATE NUCLEI

by

R.Y. Cusson and H.C. Lee

ABSTRACT

A computer code is described which performs an extended Hartree-Fock variational calculation for the intrinsic state of lighter nuclei having up to 82 protons and 82 neutrons. An exact treatment of the kinetic energy of the center-of-mass and the Coulomb two-body interaction is developed. The self-consistent orbitals can mix all the basis states including the time-reversed pairs but are restricted to having good parity; this results in an improved variational approximation which can be applied to odd nuclei. The neutron and proton wavefunctions are treated separately so as to exhibit the isospin violation due to the Coulomb force. The matrix elements of the nuclear interaction must be supplied as initial data. The initial trial wavefunction is taken to be that of the Nilsson model. Using the semi-realistic interaction No. 2 of Saunier & Pearson most calculated quantities are found to be within 10% of their experimental value, for nuclei with $4 \leq A \leq 40$.

Chalk River Nuclear Laboratories
Chalk River, Ontario
January, 1971

AECL-3801

EVALIN

Code FORTRAN IV (CDC 6600) permettant d'effectuer une approximation variationnelle étendue pour les noyaux légers et intermédiaires

par

R.Y. Cusson et H.C. Lee

Résumé

Le code d'ordinateur décrit dans ce rapport effectue un calcul variationnel Hartree-Fock étendu pour l'état intrinsèque de noyaux légers ayant jusqu'à 82 protons et 82 neutrons. Le traitement exact de l'énergie synétique du centre de masse et de l'interaction Coulomb à deux corps est expliqué. Les orbitales harmonieuses peuvent mélanger tous les états de base y compris les paires à temps inversé mais elles doivent avoir une bonne parité; ce qui permet d'obtenir une approximation variationnelle améliorée pouvant être appliquée aux noyaux impairs. Les fonctions d'onde des protons et des neutrons sont traitées séparément ce qui facilite la détection de toute violation de l'isospin due à la force de Coulomb. Les éléments de la matrice de l'interaction nucléaire doivent être fournis comme données initiales. La fonction ondique de l'essai initial est considérée comme étant celle du modèle de Nilsson. En utilisant l'interaction semi-réaliste N°2 de Saunier & Pearson, la plupart des quantités calculées sont cantonnées dans 10% de leur valeur expérimentale, pour les noyaux ayant $4 \leq A \leq 40$.

L'Energie Atomique du Canada, Limitée
Laboratoires Nucléaires de Chalk River
Chalk River, Ontario
Janvier, 1971

AECL-3801

1. INTRODUCTION

It has become increasingly clear over the past few years that quantitative solutions of the nuclear many-body problem cannot be obtained reliably in restricted configuration spaces. If we let $H = T+V^{(2)}$ be the nuclear Hamiltonian consisting of the one-body kinetic energy operator T, and the two-body interaction potential $V^{(2)}$, we must solve the Schroedinger equation

$$H|\Psi\rangle = E|\Psi\rangle \tag{1.1}$$

in order to obtain the energy eigenvalues and eigenfunctions of the many-nucleon system. Since $|\Psi\rangle$ is antisymmetric in the A nucleon coordinates, it may be expanded in a basis consisting of single Slater determinants (SSD). The first sentence of this introduction refers to the fact that no single-particle basis has ever been found such that only one SSD will constitute a quantitatively accurate representation for the components of $|\Psi\rangle$ up to about 300 MeV.

The probable reason for this lack of convergence of SSD's to $|\Psi\rangle$ is that $|\Psi\rangle$ exhibits two-body correlations, over both large and small distances, which cannot be represented by SSD's. If we now think of the many-body Hilbert space of states of the $|\Psi\rangle$'s as being a direct product of one-particle spaces we can distinguish roughly three ingredients which must be present in order to obtain an accurate

representation for $|\Psi\rangle$: (i) the single particle basis should have a large dimension, (ii) highly excited 2-particle-2 hole (2p-2h) configurations must be included to account for short range effects, (iii) many coherent (Np-Nh) low energy configurations will be required to account for long range correlations. The first ingredient is required in order to pin down the single-particle properties and also to provide the underlying space for the 2nd and 3rd ingredients.

The number of SSD's required to obtain an accurate expansion of the solutions of (1.1) in a given orthonormal many-body basis remains out of the bound of even present day large, fast computers. However it is still possible to make quantitative predictions about the states, even when the basis is truncated heavily, if a different effective Hamiltonian $\tilde{H}$, appropriate for the particular truncation[1] is used. In this case all other observables must also be represented by their effective counterpart[2] and often the burden of the problem is relegated to the operators instead of the states. For the purpose of low-energy nuclear spectroscopy it is believed that ingredient (ii) can be renormalised out of the wavefunctions and that a suitable K matrix[3,4,5] exists, which we will call $V^{(2)}$ from now on, such that only ingredients (i) and (iii) remain in $|\Psi\rangle$. It is further believed that, as long as short range effects are not involved, it will be possible to compute the matrix elements of any observables without renormalizing

them against effect (ii). But, if that last proviso holds, we need not have an exact mathematical prescription for the transition from the exact, highly singular interaction to our (smooth) $V^{(2)}$. Such a semi-realistic $V^{(2)}$ has been recently developed by Saunier & Pearson[5]. Their potential #2 (SP2) has been found to give excellent results[6-7] in the code described here.

Still, additional truncations with respect to effects (i) and (iii) are needed in order to deal with a manageable problem. We can distinguish roughly two approaches to this question. In shell model calculations the single particle basis is first heavily truncated, the interaction is then renormalised once more ($V^{(2)} \rightarrow G$) and as many configurations of type (iii) are retained as the size of the matrix diagonalisation problem will allow. The usual net result of such calculations is that one obtains a rather large (perhaps 15-30, in the s-d shell) number of eigenstates, with moderate accuracy for any single one, and requiring a substantial amount of type (i) renormalisation to obtain transition matrix elements (presence of effective charges). The second approach is the object of this report, and consists in retaining the largest possible single particle basis (effect (i)). One then concentrates on obtaining type (iii) correction for a very limited number (1-5, say) of eigenstates, usually near the ground state: to do this, the unrestricted Hartree-Fock variational method should include projection to restore the

symmetries which may be violated by the variational mixing. Although a smaller number of states are described in this way, the accuracy of the wavefunctions is improved, with the result that little or no operation renormalisation is required (unrenormalised transition operators may be used).

Actually, there are many properties of the ground state and features of the nuclear systematics which can be described without performing the projection step, and so even though the present code does not project true eigenstates, but only obtains the so-called "intrinsic state", a detailed description of the methods used is warranted both for its own sake and as a step towards further elaboration of the present work.

The main new features of the code are i) extensive use of the "Wigner-Eckart theorem", although deformed states are considered throughout, ii) a flexible modular construction where each subroutine package performs a given physical or mathematical step, iii) liberal use of random-access mass storage facilities, iv) considerable flexibility including a complete Nilsson model code (ELMOS[8]) forthe initialisation procedure which supplies the starting SSD trial wavefunction.

The first feature allows a several-fold increase in speed over the conventional methods, while the next two greatly facilitate extensions and modifications for special purposes. The last feature is of near vital

importance for such calculations because of the very large number of variational parameters involved.

The next section contains mostly a collection of the relevant mathematical equations involved in the Hartree-Fock approximation, while section 3 provides a guide to the structure of the program. Section 4 contains a brief discussion of the problems solved by this code. The appendices contain a listing of the code, a sample of the results, a list of error exit modes of termination, and finally, a short resume of the control card structure which is contained in a BLOCK DATA subprogram, instead of being read from cards.

An essential aspect of such a program is the preparation of the various two-body operator matrix elements which are required at the outset. A separate program EVAFME is also described here since it uses novel methods which considerably increase its speed and flexibility.

2. THE HARTREE-FOCK VARIATIONAL APPROXIMATION

2.1 THE BASIC EQUATIONS

The basis equations of the H.F. method are well known and we shall use mainly the notation and phase convention of Baranger[10]. Still, in order to completely define the numerical formalisms we shall review briefly the steps required to reduce the equations of the method to a computational algorithm.

We consider a nucleus composed of A nucleons (protons and neutrons of spin ½) whose Hamiltonian is the following sums of one and two body terms

$$H = \sum_{n=1}^{A} T(n) + \sum_{\substack{n=1 \\ m>n}}^{A} V(n,m)$$

$$= \sum_{n=1}^{A} T(n) + \frac{1}{2} \sum_{n\neq m=1}^{A} V(n,m) \quad , \tag{2.1}$$

where T(1) depends on the coordinates of the first particle only and V(1,2) depends (symmetrically) on the coordinates of particles 1 and 2 only, etc. Typically T(n) will be the kinetic energy of the n^{th} nucleon $\frac{1}{2m}|\underset{\sim}{p}_n|^2$, and V(1,2) is a complicated interaction between the two nucleons which will be discussed later on.

The coordinate (or classical) expression (2.1) needs to be rewritten in the quantum mechanical second quantisation formalism. We let $c^{\dagger}_{\alpha}, c_{\alpha}$ be a set of fermion creation and destruction operators obeying the usual anticommutation relations and we define our single particle basis

$$\langle \underset{\sim}{r}|\alpha\rangle = \langle \underset{\sim}{r}|c^{\dagger}_{\alpha}|0\rangle = \phi_{\alpha}(\underset{\sim}{r}), \qquad \alpha = 1, ---, B \ . \tag{2.2}$$

We now state the precise definition of $\phi_{\alpha}(r)$ which has been used throughout in this work. The index α stands for 4 labels: n_a, the radial label, j_a, m_a the total angular momentum and its projection along the z axis, and t_z, the isospin projection, which is $-\frac{1}{2}$ for neutrons and $+\frac{1}{2}$ for protons. This last label will be omitted because our basis will be the same for protons and neutrons. The structure of the basis used here is given in table 1. So we define

$$\langle \underset{\sim}{r}|njm\rangle = R_{n\ell}(\tfrac{r}{b})(i)^{\ell} \sum_{\mu=-\frac{1}{2}}^{+\frac{1}{2}} Y_{\ell}^{m-\mu}(\theta,\phi)\chi_{\frac{1}{2}}^{\mu} \times C(\ell\tfrac{1}{2}j;m-\mu,\mu,m), \tag{2.3}$$

where $R_{n\ell}(r)$ is the normalised radial wavefunction for a particle in a 3-dimensional harmonic oscillator of frequency $\hbar\omega$, and oscillator parameter $b = \sqrt{\frac{\hbar}{m\omega}}$. Following Brody & Moshinsky[11] we have for $R_{n\ell}(\rho)$, $(\rho = \frac{r}{b})$

$$R_{n\ell}(\rho) = \left[\frac{(2n)!}{\Gamma(n+\ell+3/2)}\right]^{\frac{1}{2}} \rho^{\ell}\, e^{-\frac{1}{2}\rho^2}\, L_n^{\ell+\frac{1}{2}}(\rho^2), \qquad (2.4)$$

where $L_n(z)$ is given as

$$L_n^{a}(z) = \frac{1}{n!}\, e^{z}\, z^{-a}\, \frac{\partial^n}{\partial z^n}\,(e^{-z}\, z^{n+a}) \ . \qquad (2.5)$$

The number n counts the number of sign changes of the radial wavefunction and $R_{n\ell}$ is positive at the origin. The $Y^m(\theta,\phi)$ are the usual spherical harmonics of Condon & Shortley and $\chi^{\mu}_{\frac{1}{2}}$ is the appropriate spinor. Under the time reversal operator T we have

$$T(i^{\ell}Y_{\ell}^{m}) = (-1)^{\ell-m}(i^{\ell}Y_{\ell}^{-m}), \quad T|\chi^{\mu}_{\frac{1}{2}}\rangle = (-1)^{\frac{1}{2}-\mu}|\chi^{-\mu}_{\frac{1}{2}}\rangle. \qquad (2.6)$$

The last symbol of (2.3) is the usual Clebsch-Gordan coefficient for recoupling of two angular momenta.

The wavefunctions defined by (2.3) agree in all respects but one with the usual definitions used in current shell model calculations. The factor i^{ℓ} has been introduced in order to insure that under time reversal the states $|njm\rangle$ will transform as

$$T|njm\rangle = (-1)^{j-m}|nj-m\rangle \ . \qquad (2.7)$$

This last equation is important in the present calculations where time-reversal properties are not conserved in the variational procedure.

Using c_α and $c^\dagger_\alpha$ it can be shown that eq. (2.1) can be rewritten as

$$H = \sum_{\alpha\beta} \langle\alpha|T(1)|\beta\rangle c^\dagger_\alpha c_\beta + \tfrac{1}{2} \sum_{\alpha\beta\gamma\delta} \langle\alpha\beta|V(1,2)|\gamma\delta\rangle c^\dagger_\alpha c^\dagger_\beta c_\delta c_\gamma, \quad (2.8)$$

where

$$\langle\alpha|T(1)|\beta\rangle = \int d^3\underset{\sim}{r}_1 \phi^*_\alpha(\underset{\sim}{r}_1) T(\underset{\sim}{r}_1) \phi_\beta(\underset{\sim}{r}_1),$$

and

$$\langle\alpha\beta|V(1,2)|\gamma\delta\rangle = \int d^3\underset{\sim}{r}_1 d^3\underset{\sim}{r}_2 \phi^*_\alpha(\underset{\sim}{r}_1)\phi^*_\beta(\underset{\sim}{r}_2) V(\underset{\sim}{r}_1,\underset{\sim}{r}_2)\phi_\gamma(\underset{\sim}{r}_1)\phi_\delta(\underset{\sim}{r}_2). \quad (2.9)$$

We next defined the antisymmetrised, unnormalised matrix elements $V_{\alpha\beta\gamma\delta}$ as

$$\begin{aligned} V_{\alpha\beta\gamma\delta} &= \langle\alpha\beta|V(1,2)|\gamma\delta\rangle - \langle\alpha\beta|V(1,2)|\delta\gamma\rangle \\ &= -V_{\beta\alpha\gamma\delta} = -V_{\alpha\beta\delta\gamma} = V_{\beta\alpha\delta\gamma}\ , \end{aligned} \quad (2.10)$$

so that (2.8) takes the convenient and standard form

$$H = \sum_{\alpha\beta} T_{\alpha\beta} c^\dagger_\alpha c_\beta + \tfrac{1}{4} \sum_{\alpha\beta\gamma\delta} V_{\alpha\beta\gamma\delta}\ c^\dagger_\alpha c^\dagger_\beta c_\delta c_\gamma\ . \quad (2.11)$$

The rotational invariance of H can be used to apply the Wigner-Eckart theorem to $V_{\alpha\beta\gamma\delta}$, giving the decompositions

$$V_{\alpha\beta\gamma\delta} = \sum_{JM} F(acdbJ) s_\beta s_\gamma C(j_a j_c J: m_a, \bar{m}_c M)\ C(j_d j_b J; m_d, \bar{m}_b, M)$$

$$= \sum_{JM} G(abcdJ) C(j_a j_b J; m_a m_b M) C(j_c j_d J; m_c m_d M), \tag{2.12}$$

where the second decomposition is the usual shell model one, apart from the normalisation, while the first one is more appropriate for our purpose. Note that $s_\alpha = (-1)^{j_\alpha - m_\alpha}$ and $\bar{m}_\alpha = -m_\alpha$. The F reduced matrix elements may be obtained from the G's with the relation

$$F(acdbJ') = -\sum_J (2J+1) W(j_a j_b j_c j_d; JJ') G(bacdJ). \tag{2.13}$$

Let us now consider a single Slater determinant (SSD) trial wavefunction, for A particles, of the form

$$|\phi_A\rangle = \xi^\dagger_{\lambda_1} \xi^\dagger_{\lambda_2} \cdots\cdots \xi^\dagger_{\lambda_A} |0\rangle, \tag{2.14}$$

where

$$\xi^\dagger_\lambda = \sum_\alpha U_{\alpha\lambda} c^\dagger_\alpha , \quad \lambda = 1, \cdots, A , \tag{2.15}$$

and $U_{\alpha\lambda}$ is a (complex) unitary transformation of the (fixed, laboratory, expansion) basis. The one-body density matrix $\hat{\rho}_{\alpha\beta} = c^\dagger_\beta c_\alpha$ has, on this state, the average value

$$\rho_{\alpha\beta} = \langle\phi_A|\hat{\rho}_{\alpha\beta}|\phi_A\rangle = \langle\phi_A|c^\dagger_\beta c_\alpha|\phi_A\rangle = \sum_{\lambda \text{ occupied}} U_{\alpha\lambda} U^*_{\beta\lambda} . \tag{2.16}$$

The average density defines the average potential energy matrix

$$\bar{v}_{\alpha\gamma} = \sum_{\beta\delta} V_{\alpha\beta\gamma\delta}\, \rho_{\delta\beta} \ . \qquad (2.17)$$

This potential together with $T_{\alpha\gamma}$ defines the so-called Hartree-Fock Hamiltonian $h_{\alpha\gamma}$ as

$$h_{\alpha\gamma} = T_{\alpha\gamma} + \bar{v}_{\alpha\gamma} \ . \qquad (2.17a)$$

The variational minimum to

$$\delta_U \langle \phi_A | H | \phi_A \rangle = 0 \ , \qquad (2.18)$$

where one varies with respect to the parameters of the unitary transformation $U_{\alpha\lambda}$ can be shown to lead to the self-consistent eigenvalue equations

$$\sum_{\beta} h_{\alpha\beta}\, U_{\beta\lambda} = \varepsilon_\lambda U_{\alpha\lambda}, \qquad (2.19)$$

where the eigenvalues ε_λ are called the Hartree-Fock single particle energies. The equation (2.19) implies a self-consistency condition because the matrix $h_{\alpha\beta}$ to be diagonalised is itself a function of $U_{\alpha\lambda}$, through equations (2.17) and (2.16). The self consistent U is found by first selecting an initial U^0 then computing h from it and diagonalising it to get a new U^1 by populating the states of lowest ε_λ. This process is iterated until no further change in U is detected.

The eq. (2.17) is not practical to use when B is large because there are B^4 or, in this case, approximately 45 million matrix elements $V_{\alpha\beta\gamma\delta}$. We shall want to use only the F matrix elements of eq. (2.12). This can be accomplished by using a slight variation of the Wigner-Eckart theorem for $\rho_{\alpha\beta}$ and $\bar{v}_{\alpha\beta}$. We write

$$\langle \alpha j_a m_a | \bar{v} | \beta j_b m_b \rangle = \bar{v}_{\alpha\beta} = s_\alpha \sum_J C(j_b j_a J; m_b, \bar{m}_a, M)(j_a \| \bar{v}_J^M \| j_b) \quad (2.21)$$

as the defining equation for the "reduced matrix elements" $(j_a \| \bar{v}_J^M \| j_b)$. A similar equation defines $(j_a \| \rho_J^M \| j_b)$. We substitute these in (2.17) and use (2.12) to obtain the new equation

$$(j_a \| \bar{v}_J^M \| d_c) = \sum_{bd} F(ca,db;J)(j_b \| \rho_J^M \| j_d) \quad (2.22)$$

This equation involves far fewer matrix elements of the potential ($\tilde{v}$ 15000 in the present case) and was used in the code. The inverse of eq. (2.21) is also required and comes out, using the orthogonality of the C.G. coefficients as

$$(j_a | \bar{v}_J^M \| j_b) = \sum_{m_c} C(j_c j_a J; m_c, M-m_c)(-1)^{j_a+M-m_c} \langle j_a, -M+m_c | \bar{v} | j_c m_c \rangle . \quad (2.23)$$

The value of $\langle \phi_A | H | \phi_A \rangle$ can be computed and we obtain

$$E_{H.F.} = \langle\phi_A|H|\phi_A\rangle = \sum_{\alpha\beta} (T_{\alpha\beta} + \tfrac{1}{2}\,\bar{v}_{\alpha\beta})\rho_{\beta\alpha} \,. \qquad (2.24)$$

These, then, are the basic equations which were used in the code. The remainder of this section will be devoted to the symmetry properties and other matters which influence the structure of the code.

The F matrix elements have the symmetry $F(ab,cd;J) = F(cd,ab;J)$ and thus where stored as the lower triangular parts of symmetric matrices in the double index (a,b), for each J value.

The density matrix $\rho_{\alpha\beta}$ is hermitian, as can be seen from its definition (2.16). This implies the symmetry property for its reduced matrix element $(j_a\|\rho_J^M\|j_b)$

$$(j_c\|\rho_J^{-M}\|j_a) = \theta(j_c-j_a-M)(j_a\|\rho_J^M\|j_c)^*, \qquad (2.25)$$

with $\theta(a+b+\cdots) = (-1)^{a+b+\cdots}$, so that only the matrix elements with $M \geq 0$ need be stored. This property is useful when computing the traces involved in eq. (2.24). In terms of reduced matrix elements we have

$$\mathrm{Tr}[\rho\bar{v}] = \sum_{\alpha\beta} \rho_{\alpha\beta}\bar{v}_{\beta\alpha} = \sum_J \sum_{\substack{j_a j_b \\ M\geq 0}} (2-\delta_{M,0})\,\mathrm{Re}[(j_b\|\rho_J^M\|j_a)(j_b\|\bar{v}_J^M\|j_a)^*] \qquad (2.26)$$

This equation is used to define the partial components (J,M) of the trace sum. The equation (2.23) also serves as a natural definition of the spherical tensor parts of the deformed Hartree-Fock potential.

2.2 THE MATRIX ELEMENTS OF 1 AND 2-BODY OPERATORS

Apart from the matrix elements of the one-body kinetic energy and the two-body interaction discussed earlier, several other sets of matrix elements are used in the calculation.

The matrix elements of the following one-body operators are constructed:

i) the 21 symplectic operators of Lipkin[12]:

$r_k r_\ell$ (6 operators), $p_k p_\ell$ (6 more), $r_k p_\ell$ (9) $k,\ell = 1,2,3$

ii) $r^L Y^M_L(\theta,\phi)$, $L = 0,2,4,6$, $M = 0,2,\cdots, L$.

The set i) is a general purpose one which is used to perform cranking operations and to obtain the free kinetic energy. The second set ii) is used to evaluate the multipole moments of the intrinsic state $|\phi_A>$. The set i) is obtained by relating r_i and p_i to harmonic oscillator creation and annihilation operators in the cartesian oscillator basis. Linear combinations of the operators above were computed as follows, in units of b^2,

$$N_\lambda = \sum_{ij=1}^{3} \Omega_{\lambda ij} \left(\frac{r_j p_i + p_i r_j}{2}\right) \qquad \lambda = 1,\ldots, 9 , \tag{2.27a}$$

$$A^{\pm} = \sum_{ij=1}^{3} \Omega_{\lambda ij} \left(\frac{p_j p_i \pm r_j r_i}{2}\right) \qquad \lambda = 4,\ldots, 9 , \tag{2.27b}$$

where the 9 3×3 matrices $\underset{\approx}{\Omega}_\lambda$ are

$$\underset{\approx}{\Omega}_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \quad \underset{\approx}{\Omega}_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \quad \underset{\approx}{\Omega}_3 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$\Omega_4 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \Omega_5 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 2 \end{bmatrix}, \quad \underset{\approx}{\Omega}_6 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix},$$

$$\underset{\approx}{\Omega}_7 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad \Omega_8 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \Omega_9 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The precise definition of the set ii) is

$$Q_L^M = r^L \sqrt{\frac{4\pi}{2L+1}}\, Y_L^M(\theta,\phi) = r^L\, C_L^M(\theta,\phi), \quad L \text{ even}$$

and its matrix elements are given by the formula

$$\langle n'\ell'j'm'|Q_L^M|n\ell jm\rangle = (-1)^{\frac{\ell-\ell'}{2}+j-m}\, C(j'jk;m'\bar{m}M)(n'\ell'|r^L|n\ell) \times \frac{(\ell'j'\|C(k)\|\ell j)}{\sqrt{2L+1}} \tag{2.28}$$

The radial matrix element is evaluated in the usual manner while the angular reduced matrix element is given by Edmonds[13]. The additional factor $(-1)^{\frac{\ell-\ell'}{2}}$ reflects the phase convention adopted here.

The computation of the matrix elements of the two-body operators used in the code is much more lengthy.

Apart from the nuclear potential the following operators are needed:

$$V_R^{(2)} = \frac{1}{2} \sum_{n \neq m}^{A} |\underset{\sim}{r}^n - \underset{\sim}{r}^m|^2 ,$$

$$V_P^{(2)} = \frac{1}{2} \sum_{n \neq m}^{A} |\underset{\sim}{p}^n - \underset{\sim}{p}^m|^2 , \qquad (2.29)$$

$$V_C^{(2)} = \frac{1}{2} \sum_{n \neq m}^{Z} \frac{1}{|\underset{\sim}{r}^n - \underset{\sim}{r}^m|} .$$

The first two operators are needed to treat the center of mass problem; their direct and exchange matrix elements must be calculated separately in the second quantisation formalism; thus two sets of V_R and V_P must be obtained. The last one is the Coulomb two-body potential. Since this distinction holds also for the nuclear interaction seven sets of two-body matrix elements, each of which contains ∿15000 matrix elements in our basis, must be obtained. They are computed by a separate code EVAFME (EVALIN F-Matrix Element) in a suitable order for reading in as data by EVALIN. Section 2.5 discusses some of the important features of this calculation.

2.3 THE CENTER-OF-MASS PROBLEM

The initial trial wavefunction supplied by ELMOS very nearly puts the center-of-mass (CM) of the nucleus in the ground state (0s) of a harmonic oscillator of the same frequency as $\hbar\omega$, the basis frequency. Now the one-body kinetic energy

$$T_{tot} = \sum_n \frac{|p_n|^2}{2m}$$

contains T_{CM}, the kinetic energy of the CM. If we iterate using T_{tot} the final wavefunction will also tend to minimise T_{CM}, which means spreading the CM out over all the space allowed by the basis. This will confuse the interpretation of the single particle wavefunctions since they will then tend to smear out too. This effect can be corrected either by subtracting T_{CM} from T_{tot} or by adding to T_{tot}, V_{CM}

$$V_{CM} = \frac{1}{2} M \omega_{CM}^2 |\underset{\sim}{R}|^2, \quad \underset{\sim}{R} = \sum_{n=1}^{A} \underset{\sim}{r}^{(n)} \frac{1}{A}, \tag{2.29}$$

which will bind the CM in a harmonic oscillator potential. The present code uses the first method; the second one is useful when it can not be established that the initial SSD has the desired CM wavefunction. The form of $T_{rel} = T_{tot} - T_{CM}$ can be obtained as

$$T_{rel} = \sum_n \frac{|p^n|^2}{2m} - \frac{1}{2mA} \sum_{nm} \underset{\sim}{p}^n \cdot \underset{\sim}{p}^{m'} = \frac{1}{4mA} \sum_{n \neq m} |\underset{\sim}{p}^n - \underset{\sim}{p}^{m'}|^2 = \frac{1}{2mA} V_P^{(2)} \quad . \tag{2.30}$$

This last equation shows that the Hamiltonian in the relative coordinate system has no one-body components but depends on the atomic number A.

The properties of the CM wavefunction enter again when computing the mean-square radius. One derives the identity

$$\overline{r_{rel}^2} = \frac{1}{A} \sum_{n=1}^{A} |\underset{\sim}{r}^n - \underset{\sim}{R}|^2 = \frac{1}{A^2} V_R^{(2)} . \tag{2.31}$$

The difference between (2.31) and the simplified expression $\frac{1}{A} \sum_n |\underset{\sim}{r}^n|^2$ goes to zero for heavy nuclei but is appreciable for light nuclei, as can be seen in appendix II. A formula similar to eq. (2.31) has been worked out to compute the mean squared radius of the *protons only* with respect to the *total* CM, a quantity of interest in most experimental determinations of the charge radius of nuclei.

2.4 PROTON & NEUTRON DENSITY MATRICES & THE PARITY SELECTION RULES

The equations (2.15) and (2.16) for the density matrix did not differentiate between neutrons and protons whereas the equation (2.22) involves F(ca,db;J) where a,b,c,d carry a proton or neutron label. Some detail about the actual procedure will now be given.

We can take, as a complete set of labels of our single particle states, the quantum numbers (n,j^{π},m,t_z). Next we restrict the U matrix of eq. (2.15) to be diagonal in the parity label and in the proton neutron label, that is to say the single particle Hartree-Fock states are either neutrons or protons of good parity. It is then clear from eq. (2.16) that the total density matrix breaks up into two submatrices $\underset{\approx}{\rho}_p$ and $\underset{\approx}{\rho}_n$ each of which breaks up further into two pieces, one density matrix for the positive parity states and one for the negative parity ones. The reduced matrix elements of ρ, ρ_p and ρ_n of a given J^{π} therefore contain only J^+ terms since $\pi_J = \pi_a\pi_b$. The complete set of reduced matrix elements of F in (2.22) has the labels $F(ct_c,at_a,dt_d,bt_b,J^{\pi})$. From the properties of ρ and the eq. (2.22) we see that only the J^+ matrix elements with the pair db both protons or both neutrons are needed. By conservation of charge this means that the pairs ca are also both neutrons or both protons. Since we have the symmetry relation

$$F(ca,db;J) = F(db,ca;J), \tag{2.32}$$

the F matrix breaks up into the submatrices

$$F = \begin{bmatrix} F^J_{pp} & F^J_{pn} \\ F^J_{np} & F^J_{nn} \end{bmatrix}, \tag{2.33}$$

where the subscripts label the two particle hole pairs (ca and db).

Furthermore if we let ρ,σ stand for the two pairs ca,db it can be shown that

$$F^J_{pp}(\rho,\sigma) = F^J_{pp}(\sigma,\rho) = F^J_{nn}(\rho,\sigma) = F^J_{nn}(\sigma,\rho) , \tag{2.34}$$

and

$$F^J_{pn}(\rho,\sigma) = F^J_{pn}(\sigma,\rho) = F^J_{np}(\rho,\sigma) = F^J_{np}(\sigma,\rho) . \tag{2.34a}$$

Using the ρ,σ,η notation for pairs of labels, the equation (2.23) can now be rewritten as

$$\begin{aligned} \bar{v}^M_{Jp}(\sigma) &= \sum_n (F^J_{pp}(\sigma,\eta)\rho^M_{Jp}(\eta) + F_{pn}(\sigma,\eta)\rho^M_{Jn}(\eta)), \\ \bar{v}^M_{Jn}(\sigma) &= \sum_n (F^J_{pn}(\sigma,\eta)\rho^M_{Jp}(\eta) + F_{nn}(\sigma,\eta)\rho^M_{Jn}(\eta))) . \end{aligned} \tag{2.35}$$

This last equation was used in the code.

2.5 THE CONSTRUCTION OF TWO-BODY MATRIX ELEMENTS

Perhaps the most significant feature of the program EVAFME is its use of *exact* expressions for the construction of all the required two-body matrix elements of operators which do not involve tensor terms, namely the operators of eq. (2.29). The method of Talman[14] has been used with success.

For any two-body potential $V^{(1,2)}$, the anti-symmetrized F-matrix element can be written as

$$F(ab,cd;J) = f(abcdJ) - (-)^{j_a+j_d+J} \sum_{J'} (2J'+1)(-)^{J'} W(j_a j_b j_c j_d;JJ')f(acdbJ'), \qquad (2.36)$$

where

$$f(abcdJ) = \sum_{\substack{m_a-m_b=M \\ m_c-m_d=M}} \langle\alpha\delta|V^{(1,2)}|\beta\gamma\rangle\, C(j_a j_b J;m_a \bar{m}_b M) \times C(j_c j_d J;m_c \bar{m}_d M) s_\beta s_\delta. \qquad (2.37)$$

When V(1,2) is a central force, i.e. $V(1,2) = V(|\underset{\sim}{r}_1-\underset{\sim}{r}_2|)$, Talman's scheme yields the formula

$$f(abcdJ) = c_a c_b c_c c_d (-)^{j_a - j_c + (\ell_s + \ell_c - \ell_a - \ell_d)/2} \frac{\hat{j}_a \hat{j}_b \hat{j}_c \hat{j}_d}{(2J+1)} \times$$

$$W(\ell_a \ell_b j_a j_b; J\tfrac{1}{2}) W(\ell_c \ell_d j_c j_d; J\tfrac{1}{2}) \sum_{NM\nu} C^{ab}_{NJ} C^{cd}_{MJ} R(NMJ\nu) I_\nu(\nu), \quad (2.38)$$

where $\hat{j}_a = \sqrt{2j_a+1}$; we refer the reader to Talman's[14] equations (4), (14) and (15) for the computational definitions of the normalisation coefficients c_a, the transformation coefficients C^{ab}_{NJ} and $R(NMJ\nu)$ respectively. Here I_ν is the Talmi integral

$$I_\nu = 4\pi \int_0^\infty r^{2\nu+2} e^{-r^2} V(\sqrt{2}\, r) dr \, . \quad (2.39)$$

For example the case $V_C(1,2) = \frac{e^2}{|\underline{r}_1 - \underline{r}_2|}$ gives

$$I_\nu(V_C) = e^2 \frac{\pi}{\sqrt{2}} \nu! \, ,$$

for the Talmi integrals of the Coulomb potential. The equation (2.38) implies the useful selection rules $J + \ell_a + \ell_b$ = even, $\ell_c + \ell_d + J$ = even, and the symmetry relations

$$f(abcdJ) = f(cdabJ) = -\theta(abJ) f(bacdJ).$$

3. CODING TECHNIQUES

3.1 GENERAL DESCRIPTION

The program EVALIN performs an Extended Variational Approximation for Light and Intermediate Nuclei. The size of the basis and the nuclear parameters are initial data, much of which is read in from magnetic tape at an early stage of execution. This tape must be prepared by EVAFME on a previous run. We first give the block structure of the tape.

The matrix elements of (2.29) are computed in dimensionless form and so are independent of $\hbar\omega$. They are first on the tape and require 6 blocks of data. There follows a variable number of sections consisting of 4 blocks each and pertaining to the various values of $\hbar\omega$ for which nuclear force matrix elements, in MeV, have been constructed. The first block of the dimensionless M.E. section is a 25 word control block containing the number of j shells to be considered (up to 16) and the corresponding lengths of the following 5 blocks. They in turn contain $V^{(2)}_{Cpp}$, $V^{(2)}_{Rpp}$, $V^{(2)}_{Rpn}$, $V^{(2)}_{Ppp}$, $V^{(2)}_{Ppn}$ of equation (2.29). The 4-block sections are composed of a control block, the G-matrix elements, F_{pp}, F_{pn}. We note that the G matrix elements are included for reference purposes only and are not used here. These matrix elements are *not* constructed by EVAFME either, EVAFME reads them in from magnetic tape or failing that the unsatisfied external PPME(I1,I2,I3,I4,J,KT,GGG,0) must be coded in as a supplier of the matrix elements of the two-body nuclear interaction.

After initialisation the overlay READIN is called and the initial data is transferred to a mass storage device (disk). Upon returning to the main overlay (EVALIN) the common block /BASIS/ is set up and contains detailed information related to the selected basis states (lengths of arrays etc....). Several nuclei (up to 40) may be processed in one run. This main loop begins at DO 11 etc. -- .

The processing of one nucleus begins with the generation of a trial wavefunction. It may be read from cards by calling READIN once more or constructed from the Nilsson model by calling the overlay ELMOSE[8]. ELMOSE is called in any case in order to generate the one-body matrix elements defined in equations (2.27), and the printout character set.

The overlay NUHART is next called to perform the actual iterations. Several iterations are first performed using the relative coordinate total Hamiltonian. The results of this are considered to be the "best" wavefunction which may be saved permanently on punched cards. The wavefunction is also saved temporarily (until the next nucleus) on mass storage. After computing the relative coordinate rms radii a last iteration is performed with the sole purpose of obtaining the usual single particle energies (which include the K.E. of the center of mass). Finally, further iterations may be performed in the presence of a one-body external field (cranking).

The control is then transferred to the overlay MULTIP which computes the expectation values of the multipole moments defined by (2.28).

The final task is to produce a computer output plot of the density distribution defined by the saved wavefunctions BRPS,BIPS,BRNS,BINS through equation (2.16). The code is then ready for the next nucleus.

A listing of the codes EVALIN and EVAFME is given in appendix I. Most of the actual coding details can be obtained by studying this listing together with a sample of the output for ^{4}He, given in appendix II. Appendices III and IV pertain to the execution details while appendix V describes a magnetic tape containing the card images for the code. The following subsections provide some further relevant details for the computation.

3.2 WORKING ARRAYS & MASS STORAGE RETRIEVAL

The finite size of the central memory available restricts the code to having simultaneously in core only one set of two-body matrix elements and two sets of one-body matrix elements (we view the wavefunction $U_{\alpha\gamma}$ as a one-body array.) For this purpose one long (15000) array and two short ones (4000) are used as temporary storage only. All arrays of interest are stored on disk and transferred in and out by referring to their alphanumerical name with BUFFIN and BUFOUT. It is then practical to pass to subroutines only the names of the arrays of interest via the calling sequence.

The limitations on the size of the temporary storage are felt most severely when constructing the complete Hamiltonian including Coulomb interactions, since we must then take linear combinations of two (long) sets of two-body matrix elements. A subroutine ADLONG performs these operations piecemeal.

3.3 THE STRUCTURE OF F ARRAYS

Apart from containing the two-body matrix elements of some operator, an F array contains control data. These data are checked during processing to insure that garbage has not been read from mass storage.

We have

F(1) = NV = number of levels in the basis

F(I+1) = I = code number of the levels described in table I.

There follows J_{max} blocks for each positive parity J value occurring in (2.12) ($J_{max} \leq 7$). The first element of each block is set to 64*J + DIMJ, where DIMJ = the number of different pairs (j_α, j_γ) which can couple to J^+. Within a block the element $F^J(L,M)$ is located at the position L + DIMJ*(M-1)-M(M-1)/2, $L \geq M$. In the pair index $L \leftrightarrow (j_\alpha, j_\gamma)$ the index α varies fastest in increasing order over the values of I which are allowed.

3.4 CONSTRUCTION OF REDUCED MATRIX ELEMENTS

Although most of the calculations are performed using the reduced matrix elements of $\bar{v}$ and ρ the diagonalisation involved in (2.19) must be performed using $\langle j_\alpha|\bar{v}|j_\beta\rangle$.

The coding for equations (2.21) and (2.23) is represented by the subroutines ROCONV, ROCONVP, ROCONIP, IDACSET, HISCLEB.

3.5 MISCELLANEOUS

The construction of the density matrix (eq. (2.16)) is coded in DENMAT and DENMATP. This pair of subroutines illustrates the technique of double calls to treat subblocks (in this case positive and negative parity blocks) of a matrix.

The selection of the orbits to be populated at each iteration is done by REORDER. The Z protons and N neutrons are simply put in the states with lowest $\varepsilon_{\lambda p}$ and $\varepsilon_{\lambda n}$.

The machine language code FTNUTIL is a general purpose debugging and tracing code of considerable usefulness when testing modifications to the program.

The overlay ELMOSE will not be discussed in this report since its coding techniques have been described elsewhere[8].

The subroutine HARFIT(ITE) performs ITE complete iterations including diagonalisations etc..., after which it extracts the single-particle r.m.s. radii, the deformation parameters and the expectation values of the 21 symplectic operators.

The subroutine TALMI in EVAFME constructs appropriately normalised talmi integrals instead of using the simpler expression (2.39), in order to improve the numerical stability of the sum defined by eq. (2.38).

The function TRVEROJ(RO,VB,---) returns the trace of $\rho^{*}\bar{v}$ and prints out the intermediate J sums multiplied by 0.5. This factor is not present in the total returned sum.

It will be seen that a judicious use of the routines VEEBAR and TRVEROJ allows the computation of the expectation value of *any* two-body operator whose reduced F matrix elements are in mass storage. For example this feature is used to compute the Coulomb energy and the relative coordinate rms radii.

4. RESULTS & DISCUSSION

The code has proved to be numerically quite stable and it has been found that 6-8 iterations would, in most cases, stabilise the wavefunction to about 1% or better. The most sensitive cases are those for a mid-shell odd-nucleus where the energy gap between the last occupied state and the unoccupied ones is small.

Results have been obtained using the K.L.S. G-matrix[4] in an N = 0-2, 3 major shells basis and using the Saunier-Pearson[5] potential No. 2 with N = 0-2 and N = 0-4. For both interactions the N = 0-2 results on s-d shell nuclei were in good agreement (4% or better) with those of Pal & Stamp[3], thus providing a check of the code.

Our best results have been obtained for nuclei in the range $4 \leq A \leq 40$, with N = 0-4 and using the S.P. No. 2 potential. For most of these nuclei the calculated total binding energy is within 10% of the experimental values (the calculated nuclei are *less* bound) and these nuclei have an experimental r.m.s. radius about 5% greater than the calculated ones. The computed values of the deformation parameters β and γ are in good agreement ($\pm 25\%$) with experiment when known. The position of the Fermi surface has been found to be usually within an MeV or two of the separation energy.

A striking feature of many of the density distributions determined here is that strong alpha clustering effects have been observed. In particular ^{8}Be, $^{12}C^{*}$(7.67 MeV), $^{16}O^{*}$(6.06 MeV) ^{20}Ne show promise of being quite good alpha cluster nuclei.

Perhaps the most severe limitation of the present method is that only the properties of the intrinsic state are obtained. It is probable that further improvements on the binding energies and r.m.s. radii can be obtained by using the generator coordinate method of Hill & Wheeler to project states of good angular momentum out of the intrinsic state obtained here. It is also probable that an increase in the maximum size of the basis up to N = 6,7 would improve the results on clustering effects and allow the study of much heavier nuclei, and higher multipole moments.

Another application of the code can be made to a detailed study of the **effects of the Coulomb** force, as a source of violation of isospin conservation. Preliminary results on its influence on the difference of the neutron and proton radii have proved interesting[7].

A detailed assessment of the usefulness of the code as a source of accurate nuclear wavefunctions will be published elsewhere[6].

ACKNOWLEDGMENTS

It is a pleasure to acknowledge the active cooperation of Dr. D. McPherson and the entire staff of the Mathematics & Computation Branch of C.R.N.L. We have also benefited from many conversations with Dr. M. Harvey, Dr. F.C. Khanna and Dr. S.S.M. Wong.

REFERENCES

1. See Chapter 5 entitled "Free & Effective Nucleon Interactions" in the Proc. Int. Conf. on Prop. of Nucl. States, ed. M. Harvey et al., University of Montreal Press, Montreal (1960).

2. B.H. Brandow, Rev. Mod. Phys. 39(1967)771.

3. C.W. Nestor et al, Nucl. Phys. A113(1968)14.
K.T.R. Davies & M. Baranger, Phys. Rev. C-1(1970)1640.
M.K. Pal & A.P. Stamp, Phys. Rev. 158(1467)924.
R.J. McCarthy & K.T.R. Davies, Phys. Rev. C-1(1970)1644.

4. S. Kahana, H.C. Lee & C.K. Scott, Phys. Rev. 185(1969)1378.

5. G. Saunier & J.M. Pearson, Phys. Rev. C-1(1970)1353.

6. H.C. Lee & R.Y. Cusson (submitted to Phys. Rev.).

7. H.C. Lee & R.Y. Cusson (submitted to Physics Letters).

8. R.Y. Cusson, AECL Report #3730 (1970).

9. D.R. Hartree, Proc. Cambridge Phil. Soc. 24(1928)89.
V.J. Fock, Zeit. fur Phys. 61(1930)126.

10. B. Baranger, Phys. Rev. 120(1960)957; Phys. Rev. 122 (1961)992; 1962 Cargese Lectures on Theoretical Physics, M. Levy ed., (Benjamin, N.Y. 1963).

11. T.A. Brody & M. Moshinsky, "Tables of Transformation Brackets" (Monografias del Instituto de Fisica, Mexico, 1960).

12. S. Goshen & H.J. Lipkin, Ann. of Phys. 6(1959)301.

13. A.R. Edmonds, "Angular Momentum in Quantum Mechanics", Princeton, (1957).

14. J.D. Talman, Nucl. Phys. A141(1970)273.

APPENDIX I

LISTING OF THE CODE

The following 56 pages contain a listing of EVALIN and EVAFME. A magnetic tape containing the card images of this listing is described in appendix V.

```
      OVERLAY(EVALIN,0,0)
      BLOCK DATA EVALI
*
*     EVALI ***MUST*** BE THE FIRST SUBR. OF THE (0,0) OVERLAY .
      COMMON/NUHARN/VALN(82),VALP(82),NP(2),NN(2),NAME,ITER,ITCR,ATNO
     1 ,NORBA(41),NORB,BET,GAMD,NPTOT,NNTOT
      COMMON/CONTRSN/ICON(2,40),RNU(5,40),IBLK,HBOW,TNAME,HBWA(30)
      COMMON/EXFIN/EXFID(2,4,20)
      COMMON/PRINLIM/EPS,IPR,QNO(82),IPUNCH
      LOGICAL IPUNCH
*
      DATA IPUNCH/.T./
*
      DATA EPS,IPR/.003, 0/
*
*     DATA ICON/5,1,0,0/
      DATA ICON/1,1,0,0/
*
      DATA EXFID/0/
*
      DATA IBLK/1/
*
*     DATA ITER,ITCR/6,2/
      DATA ITER,ITCR/2,2/
*
*     DATA TNAME,HBWA/10H020SP2,16L,13.5,17.5,20.0,10.5,25.0,8.0,30.0,
      DATA TNAME,HBWA/10H021SP2,6LV,13.5,17.5,20.0,10.5,25.0,8.0,30.0,
     1 15.0,0.00/
*
      DATA RNU/2.,2.,5LHE  4.,.001,0.,4.,4.,5LBE  8,.70,0.,                1,2
     1 6.,6.,5LC  12,.5,180.,          8.,8.,5LO  16,.001,0.,              3,4
     2 10.,10.,5LNE 20,.45,0.,         12.,12.,5LMG 24,.45,15.,            5,6
     3 14.,14.,5LSI 28,.50,180.,       16.,16.,5LS  32,.38,44. ,           7,8
     4 18.,18.,5LAR 36,.18,180.,       20.,20.,5LCA 40,.001,0.,            9,10
     5 20.,24.,5LCA 44,.15,0.,         20.,28.,5LCA 48,.001,0.,            11,12
     6 40.,50.,5LZR 90,.001,0.,        50.,66.,5LSN116,.12,0.,             13,14
     7 6.,6.,5LC* 12,1.16,0.,          8.,8.,5LO* 16,.832,26.,             15,16
     8 8.,12.,5LO  20,.25,10.,         8.,11.,5LO  19,.25,0.,              17,18
*    8 8.,12.,5LO  20,.25,10.,         8.,14.,5LO  24,.25,170.,            17,18
     9 2.,8.,5LHE 10,.001,0.,          0.,8.,5LZ   8,.001,0.,              19,20
     0 1.,1.,5LH   2,.001,0.,          0.,2.,5LZ   2,.001,0.,              21,22
     1 8.,8.,5LO 16*,1.46,0.,          10.,10.,6LNE 20*,1.7,0.,            23,24
     2 8.,9.,5LO  17,.05,0.,           8.,10.,5LO  18,.15,0.,              25,26
     3 2.,3.,5LHE  5,.32,0.,           3.,3.,5LLI  6,.52,0.,               27,28
     4 5.,5.,5LB  10,.50,35.,          7.,7.,5LN  14,.05,180.,             29,30
     5 7.,8.,5LN  15,.05,180.,         8.,7.,5LO  15,.05,180.,             31,32
     6 8.,9.,5LO  17,.10,0.,           8.,10.,5LO  18,.25,0.,              33,34
     7 9.,8.,5LF  17,.05,0.,           9.,9.,5LF  18,.15,0.,               35,36
     8 10.,8.,5LNE 18,.25,0.,          11.,11.,5LNA 22,.45,0.,             37,38
     8 5*0./
      END
      PROGRAM EVALIN(OUTPUT,TAPE1,TAPE9=1,PUNCH=400B,INPUT=400B)
*
*     A CODE TO DO COMPLEX HARTREE-FOCK ITERATIONS.
*     TAPE1 MUST HAVE BEEN PRESET BY EVAFME .
*     LIMITATIONS, NOLV.LE.16.ICODE MUST CONTAIN BOTH + AND- PARITIES .
      COMMON/OVERCON/ FIRST,ISKP
      COMMON/CONTRSN/ICON(2,40),RNU(5,40),IBLK,HBOW,TNAME,HBWA(30)
      COMMON/BASIS/ ICODE(16),ISTA(16),IPAR(16),INPO(16),INUC(16),
     1 ILC(16),NOLV,NB(2),MUD,IDF,IDSC,IDSJ,JAR(14,2),IDAC(14,14),JMPO,
     2 NBT,HBOM,OSPA2
      COMMON/NUHARN/VALN(82),VALP(82),NP(2),NN(2),NAME,ITER,ITCR,ATNO
```

```
     1  ,NORBA(41),NORB,RET,GAMD,NPTOT,NNTOT
        LOGICAL FIRST
        COMMON/EXSYPN/EXF(6,6),SYP(21),NOMM(20,3),COMM(20)

        COMMON/CODATPR/IBAQ(30)
        REAL NAME$ DIMENSION COCO(4)
        COMMON/NAMEN/FPR,PTS,FPF,FNE,FNF,FDN,PTD,FDF,RONI,ROPI,SPE,SPEXF
     1  , RTS,RTD
        COMMON/CLEBDAT/ALOGF(30),SQRF(30)
        COMMON/QUADSN/ QUADS(5,3)
        DATA          FPR,PTS,FPF,FNE,FNF,FDN,PTD,FDF,RONI,ROPI,SPE,SPEXF
     1  ,RTS,RTD
     1/3LFPR,3LPTS,3LFPF,3LFNE,3LFNF,3LFDN,3LPTD,3LFDF,4LRONI,4LROPI,
     1  3LKEN,5LSPEXF,3LRTS,3LRTD/
        DATA SYP/3LNL1,3LNL2,3LNL3,
     1  3LNL4,3LNL5,3LNL6,3LNL7,3LNL8,3LNL9,
     1  3LAP4,3LAP5,3LAP6,3LAP7,3LAP8,3LAP9,
     1  3LAM4,3LAM5,3LAM6,3LAM7,3LAM8,3LAM9/
        DATA COMM/     -1.,  1.,  1.,-1.,-1.,  1.,  1.,-1.,-1.,-1.,2.,9*1./
        DATA NOMM/1,1,2,2,3,3,4,4,5,5,5,6,6,7,7,8,8,9,9,9,
     1            2,3,1,3,1,2,1,2,1,2,3,2,3,1,3,1,2,1,2,3,
     2            3,2,3,1,2,1,1,2,1,2,3,3,2,3,1,2,1,1,2,3/
        DATA IBAQ/
     1  10100, 20201, 10201, 30302, 10300, 20302, 40403, 20401, 30403,
     1  10401, 50504, 30502, 40504, 10500, 60605, 20502, 50605, 40603,
     1  70706, 20601, 30603, 10601, 60706, 50704, 80807, 30702, 40704,
     1  10700, 20702, 70807/
        DATA EXF/4LR1R1,4LR2R1,4LR3R1,4LP1R1,4LP2R1,4LP3R1,
     1           4LR2R1,4LR2R2,4LR3R2,4LP1R2,4LP2R2,4LP3R2,
     2           4LR3R1,4LR3R2,4LR3R3,4LP1R3,4LP2R3,4LP3R3,
     3           4LP1R1,4LP1R2,4LP1R3,4LP1P1,4LP2P1,4LP3P1,
     4           4LP2R1,4LP2R2,4LP2R3,4LP2P1,4LP2P2,4LP3P2,
     5           4LP3R1,4LP3R2,4LP3R3,4LP3P1,4LP3P2,4LP3P3/
        DATA ELMOSE,NUHART,DENSITY,READIN,PROJECT,MULTIP/
     1    6LELMOSE,6LNUHART,7LDENSITY,6LREADIN,7LPROJECT,6LMULTIP/
        DATA MUD,FIRST/2,.T./,ISKP/-1/,COCO/4*0/
        CALL SFL(1100008)
        ALOGF(1)=0.$ SQRF(1)=1.$ DO 21 K = 2 , 30$ SQRF(K)= SQRT(FLOAT(K))
 21     ALOGF(K)= ALOGF(K-1)+ ALOG(FLOAT(K-1))
        PRINT 100
 100    FORMAT(1HR)
        PRINT 101
 101    FORMAT(////40X10H********** * MULTI-SHELL H.F. CALC. *
     1  10H**********  ///)
        CALL OVERLAY(READIN,3,0)
*       RETURN 1
        IF(IDF.LT.1) CALL ERREXIT(IDF)
        OSPA2= 41.6/HBOM
        PRINT 1006, HBOM
 1006   FORMAT(/* OSC. ENERGY= *F10.2,* MEV.*/)
*
*       INITIALISE THE HF PROCEDURE.
        NB(1)=NB(2)=0$ DO 13 K = 1 , NOLV$ KP= ICODE(K)$ IQ= IBAQ(KP)
        JPH= IQ/10000$ IT= MOD(IQ,10000)$ NT= IT/100$ LT= MOD(IT,100)
        ISTA(K)= JPH$ IPAR(K)= 2-(LT.A.1)$ INPO(K)= NT$ ILC(K)= LT
        IPT= 1+(LT.A.1)
 13     NB(IPT)= NB(IPT)+ SHIFT(JPH,1) $ NBT= NB(1)+NB(2)
        IDSC= NB(1)**2+NB(2)**2
*
        DO 11 IN= 1 , 40$ IF(ICON(1,IN).EQ.0) GO TO 99
        ISKP= ICON(2,IN)$ ILA= ICON(1,IN)$ IF(ILA.LT.1) CALL ERREXIT(ILA)
        NPTOT= RNU(1,ILA)$ NNTOT= RNU(2,ILA)$ NAME= RNU(3,ILA)
```

```
      COCO(1)= NAME$ CALL REMARK(COCO)
      GAMD= RNU(5,ILA)$ ATNO= NPTOT+NNTOT$ BET= RNU(4,ILA)
      NP(1)= NPTOT$ NN(1)= NNTOT$ NP(2)=NN(2)=0
      NORB= (MAX0(NPTOT,NNTOT)+1)/2$ DO 1 K = 1 , NORB
    1 NORBA(K)= K  $ IF(ISKP.EQ.0.A..N.FIRST) GO TO 5
      CALL OVERLAY(ELMOSE,1,0)
    5 IF(ISKP.EQ.0) CALL OVERLAY(READIN,3,0)

      CALL OVERLAY(NUHART,2,0)
      CALL OVERLAY(MULTIP,6,0)
      CALL OVERLAY(DENSITY,4,0)
      FIRST= .F.
   11 CONTINUE
   99   CONTINUE
      END
      SUBROUTINE ADLONG(FL1,C1,FL2,C2,FLT,AS1,AL1)
      COMMON/BASIS/ ICODE(16),ISTA(16),IPAR(16),INPO(16),INUC(16),
     1 ILC(16),NOLV,NB(2),MUD,IDF,IDSC,IDSJ,JAR(14,2),IDAC(14,14),JMPO,
     2 NBT,HBOM,OSPA2
      DIMENSION AS1(1),AL1(1)
      DATA FLD/3LFTD/
      IDS= MIN0(IDF,8000)$ IF(SHIFT(IDS,1).LT.IDF) CALL ERREXIT(IDS,IDF)
      CALL BUFFIN(FL1,AL1,IDF)$ CALL BUFFIN(FL2,AS1,IDS)
      DO 1 K = 1 , IDS
    1 AS1(K)= C1*AL1(K)+ C2*AS1(K)$ IF(IDF.EQ.IDS) GO TO 7
      CALL BUFOUT(FLD,AS1,IDS)$ IDP1= IDS+1$ DO 3 K = IDP1,IDF
    3 AS1(K-IDS)= AL1(K)$ CALL BUFFIN(FL2,AL1,IDF)$ DO 5 K = IDP1 , IDF
    5 AL1(K)= C2*AL1(K)+ C1*AS1(K-IDS)$ CALL BUFFIN(FLD,AL1,IDS)
      CALL BUFOUT(FLT,AL1,IDF)$ RETURN
    7 CALL BUFOUT(FLT,AS1,IDS)$ RETURN
      END
      SUBROUTINE REORDER(VAL,NP)
*
*     TO SUPPLY A NEW SET OF NP(MUD) .
      COMMON/DUMPN/TA(100),ITAG(100),TAP(100)
      DIMENSION VAL(1),NP(1)
      COMMON/BASIS/ ICODE(16),ISTA(16),IPAR(16),INPO(16),INUC(16),
     1 ILC(16),NOLV,NB(2),MUD,IDF,IDSC,IDSJ,JAR(14,2),IDAC(14,14),JMPO,
     2 NBT,HBOM,OSPA2
      NPT=0$ ILA=1$ DO 3 K = 1 , MUD$NPT=NPT+NP(K)
      ND= NB(K)$ DO 3 L = 1 , ND$ TA(ILA)= VAL(ILA)$ITAG(ILA)= K
    3 ILA= ILA+1$ CALL SORTAG(TA,1,NBT,ITAG)$ IF(NPT.EQ.0) RETURN
      NP(1)=NP(2)=0$ DO 7 L = 1 , NPT
    7 NP(ITAG(L))= NP(ITAG(L))+1$ RETURN
      END
      SUBROUTINE ROCONV(RO,ROJ,IDIR)
*
*     WHEN IDIR.GT.0 ,
*     TO CONVERT FROM RO IN (JC,MC,/RO/JA,MA) BASIS TO ROJ IN
*     (JC,JA,J,M) BASIS, NOTE JA VARIES MOST RAPIDLY
*     WHEN IDIR .LE.0 DO THE INVERSE TRANSFORMATION .
      COMMON/BASIS/ ICODE(16),ISTA(16),IPAR(16),INPO(16),INUC(16),
     1 ILC(16),NOLV,NB(2),MUD,IDF,IDSC,IDSJ,JAR(14,2),IDAC(14,14),JMPO,
     2 NBT,HBOM,OSPA2
      DIMENSION RO(1), ROJ(1)
      IC= 1$ CALL IDACSET(0)$ DO 1 K = 1 , 2$ IF(IDIR.GT.0) GO TO 3
      CALL ROCONIP(RO(IC),ROJ,K-1,NB(K))$ GO TO 1
    3 CALL ROCONVP(RO(IC),ROJ,K-1,NB(K))
    1 IC= IC+ NB(K)**2$ RETURN
      END
      SUBROUTINE IDACSET(IPR)
*
```

```
*       TO GENERATE THE BASE ADDRESSES OF ROJ(J,M,PI) IN IDAC(M,J),
*       IPI= 1 IF PI= -, IPI= 2 IF PI= + .
*       JAR(J,3-IPIT) IS ALSO CONSTRUCTED = DIM OF A J , IPIT SUBSPACE .
        COMMON/BASIS/ ICODE(16),ISTA(16),IPAR(16),INPO(16),INUC(16),
       1 ILC(16),NOLV,NB(2),MUD,IDF,IDSC,IDSJ,JAR(14,2),IDAC(14,14),JMPO,
       2 NBT,HBOM,OSPA2
        DO 21 K = 1 , 28
 21     JAR(K,1)= 0.$ JM= 1$ DO 1 K1= 1 , NOLV$ J1PH= ISTA(K1)
        IPI= IPAR(K1)$ JM= MAX0(JM,J1PH)$ DO 1 K2 = 1 , NOLV
        J2PH= ISTA(K2)$ IPI= 1+ MOD(IPI+IPAR(K2),2)
        JMIN= IABS(J1PH-J2PH)+1$ JMAX= J1PH+J2PH$ DO 1 JL= JMIN, JMAX
 1      JAR(JL,IPI)= JAR(JL,IPI)+1$ JMPO= JM+JM$ ILAB= 1$DO 7 K = 1 , 196
 7      IDAC(K,1)= 0.$ DO 11 JPO= 1 , 14
        DO 11 MPO= 1 , JPO$ IDAC(MPO,JPO)= ILAB
 11     ILAB= ILAB+ 2*JAR(JPO,1)$  IDSJ= ILAB-1
        IF(IPR.LT.3) RETURN$ PRINT 101,(ISTA(I),IPAR(I),I=1,NOLV)
        PRINT 103, JAR,IDAC$ RETURN
 101    FORMAT(//* ISTA AND IPAREXP, PAR=(-1)**IPAREXP*/(1X16(I6,I2)))
 103    FORMAT(//* DIM. FOR EACH J*//1X14I6/1X14I6//* BASE ADDRESSES FOR R
       10J*//(1X14I6))
        END
        SUBROUTINE ROCONVP(RO,ROJ,IPAC,NB)
*
*       TO CONVERT RO= (JC,MC/RO/JA,MA) OF PARITY IPAC TO
*       ROJ= RO(JC,JA/J,M)
        COMMON/BASIS/ ICODE(16),ISTA(16),IPAR(16),INPO(16),INUC(16),
       1 ILC(16),NOLV,DD(2),MUD,IDF,IDSC,IDSJ,JAR(14,2),IDAC(14,14),JMPO,
       2 NBT,HBOM,OSPA2
        COMMON/HISCLE/N00,N1,N21,N31,N12,N2,N4,N13,N3,N5,CLEB
        DIMENSION RO(NB,NB),ROJ(1)$ INTEGER TJPC,TJPA
        IRAC= 0$ DO 3 KC= 1 , NOLV$ JPHC= ISTA(KC)
        IF(MOD(IPAC+IPAR(KC),2).NE.0) GO TO 3$ TJPC= SHIFT(JPHC,1)-1
        IRAA= 0$ DO 5 KA = 1 , NOLV$ JPHA= ISTA(KA)
        IF(MOD(IPAC+IPAR(KA),2).NE.0)GOTO5$JACPO=JPHC+JPHA$TJPA=2*JPHA-1
        JAMCPO= IABS(JPHC-JPHA)+1$ DO 7 JPO= JAMCPO, JACPO
        N1S= JPHA+JPHC-JPO$ DO 7 MPO= 1 , JPO  $ MINT= JPHA-JPHC-MPO+1
        NAMIN= MAX0(0,MINT)+1$ NAMAX= MIN0(TJPA,JACPO-MPO)+1
        ROR=ROI=0.$ IPHAS= 1-SHIFT(MOD(1+NAMIN+IABS(MINT),2),1)
        DO 11 NAL= NAMIN , NAMAX$ NCL= NAL- MINT$ N1= N1S$ N2= NAL-1
        N4= TJPA-N2$ N5= NCL-1$ N3= TJPC-N5$ CALL HISCLEB$ IC= IRAC+NCL
        IA= IRAA+ NAL$ CLET= CLEB*IPHAS$ IF(IC-IA) 17,15,13
 13     ROR=ROR+CLET*RO(IC,IA)$ ROI=ROI-CLET*RO(IA,IC)$ GO TO 11
 15     ROR= ROR+ CLET*RO(IC,IC)$ GO TO 11
 17     ROR= ROR+CLET*RO(IA,IC)$ ROI= ROI+ CLET*RO(IC,IA)
 11     IPHAS= -IPHAS
        ILAB= IDAC(MPO,JPO)$ ROJ(ILAB)= ROR$ ROJ(ILAB+1)= ROI
 7      IDAC(MPO,JPO)= ILAB+2$ IRAA= IRAA+ TJPA+1
 5      CONTINUE $ IRAC= IRAC+ TJPC+1
 3      CONTINUE$ RETURN
        END
        SUBROUTINE ROCONTP(RO,ROJ,IPAC,NB)
*
*       TO DO THE INVERSE OF ROCONVP AND RETURN TO (JC,MC/RO/JA,MA) BASIS
        COMMON/BASIS/ ICODE(16),ISTA(16),IPAR(16),INPO(16),INUC(16),
       1 ILC(16),NOLV,DD(2),MUD,IDF,IDSC,IDSJ,JAR(14,2),IDAC(14,14),JMPO,
       2 NBT,HBOM,OSPA2
        COMMON/HISCLE/N00,N1,N21,N31,N12,N2,N4,N13,N3,N5,CLEB
        DIMENSION RO(NB,NB),ROJ(1),RJT(2,14)$ INTEGER TJPC,TJPA
        IRAC= 0$ DO 3 KC= 1 , NOLV$ JPHC= ISTA(KC)
        IF(MOD(IPAC+IPAR(KC),2).NE.0) GO TO 3$ TJPC= SHIFT(JPHC,1)-1
        IRAA= 0$ DO 5 KA = 1 , NOLV$ JPHA= ISTA(KA)
        IF(MOD(IPAC+IPAR(KA),2).NE.0)GOTO5$JACPO=JPHC+JPHA$TJPA=2*JPHA-1
```

```
      JAMCPO= IABS(JPHC-JPHA)+1$ DO 7 MPO= 1 , JACPO
      MINT= JPHA-JPHC-MPO+1$ JPOMIN= MAX0(MPO,JAMCPO)
      DO 8 JPO= JPOMIN , JACPO$ ILAB= IDAC(MPO,JPO)
      RJT(1,JPO)= ROJ(ILAB)$ RJT(2,JPO)= ROJ(ILAB+1)
*
*     TEST TO MAKE SURE THAT ROJ IS A HERMITIAM OPERATOR .
      IF(KC.EQ.KA.AND.MPO.EQ.1.AND.RJT(2,JPO).GT.1.E-12) CALL ERREXIT
     1 (JPO,KC,RJT,2HFL,28)
 8    IDAC(MPO,JPO)= ILAB+2$ NAMIN= MAX0(0,MINT)+1
      NAMAX=MIN0(TJPA,JACPO-MPO)+1 $ DO 7 NAL= NAMIN , NAMAX
      NCL= NAL-MINT$ ROTR=ROTI=0.$ IPHAS= 1-2*MOD(NCL-1,2)
      DO 11 JPO= JPOMIN, JACPO$ N1= JACPO-JPO$ N2= NAL-1$ N4= TJPA-N2
      N5= NCL-1$ N3= TJPC-N5$ CALL HISCLEB
      ROTR= ROTR+ CLEB*RJT(1,JPO)$ ROTI= ROTI+CLEB*RJT(2,JPO)
 11   CONTINUE$ IC= IRAC+NCL$ IA= IRAA+NAL$ IF(IC-IA)17,15,13
 13   RO(IC,IA)= ROTR*IPHAS$ RO(IA,IC)= -ROTI*IPHAS$ GO TO 7
 15   RO(IC,IC)= ROTR*IPHAS$ GO TO 7
 17   RO(IA,IC)= ROTR*IPHAS$ RO(IC,IA)= ROTI*IPHAS
 7    CONTINUE$ IRAA= IRAA+ TJPA+1
 5    CONTINUE$ IRAC= IRAC+ TJPC+1
 3    CONTINUE $ RETURN
      END
      SUBROUTINE HISCLEB
*
*     TO DO A HIGH SPEED COMPUTATION OF CLEBSCH(J1,M1,J2,M2/J3) .
*     USING THE FORMULA OF SCHWINGER(REF. QUANTUM THEORY OF ANG. MOM.,
*     BY BIEDENHARN,P246), AND THE REGGE SYMMETRIES(REF.IBID,P.296) .
*     PAR. ARE IN /HISCLE/N1=J1+J2-J3,N2=J1-M1,N3=J2-M2,N4=J1+M1,N5=J2+
*     M2. NO TESTING IS DONE, ANSWER IS IN CLEB IF ALL NIJ ARE NON-NEG.
*     INTEGERS . ......WARNING.....ALL 5 NIS MAY BE CHANGED .
*     COMMON/HISCLE/N00,N1,N21,N31,N12,N2,N4,N13,N3,N5,CLEB
      COMMON /HISCLE/NR(1),N11,N21,N31,N12,N22,N32,N13,N23,N33,CLEB
      COMMON/CLEBDAT/ALOGF(30),SQRF(30)
*
*     PREPARE THE REGGE Q-NOS IN NIJ .
      N21= N32+N33-N11$ N31= N22+N23-N11$ JT= N21+N22+N23$ IFSE=0
      N12= N23+N33-N11$ N13= N22+N32-N11$ JJP= JT-N11+1
*
*     FIND NZE THE SMALLEST NIJ AND BRING IT TO N11 POSITION .
      NZE= MIN0(N11,N12,N13,N21,N22,N23,N31,N32,N33)
      IF(NZE.EQ.N11) GO TO 7$ IF(NZE.EQ.N31) GO TO 43
      IF(NZE.EQ.N12) GO TO 44$ IF(NZE.EQ.N22) GO TO 45
      IF(NZE.EQ.N32) GO TO 46$ IF(NZE.EQ.N13) GO TO 47
      IF(NZE.EQ.N23) GO TO 48$ IF(NZE.EQ.N33) GO TO 49
 42   IFSE= N33$ NW= N21$ N21= N11$ N11= NW$ NU= N12$ N12= N22$ N22= NU
      NV= N13$ N13= N23$ N23= NV$ GO TO 7
 43   IFSE= N22$ NW= N31$ N31= N11$ N11= NW$ NU= N12$ N12= N32$ N32= NU
      NV= N13$ N13= N33$ N33= NV$ GO TO 7
 44   IFSE= N33$ NW= N12$ N12= N11$ N11= NW$ NU= N21$ N21= N22$ N22= NU
      NV= N31$ N31= N32$ N32= NV$ GO TO 7
 45   NW= N22$ N22= N11$ N11= NW$ NT= N21$ N21= N12$ N12= NT
      NU= N13$ N13= N23$ N23= NU$ NV= N31$ N31= N32$ N32= NV$ GO TO 7
 46   IFSE= N11+N32$ NW= N32$ N32= N11$ N11= NW$NT= N31$N31=N12$ N12= NT
      NU= N13$ N13= N33$ N33= NU$ NV= N21$ N21= N22$ N22= NV$ GO TO 7
 47   IFSE= N22$ NW= N13$ N13= N11$ N11= NW$ NT= N21$ N21= N23$ N23= NT
      NU= N31$ N31= N33$ N33= NU$ GO TO 7
 48   IFSE= N11+N23$ NW= N23$ N23= N11$ N11= NW$NU=N31$N31= N33$ N33= NU
      NT= N21$ N21= N13$ N13= NT$ NV= N12$ N12= N22$ N22= NV$ GO TO 7
 49   NW= N33$ N33= N11$ N11= NW$ NT= N31$ N31= N13$ N13= NT
      NU= N21$ N21= N23$ N23= NU$ NV= N12$ N12= N32$ N32= NV
*
*     CONSTRUCT DELTA*NORMALISATION AND DO THE SUM .
```

```
 7     SUM= .5*(-ALOGF(N11+1)+ALOGF(N12+1)+ALOGF(N13+1)+ALOGF(N21+1)-
      1 ALOGF(N22+1)+ALOGF(N23+1)+ALOGF(N31+1)+ALOGF(N32+1)-ALOGF(N33+1)
      2-ALOGF(JT+2))-ALOGF(N32-N11+1)-ALOGF(N23-N11+1)
       SUP= 1.$ IF(N11.LT.1) GO TO 13$ DO 15 K = 1 , N11
 15    SUP=1.-SUP*K*(N22-N11+K)*(N33-N11+K)/((N11+1-K)*(N23+1-K)*(N32+1-K
      1 ))
 13    CLEB= SUP*EXP(SUM)*(1-SHIFT(MOD(IFSE,2),1))*SQRF(JJP)$ RETURN
       END
       SUBROUTINE BUFOUT(NAME,DUA,LNGT)
*
*      TO USE MASS STORAGE FACILITIES
*      UP TO 100 RECORDS WITH ALPHA NAMES .
       COMMON/BUFFTN/INDX(201),OPEN
       LOGICAL OPEN
       DATA OPEN/.FALSE./, INDL/201/
       IF(OPEN) GO TO 1$ OPEN= .TRUE.
       CALL OPENMS(9,INDX,INDL,1)
 1      CALL WRITMS(9,DUA,LNGT,NAME)
       RETURN
       END
       SUBROUTINE DENMAT(RO,BR,BI,NO)
*
*      TO CONSTRUCT RO FROM BR,BI, ALL IN CARTESIAN BASIS .
       COMMON/BASIS/ ICODE(16),ISTA(16),IPAR(16),INPO(16),INUC(16),
      1 ILC(16),NOLV,NB(2),MUD,IDF,IDSC,IDSJ,JAR(14,2),IDAC(14,14),JMPO,
      2 NBT,HBOM,OSPA2
       DIMENSION RO(1),BR(1),BI(1),NO(2)
       M= 1$ DO 1 K = 1 , MUD$ ND= NB(K)
       CALL DENMATP(RO(M),BR(M),BI(M),NO(K),ND,ND,ND)
 1     M= M+ ND**2$ RETURN
       END
       SUBROUTINE DENMATP(RO,BR,BI,NPA,NBA,IB,IR)
*
*      TO CONSTRUCT RO(I,J)= SUM(K OCC.)B(I,K)*B(J,K)**CC .
       DIMENSION RO(IR,IR),BR(IB,IB),BI(IB,IB)
       NP= NPA$ NB= NBA$ DO 1 I = 1 , NB$ DO 1 J = 1 , I$ T=U=0.
       IF(NP.LT.1) GO TO 41$ DO 5 K = 1 , NP
       T= T+ BR(I,K)*BR(J,K)+ BI(I,K)*BI(J,K)
 5     U= U-BR(I,K)*BI(J,K)+BI(I,K)*BR(J,K)
 41    RO(I,J)= T$ IF(I.EQ.J) GO TO 1$ RO(J,I)= -U
 1     CONTINUE$ RETURN
       END
       SUBROUTINE BUFFIN(NAME,DUA,LNGT)
       CALL READMS(9,DUA,LNGT,NAME)
       RETURN
       END
       SUBROUTINE SORTAG(A,II,JJ,TAG)
C  SORTS ARRAY A INTO INCREASING ORDER, FROM A(II) TO A(JJ)
C  ARRAY TAG IS PERMUTED THE SAME AS ARRAY A
C  ORDERING IS BY INTEGER SUBTRACTION, THUS FLOATING POINT
C    NUMBERS MUST BE IN NORMALIZED FORM.
C  ARRAYS IU(K) AND IL(K) PERMIT SORTING UP TO I**(K+1)-1 ELEMENTS
       DIMENSION A(1),IU(16),IL(16),TAG(1)
       INTEGER A,T,TT
       M= 1$ I= II$ J= JJ
    5  IF(I .GE. J) GO TO 70
 10    K= I$ IJ= (J+ I)/2$ T= A(IJ)$ IF(A(I).LE.T) GO TO 20
       A(IJ)= A(I)$ A(I)= T$ T= A(IJ)$ TG= TAG(IJ)$ TAG(IJ)= TAG(I)
       TAG(I)= TG
 20    L= J$ IF(A(J).GE.T) GO TO 40 $ A(IJ)= A(J)$ A(J)= T$ T= A(IJ)
       TG= TAG(IJ)$ TAG(IJ)= TAG(J)$ TAG(J)= TG$ IF(A(I).LE.T) GOTO 40
       A(IJ)= A(I)$ A(I)= T$ T= A(IJ)$ TG= TAG(IJ)$ TAG(IJ)= TAG(I)
```

```
      TAG(I)= TG$ GO TO 40
 30   A(L)= A(K)$ A(K)= TT$ TG= TAG(L)$ TAG(L)= TAG(K)$ TAG(K)= TG
 40   L= L- 1$ IF(A(L).GT.T) GO TO 40 $ TT= A(L)
 50   K= K+ 1$ IF(A(K).LT.T) GO TO 50$ IF(K.LE.L) GO TO 30
      IF(L-I.LE.J-K) GO TO 60$ IL(M)= I$ IU(M)= L$ I= K$ M= M+ 1
      GO TO 80
 60   IL(M)= K$ IU(M)= J$ J= L$ M= M+ 1$ GO TO 80
 70   M= M- 1$ IF(M.EQ.0) RETURN$ I= IL(M)$ J= IU(M)
 80   IF(J-I.GE.11) GO TO 10$ IF(I.EQ.II) GO TO 5$ I= I- 1
 90   I= I+ 1$ IF(I.EQ.J) GO TO 70$ T= A(I+1)$ IF(A(I).LE.T) GO TO 90
      TG= TAG(I+1)$ K= I
 100  A(K+1)= A(K)$ TAG(K+1)= TAG(K)$ K= K- 1$ IF(T.LT.A(K)) GO TO 100
      A(K+1)= T$ TAG(K+1)= TG$ GO TO 90
      END
          IDENT FTNUTIL
         LIST
  USE START.
TRACE. VFD 42/7LFTNUTIL,18/FTNUTIL
TEMPA0. BSS 1B
ENTRY. BSS 0B
FTNUTIL  BSS 1B
  Sx6 A0
  SA0 A1
  SA6 TEMPA0.
FTNNOP. DATA 46000460004600046000B
  USE CODE.
  USE DATA.
NAM          VFD  60/7LCALINGP
LIN          DATA     0000000000000007777B
S          BSS 1B
T          BSS 1B
PARNO        DATA     0000000000000000001B
ADDR         DATA     0000000000000777777B
DFL          DATA     0000000000000000777B
X        BSS 3H
FL.       DIS     1,FL
TRCEBCK EQU ENTRY.
  ENTRY   TRCEBCK
NOPS. DATA 46000460004600046000B
ENTR. MACRO NAME
  LOCAL X,Z,T
  EQ T
NAME BSS 1
  ENTRY NAME
  SA2 X
  Bx6 X2
  SA6 FTNNOP.
  EQ ENTRY.+1
X EQ Z
Z SA1 NOPS.
  SA2 NAME
  Bx6 X1
  Lx7 B0,X2
  SA6 FTNNOP.
  SA7 ENTRY.
T BSS 0
  ENDM
.201       VFD  60/6C 201
  DIS   5.(/*  TRCEBCK CALLED BY *A7.* AT LINE *I4)
.202       VFD  60/6C 202
  DIS   6.(//*  TRCEDMP CALLED BY *A7.* AT LINE *I4,16X*REAL*19X*DECIN
  DIS   2,T*20X*OCTAL*)
```

```
.203      VFD  60/6C 203
  DIS   6,(/*  TIMEPRT CALLED BY *A7,* AT LINE *I4,*, CP TIME=*F9.3,*
  DIS   3,SEC, SYSTEM TIME *A10)
.204      VFD  60/6C 204
  DIS   6,(/*  ERREXIT CALLED BY *A7,* AT LINE *I4,16X*REAL*19X*DECINT
  DIS   2,*20X*OCTAL*)
.211      VFD  60/6C 211
  DIS   4,(10X*CALLED BY *A7,* AT LINE *I4)
.301      VFD  60/6C 301
  DIS   6,(/*  PARM NO. *12,* AT OCTALL *U6,*, DFL= *I3,6X1PE22.14,5XI
  DIS   1.16,8X020)
.302      VFD  60/6C 302
  DIS   3,(46X1PE22.14,5XI16,8X020)
.401      VFD  60/6C 401
  DIS   4,(//*  SYSTEM COMM. AREA OCTAL DUMP*/)
.403      VFD  60/6C 403
  DIS   6,(//*  WORKING AREA DUMP. 4000B AROUND ERREXIT CALL*/)
IEN       BSS 1B
  EXT     OUTPUT=
  EXT     OPUTCI.
  EXT     OUTPTC.
  EXT     SECOND$
  EXT     TIME$
  EXT     PDUMP$
IS        BSS 1B
IF        BSS 1B
IADR      BSS     1B
CALLA     BSS     1B
  EXT     EXIT$
CON. BSS 0B
  DATA 0000000000000000000B
  DATA 0000000000000000100B
  DATA 0000000000000000004B
  USE CODE.
* SX6 1B
  SA6 IEN
  EQ .100
  ENTR. TRCEDMP
* SX6 2B
  SA6 IEN
  EQ .100
  ENTR. TIMEPRT
* SX6 3B
  SA6 IEN
  EQ .100
  ENTR. ERREXIT
* SX6 4B
  SA6 IEN
.100 BSS 0B
         SA5     ENTRY.
         AX5     30                      X5=RET.ADDR.
         BX6     X5              SAVE
         SA6     CALLA                CALL ADDRESS
         SA4     A5-1B                X4=-VFD12/LIN,18/TRACE.
         SA3     X4                   X3=VFD42/7LCALPNAM,18/CALPNAM
         BX6     X3
         SA6     NAM
         AX4     18
         MX7     48
         BX6     -X7*X4
         SA6     LIN
         MX7     42
```

```
          BX6     -X7*X3
          SA6     IADR                    STORE PREVIOUS ENTRY
  SA4 IEN
  SX0 X4-4B
  SB1 X4-2B
  PL X0,.104
  LT B1,B0,.101
  JP B1+=GL1
=GL1  BSS 0B
  EQ .102
  EQ .103
.101 BSS 0B
+ SB2 OUTPUT=
  SB3 .201
+ RJ OPUTCI.
- VFD  12/14B,18/TRACE.
+ SB1 NAM
          SB2     2B
  RJ OUTPTC.
+ SB1 -1B
  RJ OUTPTC.
  EQ .200
.102 BSS 0B
+ SB2 OUTPUT=
  SB3 .202
+ RJ OPUTCI.
- VFD  12/17B,18/TRACE.
+ SB1 NAM
          SB2     2B              DO  BOTH MAM AND LIN
  RJ OUTPTC.
+ SB1 -1B
  RJ OUTPTC.
  EQ .300
.103 BSS 0B
+ SA1 [AP]
+ RJ SECOND$
- VFD  12/23B,18/TRACE.
+ SA5 S
  SA4 A0
  BX7 X5
  SA1 [AP2
  SA7 X4
  NO
+ RJ TIME$
- VFD  12/23B,18/TRACE.
+ SB2 OUTPUT=
  SB3 .203
+ RJ OPUTCI.
- VFD  12/24B,18/TRACE.
+ SB1 NAM
          SB2     4B              PRINT NAM,LIN,S,T
  RJ OUTPTC.
+ SB1 -1B
  RJ OUTPTC.
  EQ .99
.104 BSS 0B
+ SB2 OUTPUT=
  SB3 .204
+ RJ OPUTCI.
- VFD  12/30B,18/TRACE.
+ SB1 NAM
          SB2     2B              PRINT , NAMJ,LIN
```

```
  RJ OUTPTC.
* SB1 -1B
  RJ OUTPTC.
.200      BSS     0B          FULL TRACE ENTRY .
          SA4     IADR
          ZR      X4,.299                 JMP,TRACE ENDS.
          SA5       X4
          AX5     30                      X5=-ET.ADD-.
          SA4     X5-1B                   X4=-VFD12/L+N,18/T-ACE.
          SA3     X4                      X3=VFD42/7LCALPNAM,18/CALPNAM
          BX6     X3
          SA6     NAM
          AX4     18
          MX7     48
          BX6     -X7*X4
          SA6     LIN
          MX7     42
          BX6     -X7*X3
          SA6     IADR                    STORE PREVIOUS ENTRY
  SB2 OUTPUT=
          SB3     .211
* RJ OPUTCI.
- VFD  12/7B,18/TRACE.
*         SB1     NAM
          SB2     2B
  RJ OUTPTC.
* SB1 -1B
  RJ OUTPTC.
          EQ      .200
.299      BSS     0B
  SA5 IEN
  SX0 4B
  IX7 X5-X0
  NO
  NZ X7,.99
.300 BSS 0B
          SX6     1B
          SA6     PARNO
          SA5     A0                      BRING FIRST PAR. AD.
)AA       BSS     0B
          SA4     X5                      FETCH FIRST PAR.
          SB2     OUTPUT=
          SB3     .301
          BX7     X4
          SA7     X
          SA7     X+1B
          SA7     X+2B
          BX6     X5
          SA6     ADDR
          SA5     A0+1B
          SA4     X5
          SA3     FL.
          IX0     X3-X4
          ZR      X0,.21
          SX6     1B
          SA6     OFL
          EQ      .23
.21       BSS     0B
          SA0     A0+2B
          SA5     A0
          ZR      X5,.25
          SA4       X5
```

```
          SX0       X4-1B
          NG        X0,.25                  IF DFL LT 1 RETURN
          SX7       X4-777B
          NG        X7,.27
          SX4       777B                    DFL LE 777B
.27       BSS       0B
          BX6       X4
          SA6       DFL
.23       BSS       0B
+         RJ        OPUTCI.
-         VFD       12/12B,18/TRACE.
+         SB1       PARNO
          SB2       6B
          RJ        OUTPTC.
+         SB1       -1B
          RJ        OUTPTC.
          SA5       DFL
          SX6       2B
          IX0       X5-X6
          BX7       X0
          SA7       DFL
          NG        X0,.29                  IF DFL LT 2 GO FOR NEXT PAR.
+ SB2 OUTPUTE
          SB3       .302
+ RJ OPUTCI.
- VFD  12/6B,18/TRACE.
)AB       BSS       0B
          SA5       ADDR
          SX4       X5+1B
          SA3       X4
          BX7       X4
          SA7       ADDR
          BX6       X3
          SA6       X
          SA6       X+1B
          SA6       X+2B
+         SB1       X
          SB2       3B
          RJ        OUTPTC.
          SA5       DFL
          SX7       X5-1B
          SA7       A5
          PL        X7,)AB
+         SB1       -1B
          RJ        OUTPTC.
.29       BSS       0B
          SA0       A0+1B                   LOOK AT NEXT PAR.
          SA5       A0                      X5= AD. OF NEXT PAR.
          ZR        X5,.25                  IF NO MORE  END IT ALL.
          SA4       PARNO
          SX7       X4+1B
          SX0       X7-25B                  PARNO= 1 , 20
          SA7       A4
          NG        X0,)AA
.25       BSS       0B
  SA5 IEN
  SX0 4B
  IX7 X5-X0
  NO
  NZ X7,.99
  SB2 OUTPUTE
  SB3 .401
```

```
+ RJ OPUTCI.
- VFD  12/53B,18/TRACE.
+ SB1 -1B
  RJ OUTPTC.
+ SA1 [AP3
+ RJ PDUMP$
- VFD  12/55B,18/TRACE.
+ SB2 OUTPUT=
  SB3 .403
+ RJ OPUTCI.
- VFD  12/56B,18/TRACE.
+ SB1 -1B
  RJ OUTPTC.
+ SA1 [AP4
          SA5     CALLA
          SX7     X5-2000B
          PL      X7,.98
          SX7     0B
.98       BSS     0B
          SX6     X5+1777B
  SA7 IS
  SA6 IF
+         RJ      PDUMP$
- VFD  12/60B,18/TRACE.
+         RJ      EXIT$
-         VFD     12/61B,18/TRACE.
.99 BSS 0B
+ SA5 TEMPA0.
  SA0 X5
  NO
  EQ ENTRY.
[AP1      BSS 0B
  VFD  60/S
  DATA 0B
[AP2      BSS 0B
  VFD  60/T
  DATA 0B
[AP3      BSS 0B
          VFD     60/CON.
  VFD  60/CON.+1B
  VFD  60/CON.+2B
  DATA 0B
[AP4      BSS 0B
  VFD  60/IS
  VFD  60/IF
  VFD  60/CON.+2B
  DATA 0B
          END
      OVERLAY(READIN,3,0)
      PROGRAM READIN
*
*     TO READ IN THE TWO-BODY MATRIX ELEMENTS .
*     AL1 WIL BE OVERWRITTEN TO LENGTH 15000 .
      COMMON/BASIS/ ICODE(16),ISTA(16),IPAR(16),INPO(16),INUC(16),
     1 ILC(16),NOLV,NB(2),MUD,IDF,IDSC,IDSJ,JAR(14,2),IDAC(14,14),JMPO,
     2 NBT,HBOM,OSPA2
      COMMON/CONTRSN/ICON(2,40),RNU(5,40),I3LK,HBOW,TNAME,HBWA(30)
      COMMON/OVERCON/ FIRST,ISKP
      COMMON AS1(8000),AL1(1)
      DIMENSION NA(2),IPOS(8),COA(8),BTA(2)
      DATA BTA/4LRONI,4LROPI/
      CALL TIMEPRT(YT)$ IF(ISKP.EQ.0) GO TO 1
```

```
      CALL REDFAT1(TNAME,IDG,IBLK,AL1)$ REWIND 1
*
*     FLD CONTAINS THE DIMENSIONLESS COULOMB ANTIS. MATRIX ELEMENTS .
      E2B= SQRT(.0241145*HBOM)*1.44
      CALL ADLONG(3LFNE,1.,3LFLD,E2B,3LFPR,AS1,AL1)
      CALL BUFFIN(3LFLD,AL1,IDF)$ DO 15 K = 1 , IDF
 15   AL1(K)= E2B*AL1(K)$ CALL BUFOUT(3LFCO,AL1,IDF)$ GO TO 99
 1    DO 9 KKIN= 1 , 2
      READ 103,CNAME,NA,KIND,CTNAME,IBLKC,CHBOM,CWHEN,CNOW
      PRINT 105,CNAME,NA,KIND,CTNAME,IBLKC,CHBOM,CWHEN,CNOW
 103  FORMAT(A10,2I4,2X2A10,I5,F10.2,2A10)
 105  FORMAT(1XA10,2I4,2X2A10,I5,F10.2,2A10)
 101  FORMAT(8(I4,F6.4))
 102  FORMAT(1X8(I4,F6.4))
      IS= ICNT=0$ IDSCT= 2*IDSC$ DO 11 I = 1 , IDSCT
 11   AS1(I)= 0.$ DO 3 KC= 1 , 1340
      READ  101,(IPOS(J),COA(J),J=1,8)
      PRINT 102,(IPOS(J),COA(J),J=1,8)
      IREIM= 1$ DO 3 JC= 1 , 8$ IF(COA(JC).NE.0) GO TO 7
      IF(IPOS(JC).NE.ICNT) CALL ERREXIT(IPOS(JC),ICNT)$ IS= IDSC$ICNT=0
      IREIM= IREIM+1$ ICNT=0$ IF(IREIM.LE.2) 3,13
 7    IF(IPOS(JC).GT.IDSC) CALL ERREXIT(IDSC,IPOS(JC))
      AS1(IS+IPOS(JC))= COA(JC)$ ICNT= ICNT+1
 3    CONTINUE$ CALL ERREXIT(KKIN,CNAME)
 13   CALL DENMAT(AL1,AS1,AS1(IS+1),NA)$ CALL ROCONV(AL1,AS1,1)
      CALL BUFOUT(BTA(KKIN),AS1,IDSJ)
 9    CONTINUE
 99   CONTINUE
      END
      SUBROUTINE REDFAT1(FLNAME,IDG,IBLK,AL1)
*
*     FLD= FCS
*     READ FROM TAPE1 INTO DISC THE 7 ARRAYS .
*     NOLV= F(1)= NUMBER OF LEVELS .
*     ICODE(NOLV)= CODENUMBER ARRAY= F(2,...,NOLV+2) .
*     IDF= NUMBER OF F ELEMENTS READ .
      COMMON/BASIS/ ICODE(16),ISTA(16),IPAR(16),INPO(16),INUC(16),
     1 ILC(16),NOLV,NB(2),MUD,IDF,IDSC,IDSJ,JAR(14,2),IDAC(14,14),JMPO,
     2 NBT,HBOM,OSPA2
      DIMENSION TA(25),NAA(7),AL1(1)
      DATA NAA/3LFNE,3LFDN,3LFLD,3LRTS,3LRTD,3LPTS,3LPTD/
      REWIND 1
      IBLP= IBLK+1$ DO 11 IBL= 1 , IBLP.IBLK$IB=IBL-1
      IF(IB-1.LT.1) GOTO5$IBS= IB-1$ DO 3 ISK= 1 , IBS$ DO 3 K = 1 , 4
 3    READ(1)
 5    READ(1) TA $ TITLE= TA(1)$ IF(TITLE.NE.FLNAME) GO TO 101
      NOLV= TA(2)$ DO 7 K = 1 , NOLV
 7    ICODE(K)= TA(K+2)
      IDF= TA(21)$ IDG= TA(22)$ HBOM= TA(23)$ IF(IB.LT.1) GO TO 13
      READ(1) (AL1(I),I=1,IDG)$  CALL BUFOUT(5LGMATR,AL1,IDG)
 13   DO 1 L = 1 , 7$ IF((IB.LT.1.A.L.LT.3).OR.(IB.GT.0.A.L.GT.2))GOTO1
      READ(1)(AL1(I),I=1,IDF)$ CALL BUFOUT(NAA(L),AL1,IDF)
 1    CONTINUE
 11   CONTINUE
 105  RETURN
 101  PRINT 103
 103  FORMAT (/* F ARRAY NOT ON TAPE) */)
      GO TO 105
      END
      OVERLAY(ELMOSE,1,0)
      PROGRAM ELMOSE
*
```

```
*       ELMOSE PROVIDES THE FOLLOWING SERVICES FOR EVALIN.
*       IT SETS QNO(NBT), THE CHARACTER ARRAY, THE SINGLE PART. MES.
*       IT PROVIDES THE INITIAL GUESS FOT THE NUCLEAR WAVEFUNCTION TO BE
*       USED IN NUHART, IF ISKP.NE.0 .
        COMMON/BASIS/ ICODE(16),ISTA(16),IPAR(16),INPO(16),INUC(16),
       1 ILC(16),NOLV,NB(2),MUL,IDF,IDSC,IDSJ,JAR(14,2),IDAC(14,14),JMPO,
       2 NBT,HBOM,OSPA2
        COMMON/OVERCON/FIRST,ISKP
        COMMON/NUHARN/VALN(82),VALP(82),NP(2),NN(2),NAME,ITER,ITCR,ATNO
       1 ,NORBA(41),NORB,BET,GAMD,NPTOT,NNTOT
        COMMON/NAMEN/FPR,PTS,FPF,FNE,FNF,FDN,PTD,FDF,RONI,ROPI,SPE,SPEXF
       1 , RTS,RTD
        COMMON AS1(4000),AS3(4000)
        COMMON/PRINLIM/EPS,IPR,QNO(82)
        COMMON /SPINORB/ CHI(7),RD(7)/EXSYPN/EXF(6,6),SYP(21),NOMM(20,3),
       1 COMM(20)/CARTEMP/AS2(5726)
        LOGICAL FIRST
        EQUIVALENCE (KTMP,IDSJ)
        DATA CHI,RD/4*.07,3*.05,3*.0,.35,3*.45/
        CALL TIMEPRT(Y)$ CALL CARTOJ(0)$ IF(ISKP.EQ.0) GO TO 20
        PRINT 103,NAME$ CALL ELTRIAL(AS1,IPR)
 103    FORMAT(*1 TRIAL WAVE FUNCTION COMPUTATION FOR *A10)
        DO 11 K = 1 , IDSC
 11     AS2(K)= 0.$ CALL REORDER(VALP,NP)$ CALL DENMAT(AS3,AS1,AS2,NP)
        CALL ROCONV(AS3,AS2,1)$ CALL BUFOUT(ROPI,AS2,IDSJ)$ DO 13 K=1,IDSC
 13     AS2(K)= 0.$ CALL REORDER(VALP,NN)$ CALL DENMAT(AS3,AS1,AS2,NN)
        CALL ROCONV(AS3,AS2,1)$ CALL BUFOUT(RONI,AS2,IDSJ)
*
*       FIRST TIME AROUND COMPUTE THE 21 EXTERNAL FIELD OPS. XI*XJ,PI*PJ,
*       RI*PJ,I,J=1,2,3,IN DIMENSIONLESS FORM .
 20     IF(.NOT.FIRST)  GO TO 22$ DO 21 K1= 1 , 6$ DO 21 K2= 1 , K1
        CALL EXFI1H(AS3,K1,K2)
        IF(IPR.GT.4) PRINT 101,EXF(K1,K2),(AS3(I),I=1,IDSC)
        CALL ROCONV(AS3,AS1,1)$ CALL BUFOUT(EXF(K1,K2),AS1,KTMP)
 101    FORMAT(//* EXT FIELD NAME *A10//(1X10F12.5))
 21     CONTINUE
*
*       NOW PUT THE 21 OPERATORS IN THE SYMPLECTIC BASIS .
        ICO=1$ DO 201 IS= 1 , 9$ DO 203 K = 1 , KTMP
 203    AS1(K)= 0.
 207    CALL BUFFIN(EXF(NOMM(ICO,3),NOMM(ICO,2)+3),AS2,KTMP)
        CONS= COMM(ICO)$ DO 205 K = 1 , KTMP
 205    AS1(K)= AS1(K)+ AS2(K)*CONS
        ICO= ICO+1$ IF(NOMM(ICO,1).EQ.IS) GO TO 207
 201    CALL BUFOUT(SYP(IS),AS1,KTMP)
        ICO= 7$ DO 211 IS= 4 , 9$ DO 213 K = 1 , KTMP$ AS3(K)= 0
 213    AS1(K)= 0
 217    CALL BUFFIN(EXF(NOMM(ICO,3),NOMM(ICO,2)),AS2,KTMP)
        CONS= COMM(ICO)$ DO 215 K = 1 , KTMP
 215    AS1(K)= AS1(K)+ AS2(K)*CONS
        CALL BUFFIN(EXF(NOMM(ICO,3)+3,NOMM(ICO,2)+3),AS2,KTMP)
        DO 225 K = 1 , KTMP
 225    AS3(K)= AS3(K)+AS2(K.*CONS$ ICO=ICO+1$IF(NOMM(ICO,1).EQ.IS)GOTO217
*       SAVE THE KIN ENERGY .
        IF(IS.NE.9) GO TO 226$ RKEN= .5*HBOM/CONS$ DO 227 K = 1 , KTMP
 227    AS2(K)= RKEN*AS3(K)$ CALL BUFOUT(SPE,AS2,KTMP)
 226    DO 233 K = 1 , KTMP$ AS2(K)= .5*(AS3(K)+AS1(K))
 233    AS1(K)= .5*(AS3(K)-AS1(K))
        CALL BUFOUT(SYP(IS+6),AS2,KTMP)$CALL BUFOUT(SYP(IS+12),AS1,KTMP)
*       SYP(10,,,15)= .5*(PI*PJ+RI*RJ),SYP(16,,,21)= .5*(PI*PJ-RI*RJ)
 211    CONTINUE$ CALL CARSET(QNO)
 22     CONTINUE
```

```
      END
      SUBROUTINE ELTRIAL(BR,IPR)
*
*     BR(ND,ND)= REAL PART OF EIG. VECT MATRIX TO BE FILLED.
*     NB= RETURNED VALUE OF BASIS DIM.
*     NORBA(NORB)= ARRAY OF NILSSON ORBITS TO BE FILLED,
*     AT VALUES OF BET= BETA GAMD= GAMMA IN DEGREES .
*     CODA(NOLV)= CODENUMBER ARRAY TO DESCRIBE THE BASIS .
*     IPR= PRINT INDEX VARIES FROM -1 TO +6 .
      COMMON/BASIS/ ICODE(16),ISTA(16),IPAA(16),INPO(16),INUC(16),
     1 ILC(16),NOLV,NB(2),MUD,IDF,IDSC,IDSJ,JAR(14,2),IDAC(14,14),JMPO,
     2 NRT,HBOM,OSPA2
      COMMON/NUHARN/VALN(82),VALP(82),NP(2),NN(2),NAME,ITER,ITCR,ATNO
     1 ,NORBA(41),NORB,BET,GAMD,NPTOT,NNTOT
      DIMENSION BR(1),MLAM(2),ITE(2),IFL(2),IBSAD(7,7)
      COMMON/SPINORB/CHI(7),RD(7)
      COMMON /WAVN/ ARRAYS(4116),TAU(3),SR(3)
      COMMON /DISTORA/ FNNP(7,4,3) /DUMPN/C1(75,2),C2(75,2)
      DO 1 K = 1 , 7$ DO 1 L = 1 , 7
 1     IBSAD(K,L)=0$ IFL(1)=1$ IFL(2)= 1
      DO 3 K = 1 , NOLV$ JPH= ISTA(K)$ NPO= INPO(K)$ IPAR= 3-IPAA(K)
      IBSAD(NPO,JPH)= IFL(IPAR)$ IFL(IPAR)= IFL(IPAR)+ 2*JPH
 3    CONTINUE
      CALL  WAVSETE(BET,GAMD,CHI,RD,IPR)
      DO 45 K = 1 , 3
   45 CALL DISTOR(TAU(K),FNNP(1,1,K),7,4)
      KMA= IDSC$ DO 5 I = 1 , NRT
 5    VALP(I)= 100.$ DO 35 I = 1 , KMA
 35   BR(I)=0.$ MLAM(1)=0$ MLAM(2)= NB(1)**2$ ITE(1)=1$ ITE(2)=1+NB(1)
      ENLA= 1.$ DO 49 INP= 1 , NORB$ JNIL= NORBA(INP)
      DO 49 KTR= 1 , 2 $ IF(KTR.GT.1) GO TO 53
      CALL OBCFP(JNIL,C1,N1,75,1)$ GO TO 51
   53 CALL TIMEREV(C1,N1,75)$ CALL SPRNTC(C1,N1,C1(1,2))
   51 DO 55 N= 1 , N1$ IQN= C1(N,1)$ NSHELL= IQN/10000
      IPAR= 1+ MOD(NSHELL,2)
      NREM= IQN- 10000*NSHELL$ JPK= NREM/100$ JMK= NREM-100*JPK
      JPH= (JPK+JMK+1)/2$ IADR= IBSAD(NSHELL+1,JPH)
      IF(IADR.EQ.0) GO TO 55
      IALAM= IADR+ JMK+ MLAM(IPAR)
      BR(IALAM)= C1(N,2)
 55   CONTINUE $ MLAM(IPAR)= MLAM(IPAR)+ NB(IPAR)
      VALP(ITE(IPAR))= ENLA$ ENLA= ENLA+1.
      ITE(IPAR)= ITE(IPAR)+1
 49   CONTINUE$ IF(IPR.GT.5) PRINT 103,(BR(I),I=1,IDSC)$ RETURN
 103  FORMAT(//* THE TRIAL WF*//(1X10F12.5))
      END
      SUBROUTINE CARTOJ(IPR)
*
*     TO GENERATE THE ARRAYS STUCO(1260,2) AND CSS(300,2) WITH THEIR
*     DICTIONARY IQN(34,6), ICSD(49) .
*     STUCO CONTAINS THE COMPONENTS IN THE SPIN-CART. BASIS OF THE
*     SPHERICAL EIGENSTATES/N,J,K), WITH K-.5= EVEN, WITH THE
*     BRODY-MOSHINSKY PHASE . CSS CONTAINS THE COMPONENTS OF THE
*     SOLID SPHERICAL HARMONICS IN THE S-C OPERATOR BASIS.
*     THE METHOD USED IS BASICALLY THAT OF MOSHINSKY IN THE NOTES,
*     GROUP THEORY AND COLLECTIVE MOTION,1962, LATIN AM. SCHOOL OF PHYS.
      COMMON /CARJN/ STUCO(1260,2),IQN(84,6),CSS(300,2),ICSD(49)
      COMMON /CARTEMP/ CS(25,2,13),CU(15,2,13),CJA(30,2,49),IAD(7,14,14)
     1 ,NPA(13),SFAC(7) ,DSFAC(7),IJA(49),ITQN(49,2),C(50,2),CP(50,2)
      DATA PI/3.14 159 265 358 98/
      DSFAC(1)= SFAC(1)= 1.$ NX= 1$ DO 21 I = 1 , 6
      DSFAC(I+1)= DSFAC(I)*SQRT(FLOAT(2*I+1))
```

```
 21   SFAC(I+1)= SFAC(I)*SQRT(FLOAT(I))
*
*     FOR N = 0-6 SHELLS, L= 0-6
      ICL= 1$ DO 1 IL= 1 , 7$ L= IL-1$ RIL= L$ RLH= 1./SQRT(2.*RIL+1.)
*
*     FIRST CONSTRUCT YLL IN THE A,B,C BASIS AS (4*PI/(2*L+1))**.5*R**L*Y(L,L)
      N= 1$ C(1,1)= FLOAT(10000*L)
 503  C(1,2)= DSFAC(IL)/SFAC(IL)*RLH*(1-2*MOD(L,2))
 505  KMM= 2*L+1$ M= L$ DO 3 KM= 1 , KMM
      CALL POLTRA(CS(1,1,KM),NPA(KM),25,C,N,50)
      CALL TRIMUP(CS(1,1,KM),NPA(KM),25,1.E-8)
      ICM= NPA(KM)$ ITE= 10000*ICL+ ICM
      DO 511 IC= 1 , ICM$ DO 513 J = 1 , 2
 513  CSS(ICL,J)= CS(IC,J,KM)
 511  ICL= ICL+ 1$ IF(KM.EQ.KMM) GO TO 3
      CALL NOLMIN(CP,NP,50,L,M, C,N,50)
      DO 7 J = 1 , NP$ DO 7 I = 1 , 2
 7    C(J,I)= CP(J,I)$ N= NP
 3    CONTINUE
 5    IF(IPR.LT.2) GO TO 10$ PRINT 101, L
 101  FORMAT(///* SPHER. HARM. OF ORDER *I2,*, IN SPIN-CART. OP. BASIS*
     1 //)
*
*     NOW USE CS THE OPERATOR TO CONSTRUCT CU THE STATE VECTOR .
 10   DO 9 KM= 1 , KMM$ M= L+ 1- KM$ ISWI= MOD(IABS(M),2)+ 1
      JM= NPA(KM)$ DO 23 J = 1 , JM$ NTE= IFIX(CS(J,1,KM))
      N1= NTE/10000$ NTEP= NTE- 10000*N1$ N2= NTEP/100$ N3= NTEP-100*N2
      N12= MOD(N1*N2,2)$ GO TO (25,27) ISWI
 25   IPHAS= 1- 2*MOD(N2/2+N12,2) $ GO TO 29
 27   IPHAS= 1-2*MOD((N2+1)/2+N12,2)
 29   CNOR= SFAC(N1+1)*SFAC(N2+1)*SFAC(N3+1)*IPHAS$CU(J,1,KM)=CS(J,1,KM)
 23   CU(J,2,KM)= CS(J,2,KM)*CNOR$ IF(IPR.LT.2) GO TO 9
      PRINT 119, M
 119  FORMAT(4HOM=   I3)
      CALL SPRNTC(CS(1,1,KM),NPA(KM),CS(1,2,KM))
 9    CONTINUE
*
*     NEXT CONSTRUCT STATES OF GOOD J,K IN CJA , WITH K-.5= EVEN.
      DO 31 IPH= 1 , 2$ JPH= IL+ IPH- 2$ IF(JPH.LT.1) GO TO 31
      ISWI= IPH$ DO 33 IAK= 1 , JPH
      IAKM= MOD(IAK-1,2)$ RK= (FLOAT(IAK)-.5)*(1-2*IAKM)
      IKPH= IAK*(1-2*IAKM)+ IAKM $ CO1= SQRT(RIL+ .5+RK)*RLH
      CO2= SQRT(RIL+.5-RK)*RLH$ GO TO (35,37) ISWI
 35   A1= -CO2$ A2= + CO1$ GO TO 39
 37   A1= CO1$ A2= CO2
 39   KM1= IL- IKPH+ 1$ KM2= KM1-1$ N1= 0
      IF(KM1.LT.1.OR.KM1.GT.2*IL-1) GO TO 41$ N1= NPA(KM1)
 41   N2= 0$ IF(KM2.LT.1.OR.KM2.GT.2*IL-1) GO TO 43$ N2= NPA(KM2)
 43   CALL LINCOM(A1,CU(1,1,KM1),N1,15,A2,CU(1,1,KM2),N2,15,
     1 CJA(1,1,NX),IJA(NX),30)
      IN1= JPH+ IKPH$ IN2= JPH- IKPH+1$ IAD(IL,IN1,IN2)= NX
      ITQN(NX,1)= IN1- 1$ ITQN(NX,2)= IN2- 1$ NX= NX+ 1
 33   CONTINUE
 31   CONTINUE
 1    CONTINUE
*
*     START COPYING INTO MAIN ARRAY STUCO .
      ILAB= NY= 1$ DO 51 IN= 1 , 7 $ NSH= IN- 1$ ILU= NSH/2+ 1
      IREM= MOD(NSH,2)$ IL= IREM+ 1$ DO 53 ILT= 1 , ILU$ L= IL- 1
      NEX= (NSH- L)/2$ IN1= MAX0(IL,2)$ IN2= IN1- 1$ KMM= 2*IL- 1
      NAST= IAD(IL,IN1,IN2)$ DO 55 KM= 1 , KMM
      TFAC= SQRT(FLOAT(2*IL-1))/DSFAC(IL)$ BNOR= 1.
```

```
      IF(NEX.EQ.0) BNOR= TFAC$ INEM= IJA(NAST)$ DO 59 INE= 1 , INEM
      C(INE,1)= CJA(INE,1,NAST)
 59   C(INE,2)= CJA(INE,2,NAST)$ IF(NEX.EQ.0) GO TO 71
*
*     HERE PUT IN THE REQUIRED POWERS OF (ABS(ACROSS))**2 .
      DO 57 JNE= 1 , NEX $ A= 1.
      TFAC= -TFAC/SQRT(FLOAT((2*JNE+2*IL-1)*2*JNE))
      IF(JNE.EQ.NEX) A= TFAC$ CALL SCALN2(A,CP,NP,50,C,INEM,50)
      CALL TRIMUP(CP,NP,50,1.E-8)$ DO 61 INE= 1 , NP$ C(INE,1)=CP(INE,1)
 61   C(INE,2)= CP(INE,2)$ INEM= NP
 57   CONTINUE
 71   IQN(NY,5)= ILAB$ DO 62 INE= 1 , INEM$ STUCO(ILAB,1)= C(INE,1)
      STUCO(ILAB,2)= C(INE,2)*BNOR
 62   ILAB= ILAB+ 1$ IQN(NY,6)= ILAB- 1$ IQN(NY,1)= NY
      IQN(NY,2)= NSH$ IQN(NY,3)= ITQN(NAST,1)+ ITQN(NAST,2)
      IQN(NY,4)= ITQN(NAST,1)- ITQN(NAST,2)
      STUCO(ILAB,1)= 999900+ NY+ 1 $ STUCO(ILAB,2)= 0.$ ILAB= ILAB+ 1
      NAST= NAST+ 1$ NY= NY+ 1
 55   CONTINUE $ IL= IL+ 2
 53   CONTINUE
 51   CONTINUE
      ILABT= ILAB- 1$ IF(ILABT.GT.1260) CALL EXIT
*
*     ORDER STUCO FOR FUTURE USE IN SCALP .
      DO 81 I = 1 , 8$$ II= IQN(I,5)$ JJ= IQN(I,6)
 81   CALL SORTAG(STUCO(1,1),II,JJ,STUCO(1,2))$ IF(IPR.LT.1) RETURN
      PRINT 213
 213  FORMAT(//*1 XPANS. COEF. OF SPHER. STATES IN S-C. BASIS*///)
      CALL SPRNTC(STUCO(1,1),ILABT,STUCO(1,2))$ PRINT 217
 217  FORMAT(//*1QUANTUM NUMBER DICTIONARY */)
      ITPR= 1$ DO 69 K = 1 , 7$ ITPM= ITPR+11
      PRINT 219, ((IQN(I,J),I=ITPR,ITPM),J=1,6)
 219  FORMAT (5H I=   12(3XI4,3X)/5H NH=  12(3XI4,3X)/5H 2*J=
     1 12(3XI4,3X)/5H 2*K= 12( 3XI4,3X)/5H IAS= 12(3XI4,3X)/5H IAF=
     2 12(3XI4,3X)//)
 69   ITPR= ITPR+ 12$ RETURN
      END
      SUBROUTINE POLTRA(CP,NP,NDP,C,N,ND)
C
C     CP(N,1)= 10000*NP1+ 100*NP2+ NP3 .
C     CP(N,2)= COEFFICIENT .
C     THIS SUBR. DOES POLYNOMIAL MANIPULATIONS.
C     CP = CP+ C*(X-Y)**N1*(X+Y)**N2*Z**N3/2**((N1+N2)/2) .
C
      DIMENSION CP(NDP,2),C(ND,2)
      DATA SQA/.70710 67811 86548/
      NP= 0$ IF(N.EQ.0) RETURN$ DO 1 NC= 1 , N$ BN= C(NC,2)
      INT= IFIX(C(NC,1))$ N1= INT/10000$ INR= INT- 10000*N1$N2=INR/100
      N3= INR- 100*N2$ SQ= SQA**(N1+N2)*BN$ CT1= 1.$ N1F= N1+1
      DO 1 NR = 1 , N1F$ CT2= 1.$ N2F= N2+ 1$ DO 3 NS = 1 , N2F
      CST = 10000*(N1F-NR+N2F-NS )+ 100*(NR+NS-2)+ N3
      COE = SQ*CT1*CT2$ IF(NP.EQ.0) GO TO 7 $ DO 5 NPP = 1 , NP
      IF(CST.NE.CP(NPP,1)) GO TO 5$ CP(NPP,2)= CP(NPP,2)+ COE$GOTO9
    5 CONTINUE
    7 NP = NP+ 1$ IF(NP.GT.NDP) CALL EXIT$ CP(NP,1)= CST$ CP(NP,2)=COE
    9 CT2= CT2*(N2F-NS)/FLOAT(NS)
    3 CONTINUE
      CT1= -CT1*(N1F-NR)/FLOAT(NR)
    1 CONTINUE$ RETURN
      END
      SUBROUTINE TRIMUP (C,N,ND,EPS)
C     EPS= SMALLEST TERM TO BE RETAINED .
```

```
C      N WILL BE CHANGED .
C
       DIMENSION C(ND,2)
       IF (N.EQ. 0) RETURN$ NP= 0$ DO 1 NT= 1 , N
       IF (ABS(C(NT,2)).LT.EPS) GO TO 1$ NP= NP+ 1
       IF (NP.GT.ND) CALL EXIT$ C(NP,1)= C(NT,1)$ C(NP,2)= C(NT,2)
     1 CONTINUE$ N= NP$ RETURN
       END
       SUBROUTINE NORMIN(CP,NP,NDP,L,M,C,N,ND)
C
C      COMPUTE THE ACTION OF THE NORMALISED LOWERING OPERATOR.
C
       DIMENSION CP(NDP,2),C(ND,2),IN(2),TEC(2),SIG(2)
       DATA TEC/99.,-9999./, SIG/1.,-1./
       NP = 0$ IF(N.EQ.0) RETURN $ IF(M.LE.-L) GO TO 17
       CNOR= SQRT(2./FLOAT((L+M)*(L-M+1)))$ DO 1 NT= 1 , N
       INT = IFIX(C(NT,1))$ INTP= INT/100$ IN(1)= INT- 100*INTP
       IN(2)= INT/10000$ DO 1 J = 1 , 2$ IF(IN(J).LT.1) GO TO 1
       QNT = C(NT,1)+ TEC(J)$ CT= SIG(J)*IN(J)*C(NT,2)*CNOR
       IF(NP.EQ.0) GO TO 7$ DO 5 NPT= 1 , NP$ IF(QNT.NE.CP(NPT,1))GOTO 5
       CP(NPT,2)= CP(NPT,2)+ CT$ GO TO 1
     5 CONTINUE
     7 NP = NP + 1$ IF(NP.GT.NDP) CALL EXIT $ CP(NP,1)= QNT$CP(NP,2)= CT
     1 CONTINUE
    17 M = M-1$ RETURN
       END
       SUBROUTINE SPRNTC(RQN,LGT,COE)
*
*      TO PRINT THE LGT QUANTUM NUMBERS IN RQN AND UNDER THEM THE LGT
*      CORRESPONDING COEFFICIENTS .
       DIMENSION RQN(1),COE(1),IQD(12)
       DATA LEFSQB,RITSQB/1L[,1L]/
       ITPR= 1$ KTOT= (LGT+ 11)/12$ DO 67 K = 1 , KTOT
       ITPM= MIN0(LGT,ITPR+11)$ IMX= ITPM- ITPR+ 1
       DO 1 I = ITPR, ITPM$ IT= I- ITPR+ 1
  1    IQD(IT)= RQN(I)
       PRINT 215,(LEFSQB,IQD(I),RITSQB,I=1,IMX)
 215   FORMAT(* QN=*12(2XA1,I7,A1))
       PRINT 217,(COE(I),I=ITPR,ITPM)
 217   FORMAT(* CI=* 12F11.7)
  67   ITPR= ITPR+ 12$ RETURN
       END
       SUBROUTINE LINCOM(A1,C1,N1,ND1,A2,C2,N2,ND2,C,N,ND)
C      DOES C = A1*C1+ A2*C2 .
       DIMENSION C1(ND1,2),C2(ND2,2),C(ND,2)
       N = 0$ IF(N1.EQ.0) GO TO 1$ IF(N1.GT.ND) CALL EXIT
       DO 3 IT= 1 , N1$ C(IT,1)= C1(IT,1)
     3 C(IT,2)= A1*C1(IT,2)
       N = N1
     1 IF (N2.EQ.0)  RETURN$ DO 5 IT= 1 , N2$ IF(N.EQ.0) GO TO 9
       DO 7 JT= 1 , N$ IF(C(JT,1).NE.C2(IT,1)) GO TO 7
       C(JT,2)= C(JT,2)+ A2*C2(IT,2)$ GO TO 5
     7 CONTINUE
     9 N = N+ 1$ IF(N.GT.ND) CALL EXIT$ C(N,1)= C2(IT,1)
       C(N,2)= A2*C2(IT,2)
     5 CONTINUE$ RETURN
       END
       SUBROUTINE SCALN2(A,CP,NP,NDP,C,N,ND)
*
*      COMPUTES (ABS(ACROSS))**2 ON THE SPIN-CART. STATE IN C AND PUTS
*      THE RESULTING (UNNORMALISED) STATE IN CP .
       DIMENSION CP(NDP,2),C(ND,2),RINC(3),IAN(3)
```

```
      DATA RINC/20000.,200.,2./
      NP = 0$ IF(N.EQ.0) RETURN$ DO 1 IN= 1 , N$ CT= C(IN,1)
      NIN = IFIX(CT)$ CT2= C(IN,2)$ N1= NIN/10000$ NTE= NIN-10000*N1
      N2 = NTE/100$ N3= NTE- 100*N2$ IAN(1)= N1$ IAN(2)=N2$IAN(3)=N3
      IPHAS= 1$ DO 1 JK= 1 , 3$ IF(JK.EQ.2) IPHAS= 1- 2*MOD(N1,2)
      IF(JK.EQ.3) IPHAS= 1- 2*MOD(N1+N2,2)
      FAC= SQRT(FLOAT((IAN(JK)+1)*(IAN(JK)+2)))$ CTP= CT+ RINC(JK)
      IF(NP.EQ.0) GO TO 7$ DO 5 INP= 1 , NP
      IF(CTP.NE.CP(INP,1)) GO TO 5
      CP(INP,2)= CP(INP,2)+ A*FAC*CT2$ GO TO 1
    5 CONTINUE
    7 NP = NP+ 1$ IF(NP.GT.NDP) CALL EXITS$ CP(NP,1)= CTP
      CP(NP,2)= A*FAC*CT2
    1 CONTINUE$ RETURN
      END
      SUBROUTINE WAVSETE(BET,GAMD,CHI,RD,IPR)
*
*     TO COMPUTE THE WAVEFUNCTIONS OF A SINGLE NUCLEON IN AN ELLIP-
*     SOIDAL MODIFIED OSCILLATOR, FOR N= 0-6.  SEE REF. S.G. NILSSON,
*     DAN.MAT.FYS.MEDD.29,NO.16(1955), FOR INTERPRETATION AS A DEF.
*     BASIS OR AS A SPHERICAL BASIS.
*     BET,GAMD,CHI(7), RD(7), ARE INPUT PARAMETERS OF THE MODEL.
*     IPR= -1 ELIMINATES  PRINTOUT, 0.LE.IPR.LE.6 SPECIFIES PRINT
*     DETAILING.
*     THE BLOCK/WAVN/ CONTAINS ALL COMPUTED RESULTS.
*     NOTE THAT THE NATURAL ORDER IS THE ONE IN WHICH THE STATES ARE
*     OBTAINED WHILE THE INCREASING ORDER IS THE ONE FOR WHICH THE
*     ENERGY EIGENVALUES ARE INCREASING. ALL ARRAY SUBSCRIPTS WITH 1.LE.I.LE.
*     84 REFER TO THE NATURAL ORDER EXCEPT THOSE OF EAV(J) AND ITAG(J,1)
*     EAV(J)= AVER. ENERGY, T+.5*V, OF THE STATE WHICH HAS THE JTH
*     LOWEST EIGENVALUE, ITAG(J,1)=I=NATURAL ORDER OF THIS JTH LOWEST
*     STATE. ENE,EAV, ARE IN UNITS OF H BAR OMEGA.
*     ITAG(I,2)= STARTING POSITION IN WAVSET WHERE THE COMPONENTS AND
*     Q.N. OF STATE I ARE STORED. THE END OF THE STATE I IS SIGNIFIED
*     BY STORING 9999XX,(XX=I) IN WAVF(IEND,1) .
*     RJZ(I), QUAD(I,3)= AVERAGES OF JZ,X**2,Y**2,Z**2,DIMENSIONLESS.
*     NHA(I)= MAJOR SHELL QUANTUM NUMBER .
*     TAUP(K)= NON-AXIAL DEFORMATION PARAMETERS .
*     SR(K)= OMEGK/OMEGO.
*     ALL INITIAL SPINS ARE + .
*     THE Q.N. OF THE STATES +000 IS STORED AS +.1.
      COMMON /WAVN/WAVF(1680,2),EAV(84),ENE(84),QUAD(84,3),ITAG(84,2),
     1 RJZ(84),NHA(84),TAUP(3),SR(3)
      COMMON /CARTEMP/ HAM(28,28),VEC(28,28),XHTQ(28,3),JLAB(84),
     1 XHTL(28),RESTA(3962)
      DIMENSION CHI(7),RD(7),TOU(3),SUMA(3),BSQ(3),INS(7),BOBO(3)
      DATA PI,INS,TT/3.1415926535897,0,1,2,3,4,5,6,.66666666666666/
      ND= 28$ GAM= PI*GAMD/180.$ C1= .5*SQRT(1.25/PI)*BET
      DO 1 K = 1 , 3$ TOU(K)= TAUP(K)= C1*COS(GAM- TT*K*PI)
      BOBO(K)= EXP(TOU(K))$ BSQ(K)= BOBO(K)**2
    1 SR(K)= 1./BSQ(K)$ ILAB= IWL= 1
      DO 3 NZ= 1 , 7$ IRET= 0$ NH= NZ-1$ NST= (NH+1)*(NH+2)/2
      CALL MOSDI(NH,TOU,CHI(NZ),RD(NZ),ENE(ILAB),RJZ(ILAB),IPR,IRET,
     1 HAM,VEC,XHTL,XHTQ,ND)
*
*     GENERATE/WAVN/BLOCK
      DO 4 II = 1 , NST$ SUMA(1)= SUMA(2)= SUMA(3)= SUM1= 0.
      ITAG(ILAB,2)= IWL$ DO 5 JJ = 1 , NST$ WAVF(IWL,1)= XHTL(JJ)
      WAVF(IWL,2)= ETA= VEC(JJ,II)$ ETA= ETA**2$ HAMD= 0.
      DO 104 K = 1 , 3$ HAMD= HAMD+ XHTQ(JJ,K)*SR(K)
  104 SUMA(K)= SUMA(K)+ XHTQ(JJ,K)*ETA$ SUM1= SUM1+ HAMD*ETA
    5 IWL= IWL+ 1$ EAV(ILAB)= .5*ENE(ILAB)+ .25*SUM1
```

```
      ITAG(ILAB,1)= ILAB$ NHA(ILAB)= NH
      WAVF(IWL,1)= 999900.+ FLOAT(ILAB)$ WAVF(IWL,2)=0.$IWL= IWL+ 1
      DO 9 K = 1 , 3
*9    QUAD(ILAB,K)= SUMA(K)                                          SPHE.BA.
 9    QUAD(ILAB,K)= HSQ(K)*SUMA(K)                                   DEFO.BA.
 4    ILAB= ILAB+ 1
 3    CONTINUE$ ILAB= ILAB- 1
*
*     ORDER EAV ACCORDING TO INCREASING ENE .
      DO 21 K = 1 , 84$ JLAB(K)= K
 21   XHTQ(K,1)= ENE(K)$ CALL SORTAG(XHTQ,1,84,ITAG)$ DO 23 K = 1,84
 23   XHTQ(K,1)= EAV(ITAG(K,1))$ DO 25 K = 1 , 84
 25   EAV(K)= XHTQ(K,1)$ IF(IPR.LT.0) RETURN
      PRINT 101, BET,GAMD,(INS(J),CHI(J),RD(J),J=1,7)
 101  FORMAT(//43X*ELLIPSOIDAL MODIFIED HARMONIC OSCILLATOR COMPUTATION*
     1 //36X* DEF. PAR. OF POTENTIAL ARE, BETA=*F9.6,*, GAMMA=*F9.3//
     2 33X*MAJOR SHELL N*6X*SPIN-ORBIT STRENGTH CHI*5X*LSQR STRENGT D*
     3 //(38X12,19XF7.4,17XF7.4 ))
      PRINT 103
 103  FORMAT(///25X* STATE ORDERING ACCORDING TO INCREASING ENERGY EIG.
     1VAL., FOR SUMMING PURPOSES*//)
      ITPR= 1$ KMX= (ILAB+ 11)/12$ DO 12 K = 1 , KMX$ ITPM= ITPR+ 11
      PRINT 105,(JLAB(I),I= ITPR,ITPM),(ITAG(I,1),I=ITPR,ITPM)
     1 ,(EAV(I),I=ITPR,ITPM)
 105  FORMAT(16H INCR. ORD.NO. J12(4X12,4X)/16H NATU. ORD.NO. I
     2 12(4X12,4X)/ 16H EAV. OF JTH ST.12(1XF9.6)/)
 12   ITPR= ITPR+ 12$ PRINT 107
 107  FORMAT(//21X*ENERGY EIG.VAL., JZ AVER. AND MAJOR SHELL Q.N. IN THE
     1 NATURAL ORDER(ORDER OF COMPUTATION)*/)
      ITPR= 1$ DO 14 K = 1 , KMX$ ITPM= ITPR+ 11
      PRINT 109, (JLAB(I),I=ITPR,ITPM),(ENE(I),I=ITPR,ITPM),
     1   (RJZ(I)   , I= ITPR, ITPM), (NHA(I),I=ITPR,ITPM)
 109  FORMAT(/16H NATU. ORD.NO. I12(4X12,4X)/16H ENE. OF ITH ST.
     1 12(1XF9.6)/16H JZA, OF ITH ST.12(1XF9.5)/16H NMSH OF ITH ST.
     3 12(4X12,4X))
 14   ITPR= ITPR+ 12$ IF(IPR.LT.1) RETURN$ PRINT 111
 111  FORMAT(//31X*THE EIGENFUNCTIONS IN THE S-C. BASIS FOR INIT. SPIN=
     1+, NAT. ORD.*//)
      CALL SPRNTC(WAVF(1,1),IWL-1,WAVF(1,2))$ RETURN
      END
      SUBROUTINE MOSDI(NH,TOU,CHI,D,ENE,RJZ,IPR,IRET,HAM,VEC,XHTL,
     1XHTQ,ND)
C
C     NH = MAJOR SHELL
C     TOU(K)= SQRT(5/4PI)/2*COS(GAM-2/3K*PI)*BETA .
C     CHI=-L.SIG STRENGTH
C     D= L**2 STRENGTH .
*     IF IRET= 0 SET VEC TO 1      IF IRET= 1 USE CURRENT VALUE OF VEC .
*     RJZ= AVERAGE VALUE OF JZ . VEC= EIGENVECTORS BY COLUMBS,
*     ENE= EIGENVALUES , XHTL(N)= Q. NOS. OF THE BASIS STATES.
*     XHTQ(N,K)=NK*.5 .
      DIMENSION TOU(3),ROMG(3),SR(3),ENE(1),RJZ(1),XHTL(1),XHTQ(ND,3),
     1 HAM(ND,ND),VEC(ND,ND)
      NST= ((NH+1)*(NH+2))/2$ DO 1 K = 1 , 3$ ROMG(K)= EXP(-TOU(K))
 1    SR(K)= ROMG(K)**2$ ITM= NH+1$ DO 3 I = 1 , NST$ DO 3 J = 1 , NST
 3    HAM(I,J)= VEC(I,J)=0.$ C= -CHI*D
*
*     IN HAM PUT H0- X*D*L*SIG*ROMG , IN VEC PUT L*SIG*ROMG .
      N= 1$ DO 7 IT= 1 , ITM$ DO 7 IMU= 1 , IT$ INU= IT- IMU+ 1
      RINU= FLOAT(INU)$ RIMU= FLOAT(IMU)$ RITD= FLOAT(ITM-IT)
      XHTQ(N,1)= RINU- .5$ XHTQ(N,2)= RIMU- .5$ XHTQ(N,3)= RITD+ .5
      XHTL(N)= 10000.*(RINU-1.)+ 100.*(RIMU-1.)+ RITD
```

```
      IF(XHTL(N).LT.1.E-5) XHTL(N)= .1
      HAMT= 0.$ DO 501 K= 1 , 3
 501  HAMT= HAMT+ SR(K)*XHTQ(N,K)$ HAM(N,N)= HAMT$ IF(INU.LE.1) GO TO 5
      T= (1- 2*MOD(INU+1,2))*SQRT((RINU-1.)*(FLOAT(IT)-RINU+1.))
      HAM(N+1,N)= T*ROMG(3)*C$ VEC(N+1,N)= T/ROMG(3)
 5    IF(IT.GE.ITM) GO TO 7
      T= (1-2*MOD(IMU,2))*SQRT(RITD*RIMU)
      HAM(N+IT+1,N)= T*ROMG(1)*C$ VEC(N+IT+1,N)= T/ROMG(1)
      T= (1-2*MOD(IT,2))*SQRT(RITD*RINU)
      HAM(N+IT,N)= T*ROMG(2)*C$ VEC(N+IT,N)= T/ROMG(2)
 7    N= N+1
*     NEXT DO HAM= HAM-X*(VEC+ D*(VEC)**2) AS MATRICES
      DO 23 I = 1 , NST$ DO 23 J = 1 , I$ TPP= 0.
      DO 21 K = 1 , NST$ IF(J-K) 11,13,13
 11   IF(I-K) 17,19,19
 13   TPP= TPP+ VEC(I,K)*VEC(J,K)$ GO TO 21
 19   TPP= TPP+ VEC(I,K)*VEC(K,J)$ GO TO 21
 17   TPP= TPP+ VEC(K,I)*VEC(K,J)
 21   CONTINUE
 23   HAM(I,J)= HAM(I,J)- CHI*(VEC(I,J)+ D*TPP)
 67   CALL DIAGTVD(NST,28,VEC,HAM,IRET)
*
*     COMPUTE JZAVERAGE, SELEC. RULE IS N-NP= 0,+1,-1 .
      DO 120 I = 1 , N
 120  ENE(I)= HAM(I,I)$ N= 1$ DO 27 IT= 1 , ITM
      DO 27 IMU= 1 , IT$ INU= IT-IMU+1$ IF(N.LT.NST) HAM(N+1,N)= 0.
      HAM(N,N)= .5*(1- 2*MOD(IT-1,2))$ IF(INU.LE.1) GO TO 27
      HAM(N+1,N)= (1- 2*MOD(IT+INU,2))*SQRT(FLOAT((INU-1)*IMU))
 27   N= N+1
      DO 31 L = 1 , NST$ T= 0.$ DO 33 N = 1 , NST
      T= T+ HAM(N,N)*VEC(N,L)**2$ IF(N.EQ.NST) GO TO 31
 33   T= T+ 2*HAM(N+1,N)*VEC(N+1,L)*VEC(N,L)
 31   RJZ(L)= T$ IF(IPR.LT.6) RETURN$ PRINT 10
 10   FORMAT(//* EIGENVALUES*20X*EIGENVECTORS*/)
      DO 39 I = 1 , NST
 39   PRINT 61, ENE(I),(VEC(J,I),J=1,NST)
 61   FORMAT(1PE14.6,7E16.6/(14X,7E16.6))
      RETURN
      END
      SUBROUTINE DIAGTFC(N,ND,B,E,IT)
C     B(I,K) CONTAINS THE KTH EIGENVECTOR STORED BY COLUMNS.
C     E(I,J) I.GE.J CONTAINS THE LOWER TRIANGULAR PART OF THE MATRIX
C     TO BE DIAGONALISED . E IS DESTROYED. WHEN RETURNING E(K,K) CON
C     TAINS THE KTH EIGENVALUE .
C     N= DIMENSION OF MATRIX TO BE DIAGONALISED . 1.LE.N.LE.100.
C     ND=DIM.OF ARRAY IN WHICH B AND E ARE STORED,N.LE.ND.
C     DIMENSION B(ND,ND),E(ND,ND)
C     IF IT=0 WHEN ENTERING SET B TO 1 INITIALLY,
C     IF IT.GT.0 WHEN ENTERING USE CURRENT VALUE OF B,
C     THIS OPTION CAN BE USEFUL FOR ITERATIONS.
C     IF DIAGONALISATION FAILS IT IS RETURNED AS -1.
*     ACCURACY, 1 IN 10 TO THE 6.
      DIMENSION B(1),E(1)
      DIMENSION XMS(100),JS(100)
 10   FORMAT(/5X*DIAG FAILS*E11.2/)
      ENTRY DIAGTVD
      IF (N.LT.2) GO TO 501$ IF(IT.GT.0) GO TO 55
      DO 57 I = 1 , N$ IJ= I$ DO 57 J = 1 , N$ B(IJ)= 0.
      IF(I.EQ.J) B(IJ)= 1.
 57   IJ= IJ+ ND
 55   TEST = 0.$ DO 20 I = 1 , N$ IJ= I$ DO 20 J = 1 , I$ EPSP= 2.
      IF(I.EQ.J) EPSP= 1.$ T= E(IJ)$ TEST= TEST+ EPSP*T*T
```

```
 20    IJ= IJ+ ND $ IF(TEST.EQ.0.) GO TO 502$ XNOM= (N*N-N)/TEST
       NMAX= 5*N*N$ KT= 0$ DO 123 K = 2 , N$ XM= 0.$ J= 1$ KK= K- 1
       KL= K$ DO 22 L = 1 , KK$ Q= ABS(E(KL))$ KL= KL+ ND
       IF(Q.LE.XM) GO TO 22$ XM= Q$ J= L
 22    CONTINUE$ XMS(K)= XM
 123   JS(K)= J
*
*      LOOP ENTRY POINT FOR 1 JACOBI ITERATION .
 27    XM= 0.$ DO 222 K = 2 , N$ Q= XMS(K)$ IF(Q.LE.XM) GO TO 222$ XM= Q
       I= K
  222  CONTINUE
       R= XNOM*XM*XM$ IF(R.LT.1.E-12) GO TO 502$ KT= KT+ 1
       IF(KT.GT.NMAX) GO TO 26$ J= JS(I)$ JT= MAX0(2,J)$ MJ= ND*(J- 1)
       IJ= MJ+ I$ P= E(IJ)$ E(IJ)= 0.$ MI= ND*(I- 1)$ JJ= MJ+ J$II=MI+ I
       Y= .5*(E(JJ)- E(II))$ X= P/SIGN(ABS(Y)+ SQRT(Y*Y+ P*P),Y)$ P= P*X
       E(JJ)= E(JJ)+ P$ E(II)= E(II)- P$ C= 1./SQRT(1.+ X*X)$ S= X*C
       KJ= MJ+ 1$ KI= MI+ 1$ JK= J$ IK= I$ DO 25 K = 1 , N$ Y= B(KJ)
       Z= B(KI)$ B(KJ)= C*Y+ S*Z$ B(KI)= C*Z- S*Y$ IF(K-I) 29,24,28
   28  IZJZ=KI
   30  IYJY= KJ$ GO TO 290
   29  IZJZ= IK$ IF(K-J) 31,24,30
   31  IYJY= JK
  290  Y= E(IYJY)$ Z= E(IZJZ)$ E(IYJY)= Y*C+ S*Z$ E(IZJZ)= C*Z- S*Y
 24    KJ= KJ+ I$ KI= KI+ 1$ JK= JK+ ND$ IK= IK+ ND
   25  CONTINUE$ DO 301 K = JT , N$ IF(K-I) 308,309,310
  310  JSK= JS(K)$ IF(J.EQ.JSK.OR.I.EQ.JSK) 309,301
  308  IF(J.NE.JS(K).AND.J.NE.K) GO TO 301
  309  KK= K-1$ XM= 0.$ JJ= 1$ KL= K$ DO 305 L = 1 , KK$ Q= ABS(E(KL))
       IF (Q .LE. XM) GO TO 305$ XM= Q$ JJ= L
  305  KL= KL+ND$ XMS(K)= XM$ JS(K)= JJ
  301  CONTINUE$ GO TO 27
   26  IT= -1$ PRINT 10, R $ GO TO 502
C      ASSUME N= 1
  501  B(1)= 1.
  502  RETURN
       END
       SUBROUTINE DISTOR (TAU1,FNNP,ND,NOD)
*
*      COMPUTE THE XFORMATION FROM THE DEFORMED BASIS TO THE ISOTROPIC
*      ONE ALONG ONE OF THE AXIS, AS A XFORM. ON A 1-DIM. OSCILLATOR.
       DIMENSION FNNP(ND,NOD),TSQR(60)
       EXT= EXP(TAU1)$ EXTI= 1./EXT$ CHT= .5*(EXT+ EXTI)
       SHT = (EXT- EXTI)*.5$ THT= SHT/CHT$ FNNP(1,1)= 1./SQRT(CHT)
       FNNP(2,1)= FNNP(1,1)/CHT$ DO 1 I = 1 , 60
     1 TSQR(I)= SQRT(FLOAT(I))$ DO 3 IDL= 2 , NOD$ IN= 1
       NP = 2*IDL- 3+ IN
       FNNP(1,IDL) = +TSQR(NP-1)/TSQR(NP)*THT*FNNP(1,IDL-1)$ NP= NP+ 1
     3 FNNP(2,IDL)= TSQR(NP)*FNNP(1,IDL)/CHT
       DO 5 IDL= 1 , NOD$ DO 5 IN= 3 , ND$ N= IN- 1$ NP= 2*IDL- 3+ IN
     5 FNNP(IN,IDL)= ((N+NP-1)*FNNP(IN-1,IDL)- TSQR(N-1)*TSQR(NP-1)*CHT*
      1  FNNP(IN-2,IDL))/(TSQR(N)*TSQR(NP)*CHT)$ RETURN
       END
       SUBROUTINE OBCFPE(NTH,CP,NP,ND,IPR)
*
C      COMPUTES COMP. OF NTH LOWEST NILSSON STATE IN N,J+K,J-K, BASIS .
       COMMON /WAVN/WAVF(1680,2),EAV(84),ENE(84),QUAD(84,3),ITAG(84,2),
      1 RJZ(84),NHA(84),TAUP(3),SR(3)
       DIMENSION CP(ND,2),C(84,2)
       ENTRY OBCFP
       J= ITAG(NTH,1)$ ILAB= ITAG(J,2)$ NJ= ITAG(J+1,2)-ILAB-1
       DO 3 I = 1 , NJ$ ILT= ILAB-1+I$ CP(I,1)= WAVF(ILT,1)
     3 CP(I,2)= WAVF(ILT,2)$ CALL TRIMUP(CP,NJ,ND,1.E-5)
```

```
      IF(IPR.LT.0) GO TO 5$ PRINT 101, NTH
  101 FORMAT(*0COMP. OF*I3,*TH L. ORB IN SPIN-CART. DEFORM. BASIS,[MU,NX
     1,NY,NZ] .*)
      CALL SPRNTC(CP,NJ,CP(1,2))
    5 CALL DISTRA(C,N,84,CP,NJ,ND)$ CALL TRIMUP(C,N,84,1,E-5)
      CALL CNNJK(CP,NP,ND,C,N,84)$ CALL TRIMUP(CP,NP,ND,1,E-4)
      IF(IPR.LT.0) RETURN$ PRINT 103
  103 FORMAT(* EXPANSION COEF. IN [NT,J+K,J-K] SPHERICAL BASIS*)
      CALL SPRNTC(CP,NP,CP(1,2)) $ RETURN
      END
      SUBROUTINE DISTRAE(CP,NP,NDP,C,N,ND)
*
*     HERE COMPUTE THE 3-DIM. DEFORMATION XFORMATION USING PRODUCTS OF
*     THE COEFS IN FNNP .
*     C AND CP MUST BE IN THE SPIN-CART. BASIS.
*     C HAS THE COMP. IN THE DEF. BASIS , CP WIL HAVE THE COMP. IN THE
*     ISOTROPIC BASIS .
C     THE MES OF THE TRANSF. FROM THE DEF BASIS TO THE ISOTROPIC ONE
C     ARE CONTAINED IN FNNP(ND,NOD,K), K=1,2,3.
      DIMENSION CP(NDP,2),C(ND,2)
      COMMON/DISTORA/ FNNP(7,4,3)
      ENTRY DISTRA
      NP = 0$ IF(N.LT.1) RETURN$ DO 1 IN= 1 , N$ IQN= IFIX(C(IN,1))
      N1= IQN/10000$ NTE= IQN- 10000*N1$ N2= NTE/100$ N3= NTE- 100*N2
      IR1= MOD(N1,2)$ IR2= MOD(N2,2)$ IR3= MOD(N3,2)
      DO 3 J1 = 1 , 7 , 2$ NP1= J1- 1+ IR1$ IF(NP1.GT.6) GO TO 1
      IQNP1= 10000*NP1$ K1= MIN0(N1,NP1)+ 1$ L1= IABS((NP1-N1)/2)+ 1
      F1= (1-2*MOD(IDIM((N1-NP1)/2,0),2))*FNNP(K1,L1,1)
      DO 5 J2= 1 , 7 , 2$ NP2= J2- 1+ IR2$ IF(NP1+ NP2.GT.6) GO TO 3
      IQNP2= IQNP1+ 100*NP2$ K= MIN0(N2,NP2)+ 1$ L= IABS((NP2-N2)/2)+ 1
      F2= (1-2*MOD(IDIM((N2-NP2)/2,0),2))*FNNP(K,L,2)
      DO 7 J3= 1 , 7 , 2$ NP3= J3- 1+ IR3$ IF(NP1+NP2+NP3.GT.6) GO TO 5
      IQNPT= IQNP2+ NP3$ K= MIN0(N3,NP3)+ 1$ L= IABS((NP3-N3)/2)+ 1
      F3=(1-2*MOD(IDIM((N3-NP3)/2,0),2))*FNNP(K,L,3)
      COET= C(IN,2)*F1*F2*F3$ IF(NP.EQ.0) GO TO 9
      DO 11 INP= 1 , NP$ IF(CP(INP,1).NE.IQNPT) GO TO 11
      CP(INP,2)= COET+ CP(INP,2)$ GO TO 7
   11 CONTINUE
    9 NP = NP+ 1$ IF(NP.GT.NDP) CALL EXIT$ CP(NP,1)= IQNPT
      CP(NP,2)= COET
    7 CONTINUE
    5 CONTINUE
    3 CONTINUE
    1 CONTINUE$ RETURN
      END
      SUBROUTINE CNNJKTE(CP,NP,NDP,C,N,ND)
*
*     WHEREAS STUCO GIVES THE SPIN-CART. COMPONENTS OF THE SPHERICAL
*     STATES THIS ROUTINE STARTS FROM A STATE C IN THE SPIN-CART. BASIS
*     AND PUTS IN CP ITS COMPONENTS IN THE SPHERICAL BASIS .
C     CP(NP,1)= 10000*NT+100*(J+K)+(J-K)
C     CP(NP,2)= COEFFICIENTS .
*     THIS VERSION HAS CORRECT TIME REV. PHASE.
      DIMENSION CP(NDP,2),C(ND,2),CT(30,2)
      COMMON /CARJN/ STUCO(1260,2),IQN(84,6),CSS(300,2),ICSD(49)
      ENTRY CNNJK
      CALL SORTAG(C(1,1) ,1,N,C(1,2))$ NP= 0$ IF(N.EQ.0) RETURN
      DO 1   NY= 1 , 84$ II= IQN(NY,5)$ JL= IQN(NY,6)- IQN(NY,5)+ 1
      DO 3 I = 1 , JL$ IT= II+ I- 1$ CT(I,1)= STUCO(IT,1)
    3 CT(I,2)= STUCO(IT,2)$ SCT= SCALP(CT,JL,30,C,N,ND)
      IF(ABS(SCT).LT.1.E-4) GO TO 1 $ NP= NP+ 1$ IF(NP.GT.NDP) CALL EXIT
      CP(NP,1)= 10000*IQN(NY,2)+50*(IQN(NY,3)+IQN(NY,4))+
```

```
     1 (IQN(NY,3)-IQN(NY,4))/2
*      MODIFY PHASE FROM MUSHINSKY DEF.
       ICH= MOD(IQN(NY,3),8)-2*MOD(IQN(NY,2),2)
       IF(ICH.EQ.3.OR.ICH.EQ.5) SCT= -SCT$ CP(NP,2)= SCT
     1 CONTINUE$ RETURN
       END
       FUNCTION SCALP(CA,NB,NDB,CA,NA,NDA)
C      EACH CI(I,1) MUST BE UNIQUE .
*      CA AND CB MUST BE ORDERED .
       DIMENSION CA(NDA,2), CB(NDB,2)
       SCT= 0.$ IA= IB= 1$ IF(NA.LT.1.OR.NB.LT.1) GO TO 101
       IT= MAX0(NA,NB)$ DO 1 J = 1 , IT
     9  IF (CB(IB,1)- CA(IA,1)) 3, 5, 7
     7 IA = IA+ 1$ IF(IA.GT.NA) 101,9
     3  IB = IB + 1$ IF(IB.GT.NB) 101,9
     5 SCT = SCT+ CB(IB,2)*CA(IA,2) $ IA= IA+ 1$ IB= IB+ 1
       IF(IA.GT.NA.OR.IB.GT.NB) GO TO 101
     1  CONTINUE
   101 SCALP= SCT$ RETURN
       END
       SUBROUTINE TIMEREV(C,NC,ND)
*
*      THE STATE C(ND,2) IS TIME REVERSED.
*
       DIMENSION C(1)
       IQ=1$ IC= ND+1
       DO 1 N = 1 , NC
       IQN= C(IQ)
       JMK= MOD(IQN,100)$ JPK= MOD(IQN/100,100)
       C(IC)= C(IC)*(1-2*MOD(JMK,2))
       IQN= IQN+100*(JMK-JPK)+JPK-JMK$ C(IQ)= IQN
       IC= IC+1$ IQ= IQ+1
     1 CONTINUE $ RETURN
       END
       SUBROUTINE EXFI1B(EXM,K1,K2)
*
*      TO GENERATE MUD CALLS TO EXFI1BP
       COMMON/BASIS/ ICODE(16),ISTA(16),IPAR(16),INPO(16),INUC(16),
      1 ILC(16),NOLV,NB(2),MUD,IDF,IDSC,IDSJ,JAR(14,2),IDAC(14,14),JMPO,
      2 NBT,HBOM,OSPA2
       DIMENSION EXM(1),ICO(16)
       M=1$ DO 1 K = 1 , MUD$ NLP=0$ DO 3 L = 1 , NOLV$ IPR= 3-IPAR(L)
       IF(IPR.NE.K) GO TO 3$ NLP= NLP+1$ ICO(NLP)= ICODE(L)
   3   CONTINUE
       CALL EXFI1BP(EXM(M),ICO,NLP,K1,K2)
   1   M= M+ NB(K)**2
       RETURN
       END
       SUBROUTINE EXFI1BP(EXM,CODA,NOLV,K1,K2)
*      KI= 1,2,3, DO X,Y,Z, KI=4,5,6, DO PX,PY,PZ
*      USING THE STUCO ARRAYS OF J,M STATES IN CARTESIAN BASIS .
*      EXM(NB,NB)= HERMITIAN MATRIX EL. OF K1*K2 .
*      CODA(NOLV)= CODE NUMBER ARRAY .
*      IKIN= +1, REAL SYMM. , IKIN=-1, IM. ANTIS.
       COMMON/BASIS/ ICODE(16),ISTA(16),IPAR(16),INPO(16),INUC(16),
      1 ILC(16),NOLT,NB(2),MUD,IDF,IDSC,IDSJ,JAR(14,2),IDAC(14,14),JMPO,
      2 NBT,HBOM,OSPA2
       INTEGER CODA
       DIMENSION EXM(1),CODA(1),IFAS(6)
       COMMON /CARJN/ STUCO(1260,2),IQN(84,6),CSS(300,2),ICSD(49)
       COMMON/CARTEMP/C(100,2),CP(100,2),CT(100,2),IBSI(50),ISIGN(50),
      1 ITAG(50),ITP(50),LAR(50),RESTA(4876)
```

```
      DATA IFAS/0,1,0,3,0,3/
*
*     NEXT STATEMENT TO TELL SPIN UP FROM SPIN DOWN.
      STUCO(1,1)=.1
      IFT= MOD(IFAS(K1)+IFAS(K2),4)
      IKIN= 1-2*MOD(IFT,2)$ IPH= 1-2*(IFT/2)
      IC1= 1+MOD(K1-1,3)$ IP1= 1-2*((K1-1)/3)
      IC2= 1+MOD(K2-1,3)$ IP2= 1-2*((K2-1)/3)
      ASSIGN 11 TO IHER$ IF(IP1+IP2.EQ.0) ASSIGN 13 TO IHER
*
*     NOW FILLIN IBSI WITH NILSSON NUMBERS IN IQN .
      ILA= 1$ DO 3 K = 1 , NOLV$ KP= CODA(K)$ LAM= ILC(KP)
      JPH= ISTA(KP)$ NPO= INPO(KP)$ JPHD= SHIFT(JPH,1)
      DO 3 M = 1 , JPHD$ JMK= M-1$ JPK= JPHD-M
      ISIGN(ILA)= SHIFT(NPO-1,12)+ SHIFT(JPK,6)+JMK
      ITAG(ILA)= ILA$ LAR(ILA)= LAM
 3    ILA= ILA+1$ IDIM= ILA-1
      CALL SORTAG(ISIGN,1,IDIM,ITAG)
      DO 1 L = 1 , IDIM
 1    IBSI(L)=0
      DO 5 IL= 1 , 84 $ NH= IQN(IL,2)$ NILN= IQN(IL,1)
      TOJ= IQN(IL,3)$ TOK= IQN(IL,4)$ JPK= (TOJ+TOK)/2
      JMK= (TOJ-TOK)/2
      ISIGP= SHIFT(NH,12)+ SHIFT(JPK,6)+JMK
      IR= INBINS(ISIGN,1,IDIM,ISIGP)
      IF(IR.LT.1) GO TO 5$ IBSI(ITAG(IR))=IL$ ITP(ITAG(IR))=JMK
      ISIGP= SHIFT(NH,12)+ SHIFT(JMK,6)+ JPK
      IR= INBINS(ISIGN,1,IDIM,ISIGP)$ IF(IR.LT.1) CALL EXIT
      IBSI(ITAG(IR))= -IL$ ITP(ITAG(IR))= JPK
 5    CONTINUE
      KAREA= IDIM**2$ DO 25 K = 1 , KAREA
 25   EXM(K)= 0.
*
*     IL LOOP STARTS HERE
      DO 7 IL= 1 , IDIM
      NILNO= IBSI(IL)$ IF(NILNO.EQ.0) GO TO 101
      NILN= IABS(NILNO)$ LAL= LAR(IL)
      IAS= IQN(NILN,5)$ IAF= IQN(NILN,6)$ ILA= 1
      DO 9 K = IAS,IAF
      C(ILA,1)= STUCO(K,1)$ C(ILA,2)= STUCO(K,2)
 9    ILA= ILA+1$ ILA= ILA-1
      IF(NILNO.LT.0) CALL TIMEREC(C,ILA,100,ITP(IL))
      GO TO IHER,(11,13)
 11   NT=0$ CALL DIPSTAE(CT,NT,100,IC2,C,ILA,100,1.,IP2)
      NP= 0$ CALL DIPSTAE(CP,NP,100,IC1,CT,NT,100,1.,IP1)$ GO TO 15
 13   NT=0$ CALL DIPSTAE(CT,NT,100,IC2,C,ILA,100,1.,IP2)
      NP=0$ CALL DIPSTAE(CP,NP,100,IC1,CT,NT,100,.5,IP1)
      NT=0$ CALL DIPSTAE(CT,NT,100,IC1,C,ILA,100,1.,IP1)
      CALL DIPSTAE(CP,NP,100,IC2,CT,NT,100,.5,IP2)
 15   CALL SORTAG(CP,1,NP,CP(1,2))
*
*     IM LOOP STARTS HERE
      DO 17 IM= 1 , IL$ LAM= LAR(IM)
*     IF(MOD(LAM+LAL,2).NE.0) GO TO 17
*     THIS PUTS IN THE TIME-REVERSAL PHASE .
      TRPH= (-1)**((LAM-LAL)/2)
                          NILNOM= IBSI(IM)$ NILNM=IABS(NILNOM)
      IASM= IQN(NILNM,5)$ IAFM= IQN(NILNM,6)$ ILM=1
      DO 19 KM= IASM,IAFM$ CT(ILM,1)= STUCO(KM,1)$ CT(ILM,2)=STUCO(KM
 19   ILM= ILM+1$ ILM= ILM-1
      IF(NILNOM.GT.0) GO TO 27$ CALL TIMEREC(CT,ILM,100,ITP(IM))
      CALL SORTAG(CT,1,ILM,CT(1,2))
```

```
 27   PE= SCALP(CT,ILM,100,CP,NP,100)*TRPH
*
*     WE HAVE (M/K1*K2/L) UPPER TRIAN PART
      IF(IKIN.LT.0) GO TO 21 $ EXM(IL+IDIM*(IM-1))= IPH*PE$ GO TO 17
 21   IF(IL.NE.IM) GO TO 23 $ IF(ABS(PE).GT.1.E-10)103,17
 23   EXM(IM+IDIM*(IL-1))= IPH*PE
 17   CONTINUE
 7    CONTINUE$ RETURN
 101  PRINT 105,IL$ GO TO 109
 105  FORMAT (/* FOR IL= *I5,* WE HAVE NO SPHERICAL STATE, EXIT,..*)
 103  PRINT 107, IL,IM ,PE
 107  FORMAT (/*XXXXXXXXXXXX FOR IL AND IM = *2I5,*WE HAVE A ME=*E10.3 )
 109  CALL EXIT
      END
      SUBROUTINE TIMEREC(C,N,ND,MPS)
*
*     C(ND),2) CONTAINS N COMP. IN SPIN-CART. BASIS .
*     ON RETURNING C CONTAINS THE TIME REV. STATE*(-1)**MPS.
*     NOTE THAT /J,-M)= (-1)**(J-M)*T*C WHEN C HAS GOOD J,M .
*     FORMULA IS T*/N1,N2,N3,MU)= MU*/N1,N2,N3,-MU)
*
      DIMENSION C(1)
      IFAS= (-1)**MPS$ IF(N.GT.0) GO TO 1
 3    RETURN
 1    DO 5 K1= 1 , N$ K2= K1+ND
      RQ= C(K1)$ C(K1)= -RQ
 5    C(K2)= C(K2)*SIGN(1.,RQ)*IFAS
      GO TO 3
      END
      SUBROUTINE DIPSTAE(CP,NP,NDP,K,C,N,ND,CBT,IPH)
*
*     DIM CP(NDP,2), C(ND,2)
*     IF IPH= 1, DO XK, IF IPH=-1, DO PK ON THE STATE C.
*     CP= OK*C.
      DIMENSION CP(1),C(1),CT(2),M(3)
      IF (N.LT.1) RETURN$ DO 1 L = 1 , N$ L2= L+ ND$ IT= IFIX(ABS(C(L)))
      M(1) = IT/10000$ NR= IT- 10000*M(1)$ M(2)= NR/100
      M(3)= NR- 100*M(2)$ PHI= 1$ IF(K.LT.3) PHI= 1- 2*MOD(M(3),2)
      IF(K.LT.2) PHI= PHI*(1- 2*MOD(M(2),2))$ SGZ= 1.
      SGZ= SIGN(SGZ,C(L))$ CT(2)= PHI*C(L2)$ IF(K.GT.1)CT(2)= SGZ*CT(2)
      IF(K.LT.3) SGZ= -SGZ$ CT(1)= SGZ*ABS(C(L))$ MT= M(K)
      DO 3 J = 1 , 2$ CTT= SQRT(FLOAT(MT+2-J)/2)
      IF (CTT.LT.1.E-4) GO TO 3$ IF(J.EQ.1) CTT= CTT*IPH
      M(K)= MT+3-2*J$ CT(1)= SGZ*(10000*M(1)+ 100*M(2)+ M(3))
      IF (CT(1).EQ.0.) CT(1)= .1*SGZ$  IF(NP.LT.1) GO TO 8
      DO 7 LP = 1 , NP $ LP2= LP+NDP$ IF(CT(1).NE.CP(LP)) GO TO 7
      CP(LP2) = CP(LP2) + CTT*CT(2)*CBT$ GO TO 3
    7 CONTINUE
    8 IF (NP.LT.0) NP = 0$ CP(NP+1)= CT(1)$ CP(NP+1+NDP)= CTT*CT(2)*CBT
      NP= NP+ 1
    3 CONTINUE
    1 CONTINUE$ RETURN
      END
      SUBROUTINE CARSET(CARA)
      COMMON/BASIS/ ICODE(16),ISTA(16),IPAR(16),INPO(16),INUC(16),
     1 ILC(16),NOLV,NB(2),MUD,IDF,IDSC,IDSJ,JAR(14,2),IDAC(14,14),JMPO,
     2 NBT,HBOM,OSPA2
      DIMENSION ICOD(16),CARA(1)
      INTEGER CARA
      M=1$ DO 1 K = 1 , 2$ NV=0$ DO 3 NT= 1 , NOLV
      IF(IPAR(NT).NE.3-K) GO TO 3$ NV= NV+1$ ICOD(NV)= ICODE(NT)
 3    CONTINUE$ CALL CARSETP(ICOD,NV,CARA(M),IDM)
```

```
1     M= M+ IDM$ RETURN
      END
      SUBROUTINE CARSETP(CODA,NOLP,CARA,IDM)
      COMMON/BASIS/ ICODE(16),ISTA(16),IPAR(16),INPO(16),INUC(16),
     1 ILC(16),NOLV,NB(2),MUD,IDF,IDSC,IDSJ,JAR(14,2),IDAC(14,14),JMPO,
     2 NBT,HBOM,OSPA2
      DIMENSION CARA(1), CODA(1)$ INTEGER CODA,CARA
      DIMENSION ISHC(8),ITJC(8),ITKC(16)
      DATA ISHC/3356B,3456B,3556B,3656B,3756B,4056B,4156B,4256B/,
     1 ITJC/3456B,3656B,4056B,4256B,4456B,3434568,3436568,3440568/,
     2 ITKC/3440B,3436B,3434B,44B,42B,40B,36B,34B,
     3 4634B,4636B,4640B,4642B,4644B,463434B,463436B,463440B/
      ILA=1$ DO 3 K = 1 , NOLP$ KC= CODA(K)$ JPH= ISTA(KC)
      NPO= INPO(KC)$ LTL= SHIFT(JPH,1)$ DO 3 L = 1 , LTL$ LK= L+8-JPH
      CARA(ILA)= SHIFT(ISHC(NPO),42)+ SHIFT(ITJC(JPH),24)
     1 + SHIFT(ITKC(LK),06)+ 6100000000000000062B
3     ILA= ILA+1$ IDM= ILA-1$ RETURN
      END
          IDENT INBINS
         LIST    L,R
*     FUNCTION INBINS(A,II,JJ,X)
*     THE ARRAY A MUST BE IN INCREASING ORDER.
*     A BINARY SEARCH IS DONE IN ARRAY A FROM ELEMENT II TO JJ,
*     FOR THE VALUE X. THE VALUE IX OF THE LOCATION WHERE X IS STORED IN
*     A IS RETURNED AS A FUNCTIONAL VALUE IF THE SEARCH IS SUCCESSFUL.
*     THE VALUE 0 IS RETURNED IF X IS NOT FOUND. THE VALUE -1 IS
*     RETURNED IF  THE SEARCH FAILS BECAUSE OF IMPROPER CALLING.
*     A AND X MUST BE NORMALISED IF THEY ARE REAL.. IF X IS CONTAINED IN
*     SEVERAL LOCATIONS  THE RETURNED VALUE WIL BE ONE OF THOSE.
* THE FOLLOWING IS A FORTRAN CODE WHICH COULD BE USED.
*     INTEGER A,X
*     DIMENSION A(1)
*     IX= II
*     IF(A(IX)-X) 1 , 3 , 5
*   3 INBINS= IX
*   7 RETURN
*   5 INBINS= 0
*     GO TO 7
*   1 IL= IX
*     IX= JJ
*     IF(A(IX)-X) 5,3,9
*   9 IU= IX
*     DO 11 K = 1 , 20
*     IR= IU- IL
*     IF(IR- 1) 5, 5, 13
*  13 IX= IL+ IR/2
*     IF(A(IX)-X) 17 , 3 , 15
*  15 IU= IX
*     GO TO 11
*  17 IL= IX
*  11 CONTINUE
*     INBINS= -1
*     GO TO 7
*     END
         USE     START.
TRACE.   VFD     42/7LINBINS ,18/INBINS
TEMPA0.  BSS     1B
ENTRY.   BSS     0B
         ENTRY   INBINS
INBINS   BSS     1B
         SX6     A0
         SA0     A1                      A0= PAR. LIST FIRST ADDRESS
```

```
          SA6     TEMPA0.
          USE     CODE.
*         X6=IX,B1=IL,B2=IU,B3=AD(A)-18,X1=IR,B4=K-1,B5=19,X2=X.
          SA5     A0+1B                   GET AD(II)
          SA4     A0+3B                   GET AD(X)
          SA3     X5
          BX6     X3                      IX= II
          SA2     X4                      GET X
          SA1     A0                      GET AD(A)
          SB3     X1-1B                   B3= AD(A)-18
          SA4     X6+B3                   X4= A(IX)
          IX0     X4-X2                   X0= A(IX)-X
          ZR      X0,.7                   IF(A(IX)-X) 1,7,5
          NG      X0,.1                   IF(A(IX)-X).LT.0) GO TO 1
          EQ      .5
.7        BSS     0B
          SA4     TEMPA0.                 R
          SA0     X4                       E
          EQ      ENTRY.                     TURN
.5        BSS     0B
          MX6     0                       IX= 0
          EQ      .7
.1        BSS     0B
          SB1     X6                      IL= IX
          SA5     A0+2B                   GET AD(JJ)
          SA3     X5                      X3= JJ
          BX6     X3                      IX= JJ
          SA4     X6+B3                   X4= A(IX)
          IX0     X4-X2
          ZR      X0,.7
          NG      X0,.5
          SB2     X6                      IU= IX
          SB4     0B                      K-1= 0
          SB5     23B                     B5= 19
)AA       BSS     0B                      DO 11 K= 1 , 20
          SX1     B2-B1                   IR= IU-IL
          SB6     X1-1B
          LE      B6,.5                   IF(IR.LT.2) GO TO 5
          AX1     1B                      IR= IR/2
          SX6     X1+B1                   IX= IL+ IR
          SA4     X6+B3                   X4= A(IX)
          IX0     X4-X2
          ZR      X0,.7
          NG      X0,.17
          SB2     X6                      IU= IX
          EQ      .11
.17       BSS     0B
          SB1     X6                      IL= IX
.11       BSS     0B
          SB4     B4+1B                   K= K+1
          LE      B4,B5,)AA               CONTINUE
          SX6     -1B                     IX= -1
          EQ      .7
           END
     OVERLAY(NUHART,2,0)
     PROGRAM NUHART
*
*    FIRST DO ITER ITERATIONS WITH NO C.M. KINETIC ENERGY THEN SAVE
*    THIS WAVEFUNCTION IN RS AND ROS AND DO 1 ITERATION WITH K.E. OF
*    C.M. INCLUDED TO GET THE USUAL SINGLE-PARTICLE ENERGIES. FINALLY
*    DO ITCR ITERATIONS FOR EACH EXTERNAL FIELD EXFIO, USING THE
*    SAVED WAVEFUNCTION .
```

```
      COMMON/BASIS/ ICODE(16),ISTA(16),IPAR(16),INPO(16),INUC(16),
     1 ILC(16),NOLV,NB(2),MUD,IDF,IDSC,IDSJ,JAR(14,2),IDAC(14,14),JMPO,
     2 NBT,HBOM,OSPA2
      COMMON/NAMES/ROP,RON,ROPS,RONS,BRPS,BIPS,BRNS,BINS,VPP,VNP,VPN
     1 , PROT,NEUT,NUCL
      COMMON AS1(4000),AS2(4000),AL1(15000)
      COMMON/NUHARN/VALN(82),VALP(82),NP(2),NN(2),NAME,ITER,ITCR,ATNO
     1 ,NORBA(41),NORB,BET,GAMD,NPTOT,NNTOT
      COMMON/EXFIN/EXFID(2,4,20)
      COMMON/NAMEN/FPR,PTS,FPF,FNE,FNF,FDN,PTD,FDF,RONI,ROPI,SPE,SPEXF
     1 , RTS,RTD
      DATA          ROP,RON,ROPS,RONS,BRPS,BIPS,BRNS,BINS,VPP,VNP,VPN
     1 , PROT,NEUT,NUCL/3LROP,3LRON,4LROPS,4LRONS,4LBRPS,4LBIPS,4LBRNS,
     1 4LBINS,3LVPP,3LVNP,3LVPN,9LPROTON(S),10LNEUTRON(S),10LNUCLEON(S)/
      PRINT 103$ CALL TIMEPRT(Y)$ FACF= .5*HBOM/ATNO
      CALL ADLONG(FPR,1.,PTS,FACF,FPF,AS1,AL1)
      CALL ADLONG(FNE,1.,PTS,FACF,FNF,AS1,AL1)
      CALL ADLONG(FDN,1.,PTD,FACF,FDF,AS1,AL1)
      CALL HARFIT(RONI,ROPI,FPF,FNF,FDF,0,ITER,.T.)
*
*     NOW COMPUTE THE R.M.S. RADIUS W.R.T. THE C.M.
      CALL BUFFIN(ROP,AS1,IDSJ)$ CALL BUFFIN(RTS,AL1,IDF)
      CALL VEEBAF(AS2,AL1,AS1,0)$ TE1= TRVEROJ(AS1,AS2,0)
      CALL BUFFIN( RON,AS1,IDSJ)$ CALL VEEBAF(AS2,AL1,AS1,0)
      TE2= TRVEROJ(AS1,AS2,0)$ CALL BUFFIN(RTD,AL1,IDF)
      CALL VEEBAF(AS2,AL1,AS1,0)$ CALL BUFFIN(ROP,AS1,IDSJ)
      TE3= TRVEROJ(AS1,AS2,0)$ FAMU= OSPA2*.5/ATNO**2
      CALL BUFFIN(3LFCO,AL1,IDF)$ CALL VEEBAF(AS2,AL1,AS1,0)
      TMS= FAMU*(TE1+TE2+2.*TE3) $ RMS= SQRT(TMS)
      ECOUL= .5*TRVEROJ(AS1,AS2,0)$ IF(NPTOT) 703,701
  703 PFACT= OSPA2*.5/(NPTOT*ATNO**2)
      TPMS= PFACT*(TE1*(ATNO+NNTOT)- NPTOT*TE2+ 2.*NNTOT*TE3)
      PRMS= SQRT(TPMS)$ GO TO 705
  701 PRMS= 0.
  705 PRINT 101, RMS, PRMS,ECOUL
*
*     DO 1 MORE ITERATION TO GET S.P. ENERGIES. WITH K.E. ,THEN
*     PROCEED WITH THE EXTERNAL FIELDS IF ANY .
      PRINT 103$ CALL HARFIT(RON,ROP,FPR,FNE,FDN,SPE,1,.F.)
      DO 120 K = 1 , 20$ IF(EXFID(1,1,K).EQ.0) GO TO 99
      PRINT 103$ PRINT 105,((EXFID(I,J,K),I=1,2),J=1,4)
      CALL BUFFIN(EXFID(1,1,K),AS1,IDSJ)$ DO 121 M = 1 , IDSJ
  121 AS1(M)= AS1(M)*EXFID(2,1,K)$ DO 123 J = 2 , 4
      IF(EXFID(1,J,K).EQ.0.OR.EXFID(2,J,K).EQ.0) GO TO 123
      CALL BUFFIN(EXFID(1,J,K),AS2,IDSJ)$ DO 125 M = 1 , IDSJ
  125 AS1(M)= AS1(M)+ AS2(M)*EXFID(2,J,K)
  123 CONTINUE$ CALL BUFOUT(SPEXF,AS1,IDSJ)
  120 CALL HARFIT(RONS,ROPS,FPF,FNF,FDF,SPEXF,ITCR,.F.)
  101 FORMAT(/10H ********* * TOTAL R.M.S. RADIUS W.R.T. C.M.= *F9.5,
     1 * F.,   PROTON R.M.S. RADIUS W.R.T. TOTAL C.M.=*F10.5,* F.  *
     110H**********//10X*COULOMB ENERGY=*F12.5,* MEV.*/)
  103 FORMAT(1HR)
  105 FORMAT(//* THE STRUCTURE OF SPEXF IS.  NAME*10X*AMPLITUDE(MEV)*//(
     115XA10,10XF12.5))
   99 CONTINUE
      END
      SUBROUTINE HARFIT(RONI,ROPI,FPR,FNE,FDN,SPE,ITER,ISAV)
      COMMON/BASIS/ ICODE(16),ISTA(16),IPAR(16),INPO(16),INUC(16),
     1 ILC(16),NOLV,NB(2),MUD,IDF,IDSC,IDSJ,JAR(14,2),IDAC(14,14),JMPO,
     2 NBT,HBOM,OSPA2
      COMMON/NUHARN/VALN(82),VALP(82),NP(2),NN(2),NAME,DD(2),ATNO
     1 ,NORBA(41),NORB,BET,GAMD,NPTOT,NNTOT
```

```
      COMMON/EXSYPN/EXF(36),SYP(21)
      LOGICAL NOSPE,NOSAV,IPRS,ISAV,IPUNCH        $REAL NEUT,NUCL,IXX
      COMMON/NAMES/ROP,RON,ROPS,RONS,BRPS,BIPS,BRNS,BINS,VPP,VNP,VPN
     1 , PROT,NEUT,NUCL
      COMMON/DUMPN/RMES(25,3),STD(3,5),QUADN(5),NPQUAD(5),BETD(5),
     1 GAMP(5),QZOB(5),DDD(185)
      COMMON/PRINLIM/EPS,IPR,QNO(82),IPUNCH
      COMMON/QUADSN/ QUADS(5,3)
      COMMON AS1(4000),AS2(4000),AL1(15000)
      CALL BUFFIN(RONI,AS1,IDSJ)$ CALL BUFFIN(ROPI,AS2,IDSJ)
      CALL BUFOUT(RON ,AS1,IDSJ)$ CALL BUFOUT(ROP,AS2,IDSJ)
      PRINT 209,RONI,ROPI,FPR,FNE,FDN,SPE$ ES=0.
      NOSPE= SPE.EQ.0$ DO 3 IL= 1 , ITER$NOSAV=.N.(ISAV.A.IL.EQ.ITER)
      PRINT 201, IL,NPTOT,NNTOT,NAME$ IPRS=IPR.LT.1.A.IL.NE.ITER
      CALL BUFFIN(ROP,AS1,IDSJ)$ RTP=TRVEROJ(AS1,AS1,0)$ IF(NOSPE)GOTO11
      CALL BUFFIN(SPE,AL1,IDSJ)$ ES= TRVEROJ(AS1,AL1,0)
11    CALL BUFFIN(FPR,AL1,IDF)$ CALL VEEBAR(AS2,AL1,AS1,IPR)
      CALL BUFOUT(VPP,AS2,IDSJ)$ CALL BUFFIN(FDN,AL1,IDF)
      CALL VEEBAR(AS2,AL1,AS1,IPR)$ CALL BUFOUT(VNP,AS2,IDSJ)
      CALL BUFFIN(RON,AS1,IDSJ)$ CALL VEEBAR(AS2,AL1,AS1,IPR)
      CALL BUFOUT(VPN,AS2,IDSJ)$ RTN= TRVEROJ(AS1,AS1,0)$IF(NOSPE)GOTO12
      CALL BUFFIN(SPE,AL1,IDSJ)$ ES= ES+ TRVEROJ(AL1,AS1,0)
12    CALL BUFFIN(FNE,AL1,IDF)$ CALL VEEBAR(AS2,AL1,AS1,IPR)
      CALL BUFFIN(VNP,AL1,IDSJ)$ DO 21 K = 1 , IDSJ
21    AS2(K)= AS2(K)+ AL1(K)$ IXX=0$ IF(IL.EQ.ITER) IXX= NEUT
      IF(IL.EQ.1) GO TO 7$ IF((RTP-NPTOT)**2+(RTN-NNTOT)**2.GT.1.E-20)
     1 CALL ERREXIT(RTP,RTN,NPTOT,NNTOT,IL)
7     ETN= .5*TRVEROJ(AS2,AS1,IXX)$ IF(NOSPE) GO TO 13
      CALL BUFFIN(SPE,AL1,IDSJ)$ DO 23 K = 1 , IDSJ
23    AS2(K)= AS2(K)+ AL1(K)
13    CALL DIAGON(AS2,AS1,AL1,VALN,NN)
      CALL WAVPRT(AS1,AL1,VALN,NNTOT,NEUT,IPRS)
      CALL DENMAT(AS2,AS1,AL1,NN)$ IF(NOSAV) GO TO 30
      CALL BUFOUT(BRNS,AS1,IDSC)$ CALL BUFOUT(BINS,AL1,IDSC)
      IF(IPUNCH) CALL WAVPUN(AS1,AL1,NN,NEUT,NAME)
30    CALL ROCONV(AS2,AL1,1)$IF(.N.NOSAV)CALLBUFOUT(RONS,AL1,IDSJ)
      CALL BUFOUT(RON,AL1,IDSJ)
*
*     NOW REPEAT FOR PROTONS .
      CALL BUFFIN(VPN,AS2,IDSJ)$ CALL BUFFIN(VPP,AL1,IDSJ)$DO25K=1,IDSJ
25    AS2(K)= AS2(K)+AL1(K)$ IXX=0$ IF(IL.EQ.ITER) IXX= PROT
      CALL BUFFIN(ROP,AS1,IDSJ)$ ETP= .5*TRVEROJ(AS2,AS1,IXX)
      IF(NOSPE) GO TO 15$ CALL BUFFIN(SPE,AL1,IDSJ)$ DO 27 K = 1 , IDSJ
27    AS2(K)= AS2(K)+ AL1(K)
15    CALL DIAGON(AS2,AS1,AL1,VALP,NP)
      CALL WAVPRT(AS1,AL1,VALP,NPTOT,PROT,IPRS)
      CALL DENMAT(AS2,AS1,AL1,NP)$ IF(NOSAV) GO TO 32
      CALL BUFOUT(BRPS,AS1,IDSC)$ CALL BUFOUT(BIPS,AL1,IDSC)
      IF(IPUNCH) CALL WAVPUN(AS1,AL1,NP,PROT,NAME)
32    CALL ROCONV(AS2,AS1,1)$IF(.N.NOSAV)CALL BUFOUT(ROPS,AS1,IDSJ)
      CALL BUFOUT(ROP,AS1,IDSJ)$ ETOT= ETP+ETN+ES
      ECOUL= ETP-ETN$ PRINT 203,ETOT,ETP,ETN,ECOUL,ES
3     CONTINUE$ CALL BUFFIN(RON,AS2,IDSJ)$PRINT 205,OSPA2,PROT,NEUT,NUCL
      DO 117 ISY = 1 , 21$ CALL BUFFIN(SYP(ISY),AL1,IDSJ)
      RMES(ISY,1)= TRVEROJ(AS1,AL1,0)$ RMES(ISY,2)= TRVEROJ(AS2,AL1,0)
      RMES(ISY,3)= RMES(ISY,1)+ RMES(ISY,2)
117   PRINT 207,SYP(ISY),(RMES(ISY,J),J=1,3)$ QUADN(1)= PROT
      QUADN(2)= NEUT$ QUADN(3)= NUCL$ NPQUAD(1)= NPTOT$ NPQUAD(2)=NNTOT
      NPQUAD(3)= NPTOT+NNTOT$ DO 119 J = 1 , 3
      STD(3,J)= OSPA2*(RMES(15,J)-RMES(21,J)+RMES(11,J)-RMES(17,J))/3
      STD(1,J)= OSPA2*(-RMES(11,J)+RMES(17,J)+RMES(10,J)-RMES(16,J))/2+
     1 STD(3,J)$ STD(2,J)= STD(1,J)-OSPA2*(RMES(10,J)-RMES(16,J))
```

```
119   CALL GAMEXT(STD(1,J),BETD(J),GAMP(J),QZOB(J),NPQUAD(J),QUADN(J))
      IF(NOSAV) RETURN$ DO 121 J = 1 , 3$ QUADS(4,J)= QUADN(J)
      QUADS(5,J)= NPQUAD(J)$ DO 121 K = 1 , 3
121   QUADS(K,J)= SQRT(STD(K,J)/QUADS(5,J))$ RETURN
201   FORMAT(///* ITERATION NO*I3,* FOR*I3,* PROTONS AND*I3,* NEUTRONS,
     1NUCLEUS= *A10/)
203   FORMAT(///58X20H*********************//5X*TOTAL BINDING ENERGY=*F12
     1.5,* MEV, TOTAL PROTON POTENTIAL ENERGY=*F12.5,* MEV, TOTAL NEUTRO
     1N POTENTIAL ENERGY=*F12.5//25X*DELTAPN ENERGY=*F12.5,* MEV, TOTAL
     3ONE-BODY ENERGY=*F12.5,* MEV*//58X20H*********************//)
205   FORMAT(//  1X*COMPUTATION OF EXPECTATION VALUE OF THE 21 SYMPLECTI
     1C BILINEAR OPERATORS, IN DIMENSIONLESS UNITS, OSC. PAR. BSQUARE=*F
     18.3,* FERMISQ*///13X*NAME*3(10XA10)/)
207   FORMAT(13XA10,2X3(F12.5,8X))
209   FORMAT(///* HARTREE-FOCK ITERATIONS WITH THE PARAMETER LIST,*5X*RO
     1NI       ROPI       FPR        FNE        FDN        SPE*//26X*MASS STO
     2RAGE ARRAY NAMES,*3X6A10)
      END
      SUBROUTINE WAVPUN(BR,BI,NA,KIND,NAME)
*
*     TO PUNCH WITH A UNIQUE HEADER AND NO. OF COMP. TRAILER THE COMP.
*     OF THE WF WHICH ARE GT EPS/3
      COMMON/BASIS/ ICODE(16),ISTA(16),IPAR(16),INPO(16),INUC(16),
     1 ILC(16),NOLV,NB(2),MUD,IDF,IDSC,IDSJ,JAR(14,2),IDAC(14,14),JMPO,
     2 NBT,HBOM,OSPA2
      COMMON /PRINLIM/EPSP
      COMMON /CONTRSN/IDUM(280),IBLK,HBOW,TNAME
      COMMON /DUMPN/IPOS(10),COA(10),NCOM,IBUF,ILA,EPS
      DIMENSION BR(1),BI(1),NA(2)
      EPS= EPSP/3$ CALL TIME(NOW)$ CALL DATE(WHEN)
      PUNCH 103,NAME,NA,KIND,TNAME,IBLK,HBOM,WHEN,NOW
      PRINT 105,NAME,NA,KIND,TNAME,IBLK,HBOM,WHEN,NOW
      NCOM=IBUF=0$ ILA=M=1$ DO 1 K = 1 , 2
      CALL WAVPUP(BR(M),NA(K),NB(K))$ ILA= 1+NB(K)**2
1     M= M+NB(K)**2$ IBUF= IBUF+1$ IPOS(IBUF)= NCOM$ COA(IBUF)= 0
      IF(IBUF.LT.08) GO TO 3$ PUNCH 101,(IPOS(J),COA(J),J=1,08)
      IBUF=0
3     NCOM=0$ ILA=M=1$ DO 5 K = 1 , 2$ CALL WAVPUP(BI(M),NA(K),NB(K))
      ILA= 1+NB(K)**2
5     M= M+ NB(K)**2$ IBUF=IBUF+1$ IPOS(IBUF)= NCOM$ COA(IBUF)=0
      PUNCH 101,(IPOS(J),COA(J),J=1,IBUF)$ RETURN
101   FORMAT(08(I4,F6.4))
103   FORMAT(A10,2I4,2X2A10,I5,F10.2,2A10)
105   FORMAT(* WAVEFUNCTION PUNCHED WITH HEADER= *A10,2I4,2X2A10,I5,F10.
     12,2A10)
      END
      SUBROUTINE WAVPUP(BR,NPA,NBP)
      COMMON /DUMPN/IPOS(10),COA(10),NCOM,IBUF,ILA,EPS
      DIMENSION BR(NBP,NBP)
      DO 1 K = 1 , NPA$ DO 3 I = 1 , NBP$ IF(ABS(BR(I,K)).LT.EPS)GOTO 3
      IBUF=IBUF+1$ IPOS(IBUF)=ILA
      COA(IBUF)= SIGN(AMIN1(ABS(BR(I,K)),.9999),BR(I,K))$ NCOM=NCOM+1
      IF(IBUF.LT.08) GO TO 3$ IBUF= 0$ PUNCH 101,(IPOS(J),COA(J),J=1,08)
101   FORMAT(08(I4,F6.4))
3     ILA= ILA+1
1     CONTINUE$ RETURN
      END
      SUBROUTINE GAMEXT(ST,BET,GAMD,QZOB,NPAT,NAME)
*
*     ST(K)= XK2AV. , BET AND GAMD ARE RETURNED ALONG WITH R.M.S. RAD.
      DIMENSION ST(3), TAU(3)
      DATA PI/3.14159265358/
```

```
      CNUM= 2.*SQRT(5./(4.*PI))$ CNP= 180./PI
      QZOB= (ST(1)*ST(2)*ST(3))**(1./3.)$ A= 0.$ DO 1 K= 1 , 3
      TAU(K)= ALOG(ST(K)/QZOB)/CNUM
 1    A= A+ TAU(K)**2
      BET= SQRT(2.*A/3.)
      GAMD= 0$ IF(BET.LT.1.E-5) GOTO3 $ CGAM= TAU(3)/BET
      IF(CGAM.GT.1.)CGAM=1.$ IF(CGAM.LT.-1.) CGAM=-1.
      GAMD= CNP*ACOS(CGAM)
*     GAMD= CNP*ACOS(CGAM)$ QZOB= SQRT(QZOB/NPAT)
 3    QZOB= SQRT((ST(1)+ST(2)+ST(3))/FLOAT(NPAT))
      PRINT 101, NPAT, NAME , QZOB ,BET,GAMD
 101  FORMAT(/* R.M.S. S.P. RADIUS FOR *I3,1XA10,*=  *F9.5,
     1 * F.,  BET =*F8.5,*,  GAM = *F8.2,* DEG.*/)
      RETURN
      END
          FUNCTION TRVEROJ(VB,RO,NAME)
*
*     TO COMPUTE THE TRACE OF VB*RO WHERE VB AND RO ARE IN N J,M BASIS .
*     IF NAME .NE. 0 PRINT THE COMPONENTS WHICH ARE .GT. EPSP AND THE
*     PARTIAL SUMS OF .5*VB*RO .
      COMMON/BASIS/ ICODE(16),ISTA(16),IPAR(16),INPO(16),INUC(16),
     1 ILC(16),NOLV,NB(2),MUD,IDF,IDSC,IDSJ,JAR(14,2),IDAC(14,14),JMPO,
     2 NBT,HBOM,OSPA2
      DIMENSION VB(2,1),RO(2,1),ILAB(100),IQN(12),DR(12),DI(12),VR(12),
     1 VI(12)$ LOGICAL WESKIP$ COMMON /PRINLIM/ EPS,IPR,QNO(82)
      DATA LSQB,RSQB/1LI,1L]/
      WESKIP= NAME.EQ.0$ XTE= 0.$EPSP=1000.*EPS**2$ IF(WESKIP) GO TO 41
      PRINT 201, NAME, EPSP$ IBUF= 0
 41   ILA= 1$ DO 31 JPO= 1 , JMPO$ J= JPO-1$ IF(WESKIP) GO TO 43
*
*     PREPARE DICTIONARY IN ILAB .
      INC= 0$ DO 11 IPL= 1 , 2$ DO 13 KC= 1 , NOLV
      IF(MOD(IPL-1+IPAR(KC),2).NE.0) GO TO 13$ JPHC= ISTA(KC)
      DO 15 KA= 1 , NOLV$ IF(MOD(IPL-1+IPAR(KA),2).NE.0) GO TO 15
      JPHA= ISTA(KA)
      IF(IABS(JPHC-JPHA)+1.GT.JPO.OR.JPHC+JPHA.LT.JPO) GO TO 15
      INC= INC+1$ ILAB(INC)= 100*ICODE(KC)+ ICODE(KA)
 15   CONTINUE
 13   CONTINUE
 11   CONTINUE
      IF(JAR(JPO,1).NE.INC) CALL ERREXIT(JPO,ILAB,2HFL,INC)
 43   INC= JAR(JPO,1)
      TRJP= 0.$ DO 3 MPO= 1 , JPO$ EPSM= 1.$ IF(MPO.EQ.1) EPSM= .5
      SUMT= 0.$ DO 5 IAC= 1 , INC
      X= RO(1,ILA)*VB(1,ILA)+RO(2,ILA)*VB(2,ILA)
      IF(ABS(X).LE.EPSP.OR.WESKIP) GO TO 21
      IBUF= IBUF+ 1$ IQN(IBUF)= 10000*ILAB(IAC)+ 100*(JPO-1)+MPO-1
      VR(IBUF)=VB(1,ILA)$ VI(IBUF)=VB(2,ILA)$ DR(IBUF)=RO(1,ILA)
      DI(IBUF)=RO(2,ILA)$ ASSIGN 21 TO IRET$ IF(IBUF.GE.12) GO TO 101
 21   SUMT= SUMT+X
 5    ILA= ILA+1
 3    TRJP= TRJP+ EPSM*SUMT$ IF(WESKIP) GO TO 31$ ASSIGN 1 TO IRET
      IF(IBUF.GT.0) GO TO 101
 1    PRINT 103 , J,TRJP
 31   XTE= XTE+ TRJP$ TRVEROJ= 2.*XTE$ RETURN
 101  PRINT .05, (LSQB,IQN(I),RSQB,I=1,IBUF)$ PRINT 107,(VR(I),I=1,IBUF)
      PRINT 109,(VI(I),I=1,IBUF)$ PRINT 111,(DR(I),I=1,IBUF)
      PRINT 113,(DI(I),I=1,IBUF)
      IBUF= 0$ GO TO IRET,(1,21)
 103  FORMAT(* FOR MULTIPOL. LAM=*I3,*, THE TOTAL CONTRIB. TO THE POT. E
     1NERGY IS*F13.5,* MEV*/)
 105  FORMAT(* QN=*12(1XA1,I8,A1))
```

```
 107  FORMAT(* VR=*12(F11.5))
 109  FORMAT(* VI=*12(F11.5))
 111  FORMAT(* DR=*12(F11.5))
 113  FORMAT(* DI=*12(F11.5))
 201  FORMAT(///10X*MULTIPOLE ANALYSIS OF *A10,* H-F POT. V AND DENS. MA
     1TR. D, *ITH LISTING OF ALL COMPS WITH EN. CONTR. GT*F7.4//)
      END
      SUBROUTINE DIAGON(E,BR,BI,VAL,NO)
*
*     FIRST CONVERT E BACK TO (JC,MC/E/JA,MA) BASIS, THEN CALL MATRIX
*     DIAGONALISATION, ORDER THE STATES IN INCREASING ENERGY AND
*     FINALLY REDISTRIBUTE THE PARTICLES IF NEED BE .
      COMMON/BASIS/ ICODE(16),ISTA(16),IPAR(16),INPO(16),INUC(16),
     1 ILC(16),NOLV,NB(2),MUD,IDF,IDSC,IDSJ,JAR(14,2),IDAC(14,14),JMPO,
     2 NBT,HBOM,OSPA2
      DIMENSION E(1),BR(1),BI(1),VAL(1),NO(2)
      CALL ROCONV(BR,E,-1)$ DO 5 K = 1 , IDSC
 5    E(K)= BR(K)
      MA=1$ MB=1$ DO 1 K = 1 , MUD$ ND= NB(K)$ INC= ND+1
      CALL RHERMAD(E(MB),BR(MB),BI(MB),ND,ND,0,IRES,1.E-12)
      IF= MB
      DO 3 L = 1 , ND$ VAL(MA)= E(IF) $ MA= MA+1
 3    IF= IF+ INC
 1    MB= MB+ ND**2
      IS= ISH= 1$ DO 11 K = 1 , MUD$ ND= NB(K)
      CALL ORDER(VAL(IS),BR(ISH),BI(ISH),ND,ND)$ IS= IS+ND
 11   ISH= ISH+ ND**2$ CALL REORDER(VAL,NO)$ RETURN
      END
      SUBROUTINE ORDER(E,BR,BI,NB,ND)
*
*     E IS THE LINEAR EIGENVALUE ARRAY .
      COMMON/DUMPN/ TA(100),ITAG(100),TAP(100)
      DIMENSION E(ND)    ,BR(ND,ND),BI(ND,ND)
      DO 3 J = 1 , NB$ TA(J)= E(J)
 3    ITAG(J)= J$ CALL SORTAG(TA,1,NB,ITAG)$ DO 5 J = 1 , NB
 5    E(J)= TA(J)$ DO 7 I = 1 , NB$ DO 9 J = 1 , NB$ TAP(J)= BI(I,J)
 9    TA(J)= BR(I,J)$ DO 11 J = 1 , NB$ JP=ITAG(J)$ BI(I,J)= TAP(JP)
 11   BR(I,J)= TA(JP)
 7    CONTINUE $ RETURN
      END
      SUBROUTINE WAVPRT(BR,BI,VAL,NNU,NAME,SMPR)
*
*     TO DO A CONTROLLED PRINTOUT OF THE WAVEFUNCTION .
      COMMON/PRINLIM/EPS,IPR,QNO(82)$ LOGICAL SMPR
      COMMON/DUMPN/EN(100),ITA(100),QN(20),CR(20),CI(20),DDD(40)
      COMMON/BASIS/ ICODE(16),ISTA(16),IPPP(16),INPO(16),INUC(16),
     1 LCC(16),NOLV,NB(2),MUD,IDF,IDSC,IDSJ,JAR(14,2),IDAC(14,14),JMPO,
     2 NBT,HBOM,OSPA2
      DIMENSION BR(1),BI(1),VAL(1)
      NS= MIN0(NBT,NNU+12)$ IF(SMPR) NS= 0
      PRINT 101,NS,NAME,EPS$ DO 1 K = 1 , NBT$ EN(K)= VAL(K)
 1    ITA(K)= K$ CALL SORTAG(EN,1,NBT,ITA)$ NSP1= NS+1$ IF(SMPR) GO TO 4
      DO 3 K = 1 , NS$ KP= ITA(K)$ IF(KP-NB(1)) 5,5,7
 5    M=1$ KPP= KP$ IPAR=1$ IDB= NB(1)$ IND=1$ GO TO 9
 7    M= 1+NB(1)**2$ KPP= KP-NB(1)$ IPAR=-1$ IDB= NB(2)$ IND= 1+NB(1)
 9    ILA= M+ IDB*(KPP-1)$ ILB= ILA+ IDB-1$ ILC= 0$ ASSIGN 11 TO IRET
      IOCC= NAME$ IF( K.GT.NNU) IOCC=0$ PRINT 103,K,EN(K),IPAR,IOCC
      DO 11 KS= ILA,ILB$ CRT= BR(KS)$ CIT= BI(KS)$QNT= QNO(IND)
      IND= IND+1$ IF(ABS(CRT).LT.EPS.AND.ABS(CIT).LT.EPS) GO TO 11
      ILC= ILC+1$ QN(ILC)= QNT$ CR(ILC)= CRT$ CI(ILC)= CIT
      IF(ILC.GE.12) GO TO 13
 11   CONTINUE$ IF(ILC.LT.1) GO TO 3$ ASSIGN 3 TO IRET$ GO TO 13
```

```
 3     CONTINUE$ IF(NS.GE.NBT) RETURN
 4     PRINT 111,(EN(I),I=NSP1,NBT)$ RETURN
       RETURN
 13    PRINT 105,(QN(J),J=1,ILC)$ PRINT 107,(CR(I),I=1,ILC)
       PRINT 109,(CI(J),J=1,ILC)$ ILC=0$ GO TO IRET,(3,11)
 101   FORMAT(//30X10H********** * THE LOWEST *I2,1XA10,
      1 * STATES, ALL COMP. .GT. *F5.3,1X10H********** //)
 103   FORMAT(/* STATE NO.*I3,*, WITH S.P. EN.=*F12.6,* MEV., PAR.=*I3,*,
      1 OCCUPIED BY A *A10)
 105   FORMAT(4H QN=12(1XA10))
 107   FORMAT(4H RP= 12(1XF10.7))
 109   FORMAT(4H IP= 12(1XF10.7))
 111   FORMAT(/* REMAINING S.P. ORBITS AT ENERGIES(MEV)*//(2X10F12.5 ))
       END
       SUBROUTINE RHERMAD(E,BR,BI,NBE,IDIM,IBT,IRES,R2)
*
*      E= HERM. MATRIX TO BE DIAGONALISED.
*      E(I,J), I.GE.J, CONTAINS REAL PART OF E,
*      E(I,J), I.LT. J, CONTAINS IMAGINARY PART OF E .
*      B(IDM,IDM)= ARRAY CONTAINING EIGENVECTORS BY COLUMNS.
*      BR= REAL PART OF B, BI= IM. PART OF B.
*      NBE= DIM OF MATRIX TO BE DIAGONALISED.
*      IDIM= DECLARED DIM OF E AND B, EG, A(IDIM,IDIM), B(IDIM,IDIM).
*      IBT= 0, NO TRIAL EIGENVECTORS, SET B TO 1 INITIALLY.
*      IBT=1, USE CURRENT B.
*      IRES= 0, HERMAD HAS FAILED, IRES=1 IF DIAG. IS COMPLETED.
*
       COMMON/DUMPN/XMS(100),JS(100),DDD(100)
       DIMENSION E(1),BR(1),BI(1)
       N=NBE$ ND=IDIM$ IRES=1$RL=R2*R2
       IF(RL.GT..1.OR.RL.LT.1.E-24) RL= 1.E-12
  10   FORMAT(/5X,* HERMAD FAILS, R= *1PE8.2/)
       IF(N.LT.2) GO TO 501$ IF(IBT.GT.0) GO TO 55
       DO 59 I= 1 , N
       IJ= I
       DO 59 J= 1 , N
       BR(IJ)=0.$ BI(IJ)=0.$ IF(I.EQ.J) BR(IJ)=1.
  59   IJ= IJ+ ND
  55   TEST= 0.$ MI=0
       DO 20 I= 1 , N
       IJ= I$ JI= MI+1
       DO 22 J = 1 , I
       EPSP= 2.$ IF(I.EQ.J) EPSP= 1.
       H=1.$ IF(I.EQ.J) H= 0.
       TEST= TEST+ (E(IJ)**2+H*E(JI)**2)*EPSP$ IJ= IJ+ ND
  22   JI= JI+1
  20   MI= MI+ ND
       IF(TEST.EQ.0.) GO TO 502
       XNOM= (N*N-N)/TEST$ NMAX= 5*N*N$ KT=0$ MK=ND
       DO 123 K= 2 , N
       XM=0.$ J=1$ KK=K-1$ KL=K$ LK=MK+1
       DO 32 L= 1 , KK
       Q= E(KL)**2+ E(LK)**2$ KL=KL+ND$ LK=LK+1
       IF(Q.LE.XM) GO TO 32$ XM=Q$ J=L
  32   CONTINUE$ XMS(K)= XM$ JS(K)= J
 123   MK= MK+ND
  27   XM=0.
       DO 222 K= 2 , N
       Q=XMS(K)$ IF(Q.LE.XM) GO TO 222$ XM=Q$ I=K
 222   CONTINUE
       R= XNOM*XMS$ IF(R.LT.RL    ) GO TO 502
       KT= KT+1$ IF(KT.GT.NMAX) GO TO 26
```

```
      J=JS(I)$ JT=MAX0(2,J)$ MJ=ND*(J-1)$ IJ=MJ+I$ MI=ND*(I-1)
      JI= MI+J$ JJ= MJ+J$ II= MI+I
      PU= E(JI)$ YU= E(IJ)$ PSQ=PU**2+ YU**2$ E(IJ)=0.$ E(JI)=0.
      P= SIGN(SQRT(PSQ),YU)$ Y= (E(JJ)-E(II))*.5
      X= P/SIGN(ABS(Y)+SQRT(Y*Y+PSQ),Y)
      XU= PU/SIGN(ABS(YU)+ ABS(P),YU)
      C= 1./SQRT(1.+X*X)$ S= X*C
      CU= 1./SQRT(1.+XU**2)$ SU= XU*CU
      P= P*X$ E(JJ)= E(JJ)+P$ E(II)= E(II)-P
      KJ= MJ+1$ KI= MI+1$ JK=J$ IK=I
      DO 25 K= 1 , N
      YR= CU*BR(KJ)-SU*BI(KJ)$ YI=SU*BR(KJ)+CU*BI(KJ)
      ZR=CU*BR(KI)+SU*BI(KI)$ ZI=-SU*BR(KI)+CU*BI(KI)
      BR(KJ)=C*YR+S*ZR$ BR(KI)= -S*YR+C*ZR
      BI(KJ)=C*YI+S*ZI$ BI(KI)=-S*YI+C*ZI
      IF(K-I) 29,24,28
   28 YR=CU*E(KJ)+SU*E(JK)
      YI= SU*E(KJ)- CU*E(JK)
      ZR= CU*E(KI)- SU*E(IK)
      ZI= -SU*E(KI)- CU*E(IK)
      E(KJ)= YR*C + S*ZR
      E(KI)= -S*YR+ C*ZR
      E(JK)= -C*YI- S*ZI
      E(IK)= S*YI- C*ZI$ GO TO 24
   29 IF(K-J) 31, 24, 30
   31 YR= CU*E(JK)- SU*E(KJ)
      YI= -SU*E(JK)- CU*E(KJ)
      ZR= CU*E(IK)+ SU*E(KI)
      ZI= SU*E(IK)- CU*E(KI)
      E(JK)= C*YR+ S*ZR
      E(IK)= -S*YR+ C*ZR
      E(KJ)= -C*YI- S*ZI
      E(KI)= S*YI- C*ZI$ GO TO 24
   30 YR= CU*E(KJ)+ SU*E(JK)
      YI= SU*E(KJ)- CU*E(JK)
      ZR= CU*E(IK)+ SU*E(KI)
      ZI= -SU*E(IK)+ CU*E(KI)
      E(KJ)= C*YR+ S*ZR
      E(IK)= -S*YR+ C*ZR
       E(JK)= -C*YI- S*ZI
      E(KI)= -S*YI+ C*ZI
   24 KJ= KJ+1$ KI= KI+1$ JK= JK+ND$ IK= IK+ND
   25 CONTINUE $ MK= ND*(JT-1)
      DO 301 K= JT, N
      JSK= JS(K)$ IF(K-I) 308,309,310
  310 IF(J.EQ.JSK.OR.I.EQ.JSK) 309,301
  308 IF(J.NE.JSK.AND.J.NE.K) GO TO 301
  309 KK= K-1$ XM=0.$ JJ=1$ KL=K$ LK= MK+1
      DO 305 L= 1 , KK
      Q= E(KL)**2+E(LK)**2$ LK= LK+1$ IF(Q.LE.XM) GO TO 305
      XM= Q$ JJ=L
  305 KL= KL+ND$ XMS(K)= XM$ JS(K)= JJ
  301 MK= MK+ND$ GO TO 27
   26 IRES=0$ PRINT 10,R$ GO TO 502
  501 BR(1)=1.$ BI(1)=0.
  502 RETURN
      END
      SUBROUTINE VEEBAR(VB,F,ROJ,IPR)
*
*     TO FIND THE AVERAGE ONE-BODY POTENTIAL GIVEN F THE FORCE ARRAY
*     AND ROJ THE DENSITY MATRIX IN THE J,M REPRESENTATION .
      COMMON/BASIS/ ICODE(16),ISTA(),),IPAR(16),INPO(16),INUC(16),
```

```
     1 ILC(16),NOLV,NB(2),MUD,IDF,IDSC,IDSJ,JAR(14,2),IDAC(14,14),JMPO,
     2 NBT,HBOM,OSPA2
      DIMENSION VB(2,1),ROJ(2,1),F(1)
      ITSS= 1$ GO TO 15$ ENTRY VEEBAF$ ITSS= 0
15    IF(IPR.GT.0) CALL TIMEPRT(X)$ IF(ITSS.EQ.0) GO TO 13
      IF(INT(F(1)).NE.NOLV) CALL ERREXIT(NOLV,F,2HFL,200)
      DO 25 K = 1 , NOLV$ IF(INT(F(K+1)).NE.ICODE(K))
     1 CALL ERREXIT(K,ICODE,2HFL,NOLV,F,2HFL,200)
25    CONTINUE
13    IFL= NOLV+1$ IF(IPR.GT.4) PRINT 409,IDSJ,(ROJ(I,1),I=1,IDSJ)
409   FORMAT(/* ROJ ARRAY,*I5,* ELEMENTS*/1P(1X10E13.5))
      ILR= 0$ ILV= 1$ DO 1 JPO= 1 , JMPO$ JARJ= JAR(JPO,1)
      IF(ITSS.EQ.0) GO TO 17$ IF(INT(F(IFL+1)).NE.SHIFT(JPO-1,6)+JARJ)
     1 CALL ERREXIT(JPO,IFL,JAR,2HFL,14,F(IFL-100),2HFL,200)
17    IFL= IFL+1$ DO 3 MPO= 1 , JPO$ IF(JARJ.EQ.0) GO TO 3
      DO 5 L = 1 , JARJ$ VB(1,ILV)=VB(2,ILV)=0.$ LMF= L+IFL
      DO 7 M = 1 , JARJ$ VB(1,ILV)= VB(1,ILV)+ F(LMF)*ROJ(1,ILR+M)
      VB(2,ILV)= VB(2,ILV)+ F(LMF)*ROJ(2,ILR+M)$ IDEL= 1
      IF(M.LT.L) IDEL= JARJ-M
7     LMF= LMF+ IDEL
5     ILV= ILV+1
3     ILR= ILR+ JARJ
1     IFL= IFL+(JARJ*(JARJ+1))/2
      IF(IPR.GT.5) PRINT 411,IDSJ,(VB(I,1),I=1,IDSJ)
411   FORMAT(/* VBJ ARRAY,*I5,* ELEMENTS*/1P(1X10E13.5))
      RETURN
      END
      OVERLAY(DENSITY,4,0)
      PROGRAM DENSITY
      COMMON/QUADSN/ QUADS(5,3)
      COMMON/BASIS/ ICODE(16),ISTA(16),IPAR(16),INPO(16),INUC(16),
     1 ILC(16),NOLV,NB(2),MUD,IDF,IDSC,IDSJ,JAR(14,2),IDAC(14,14),JMPO,
     2 NBT,HBOM,OSPA2
      COMMON/NUHARN/VALN(82),VALP(82),NP(2),NN(2),NAME,ITER,ITCR,ATNO
     1 ,NORBA(41),NORB,BET,GAMD,NPTOT,NNTOT
      COMMON /CODAN/ICODAP(16),ICODAN(16),LVP,LVN,IQM(16),NOLT
      COMMON /PLOT1/ LAY,NMAX/ADEN1/DTHE,ITHE,IRAD,MAXX
      COMMON/PLOT3/X(3),IAXIS(3),UNIT1,XNAME(3),DA,BETA,ABSS(9),
     1    DAR(9),NPN(2),COMM(2)
      COMMON YLM(4000),HNL(4000),WAVF(328),RMS(3),AS1(2)
*     **** NOTE**** AS1 OVERWRITTEN TO LENGTH 40000 .
*     DATA COMM/10H TOTAL DEN,10HSITY      /
*      LAY IS THE NO. OF LEVELS OF DENSITIES TO BE PRINTED IN DENPLOT.
*      UNIT1 IS THE UNIT DENSITY IN DENPLOT, E.G. THE SYMBOL 2 IN DEN-
*      PLOT REPRESENTS A DENSITY OF 1.75 TO 2.25 UNIT1(NUCEON/FERMI CUBED).
      DATA LAY,NMAX,UNIT1/40,54,.017/,MAXX/300/,JWANT/3/
*
      CALL TIMEPRT(YD)$ CALL BUFFIN(4LROPS  ,YLM,IDSJ)
      ENCODE(20,101,COMM)HBOM
101   FORMAT(*TT DNS,HBOM=*F6.1)
      CALL ROCONV(AS1,YLM,-1)$ CALL BUFFIN(4LRONS  ,YLM,IDSJ)
      CALL ROCONV(HNL,YLM,-1)$ DO 1 I = 1 , 3
1     RMS(I)=  QUADS(I,JWANT)$ DO 2 I = 1 , IDSC
2     AS1(I)= AS1(I)+ HNL(I)$ NPN(1)= NPTOT$ NPN(2)= NNTOT
      CALL CALYLM(YLM,5,181,DTHE,ITHE)$ CALL CALHNL(HNL,MAXX,5,IRAD)
      LVP=LVN=0$ DO 3 II = 1 , NOLV$ IF(MOD(IPAR(II),2).EQ.0) GO TO 5
      LVN= LVN+1$ ICODAN(LVN)= ICODE(II)$ GO TO 3
5     LVP= LVP+1$ ICODAP(LVP)= ICODE(II)
3     CONTINUE$ DO 7 K = 1 , LVP
7     IQM(K)= ICODAP(K)$ DO 9 K = 1 , LVN
9     IQM(K+LVP)= ICODAN(K)$ NOLT= NOLV
      CALL DENPLOT(AS1,RMS,WAVF,YLM,HNL,NB,NBT,HBOM,NAME)
```

```
      END
      SUBROUTINE CALYLM(YLM,LM,NM,DX,IDIM)
*          CALCULATE THE REAL NORMALIZED ASSOCIATED LEGENDRE POLINOMIALS
*          FOR L=0 TO L=LM IN INTERVALS OF THETA DX=PI/(2*(NM-1)) AND
*          STORE IN THE LINEAR ARRAY YLM.  THE DIMENSION OF YLM MUST NOT BE LESS
*          THAN IDIM*NM, WHERE IDIM IS THE NUMBER OF PAIRS (L,M)= (0,0),
*          (1,0), (1,1), (2,0).......,  (LM,LM-1), (LM,LM).  OR
*               IDIM=((LM+1)*(LM+2))/2.      THE ADDRESS OF THE VALUE FOR
*           (L,M,THETA) IS   THE SAME AS THE ADDRESS (LM,IX) IN A IDIM*NM
*          MATRIX, WHERE LM=(L*(L+1))/2+M+1, I.E., ADD=IDIM*(IX-1)+LM.
      DIMENSION YLM(1)
      LMAX=LM+1$ IDIM=((LMAX+1)*LMAX)/2$ DX=3.141592653/(2.*FLOAT(NM-1))
      I=1$ DO 1 L1=1,LMAX$ L=L1-1$ X=-DX$ K=I
      DO 2 M=1,NM$ X=X+DX$ CALL NLEGEND(X,YLM(K),L)
    2 K=K+IDIM
    1 I=I+L1$  RETURN
      END
      SUBROUTINE NLEGEND(THETA,A,N)
*          BASED ON THE ALEGEND SUBROUTINE IN THE CDC 6600 LIB. AT CRNL.
*          A(M+1)= Y(N,M,THETA) = (-1)**M*SQRT((2*N+1)*FACT(N-M)/
*                                 (4*PI*FACT(N+M)))*P(N,M,THETA). FOR M.GE.0
      DIMENSION RTFAC(20),A(1)
      LOGICAL IPASS
      DATA IPASS/.FALSE./
      IF(IPASS) GOTO2$ RTFAC(1)=1.$ DO 1 I=2,20$ P=SQRT(FLOAT(I-1))

    1 RTFAC(I)=P*RTFAC(I-1)$   IPASS=.TRUE.
    2 P=SQRT(FLOAT(2*N+1)/(4.*3.141592653)) $ IF(N.GE.1)GOTO20
      A(1)=P$  RETURN
   20 IF(ABS(THETA).GT.1.E-10)GOTO30$ A(1)=P$  DO 21 K=1,N
   21 A(K+1)=.0$   RETURN
   30 IF(ABS(THETA-3.1415926536).LT.1.E-11)31,40
   31 K=N/2$  IF(K*2.EQ.N)311,312
  311 A(1)=P$ GOTO313
  312 A(1)=-P
  313 DO 314 K=1,N
  314 A(K+1)=0$  RETURN
   40 Y=SIN(THETA)$ PROD=N+1$ K1=N+2$ K2=2*N$ IF(K2.LT.K1) 52,50
   50 DO 51 K=K1,K2
   51 PROD=K*PROD
   52 PNR=A(N+1)=P*PROD*(Y/2.)**N$ PNS=.0$ COT=COS(THETA)/Y
      DO 60 MM=1,N$ M=N-MM$ A(M+1)=(2*(M+1)*COT*PNR-PNS)/((N+M+1)*(N-M))
      PNS=PNR$ PNR=A(M+1)
   60 A(M+1)=RTFAC(N-M+1)*(-1.)**M*A(M+1)/RTFAC(N+M+1)
      A(N+1)=(1-2*MOD(N,2))*A(N+1)/RTFAC(N+N+1)$  RETURN
      END
      SUBROUTINE CALHNL(HNL,XMAX,NM,IDIM)
*          CALCULATE THE HARMONIC OSC. WAVE FUNC. FOR ALL STATES WITH TOTAL
*          QUANT. NO. LE NM, FROM X=0 TO X=4.-DX IN INTERVALS OF DX AND STORE
*          IN THE LINEAR ARRAY HNL.   X=R/BETA, BETA=1./SQRT(.0241145*HW).
*          HNL(I)=RDFUNC(N,L,X),  WHERE I=IDIM*(IX-1)+INL,
*          INL=(NQ/2+1)*(NQ/2)+MOD(NQ,2)*((NQ-1)/2+1)+L/2+1,  NQ=2*N+L
*          THE DIMENSION OF HNL MUST NOT BE LESS THAN IDIM*XMAX.
      INTEGER XMAX
      DIMENSION HNL(1)
      IDIM=((NM+3)/2)*((NM+1)/2)+MOD(NM+1,2)*(NM/2+1)$ DX=4./FLOAT(XMAX)
       NMAX=NM+1$ IK=0$ X=-DX
      DO 1 K=1,XMAX$ X=X+DX$ DO 1 N1=1,NMAX$ IF(MOD(N1-1,2))6,3,4
    3 LMIN=1$ GOTO5
    4 LMIN=2
    5 DO 1 L1=LMIN,N1,2$ L=L1-1$ N=(N1-L1)/2$ IK=IK+1
    1 HNL(IK)=RDFUNC(N,L,X)$  RETURN
```

```
 6     CALL ERREXIT(N1)
       END
       FUNCTION RDFUNC(N,L,X)
C      NORMALIZED H.O.RADIAL-FUNCTION UPTO NOSC=20 (NOSC=2*N+L)
       DIMENSION FL(11),GL(21)
       LOGICAL IPASS$ DATA IPASS/.F./
       DATA N1MAX,L1MAX/11,21/
       N1= N+1$ L1= L+1$ RDFUNC=0$ IF(N.LT.0.OR.L.LT.0) RETURN
       NL= N+L+1$ IF(N1.GT.N1MAX.OR.NL.GT.L1MAX) GO TO 5
       IF(IPASS) GO TO 2$ IPASS= .T.$ FL(1)=0.$ GL(1)= ALOG(0.5)
       DO 1 M = 2 , L1MAX$ A= M-1$ IF(M.GT.N1MAX) GO TO 1
       FL(M)= FL(M-1)+ALOG(A)
 1     GL(M)= GL(M-1)+ALOG(A+0.5)
       A= 2./SQRT(3.141593)
 2     XSQ= X*X
C      EVALUATE LAGUERRE POLYNOMINAL POLAG(N,L+1/2,XSQ)
       POLAG= EXP(GL(NL)-GL(L1)-FL(N1))$ IF(N.EQ.0) GO TO 4$ B= 1.
       DO 3 M=1,N$ B= -B*XSQ
 3     POLAG= POLAG+B*EXP(GL(NL)-GL(L1+M)-FL(N1-M)-FL(M+1))
 4     B= EXP(FL(N1)-GL(NL))$ B= SQRT(A*B)
       RDFUNC= B*(X**L)*EXP(-0.5*XSQ)*POLAG$ RETURN
 5     PRINT6,N,L$ RETURN
 6     FORMAT(/1X,*XXXXX   N=*,I2,2X,*L=*,I2,2X,*N OR L TOO LARGE*,
     11* CANNOT CALCULATE RADFNC.CHANGE DIMENSION AND DATA   XXXXX*)
       END
       SUBROUTINE DENPLOT(RO,RMS,WAVF,YLM,HNL,NB,NBT,HW,NAME)
*           ROUTINE ONLY CALCULATES THE FIRST QUADRANT DENSITY FOR ANY GIVEN
*           CROSS-SECTION. IN OTHER WORDS, PARITY CONSERVATION IS ASSUMED.
       COMMON/PLOT3/X(3),IAXIS(3),UNIT1,XNAME(3),DA,BETA,ABSS(9),
      1    DAR(9),NPN(2),COMM(2)
       COMMON/PLOT2/AS2(4000)
       COMMON /PLOT1/LAY,NMAX
*      COMMON/SYMB/LIT(40)$ INTEGER  TAG(3)$  REAL NAME
       COMMON/SYMB/LIT(40)
       LOGICAL R1,R2,R3,FLIP,IPASS
       DIMENSION RO(1),RMS(1),YLM(1),HNL(1),NB(1),WAVF(1),ADEN(60),Y(3)
       DIMENSION DA9(9),DB9(9)
       EQUIVALENCE(I,IAXIS(1)), (I2,IAXIS(2)), (I3,IAXIS(3))
       DATA DB/10H+          /, IPASS/.F./
       IF(IPASS)GOTO7$ ENCODE( 90,51,DA9)DA,DA,DA,DA,DA,DA,DA,DA,DA
       ID=DATE(T)$ ENCODE(90,51,DB9)DB,DB,DB,DB,DB,DB,DB,DB,DB$IPASS=.T.
 7     JKX=1$ IF(NAME.EQ.5LMG 24.OR.NAME.EQ.5LS  32) JKX=0
       BETA=1./SQRT(.0241145*HW)$ R2=ABS(RMS(1)-RMS(2)).LT..1
*           0.17 PER FERMI CUBED IS THE NUCLEAR MATTER DENSITY.
       B3=BETA**3$ DDEN1=0.17*B3$ UNIT=UNIT1*B3/2
       R1=ABS(RMS(2)-RMS(3)).LT..1$ R3=ABS(RMS(1)-RMS(3)).LT..1
*      N2=NMAX/2$ DO10 I=1,3$ Y(I)=(RMS(I)*3.0/BETA)**2
*  10  TAG(I)=I$ CALL SORTAG(RMS,1,3,TAG)$ X(TAG(1))=X(TAG(2))=0
*      XM=AMIN1(3.9,RMS(TAG(3))*3.0/BETA)$DX=XM/N2$XMAX=XM+.5*DX$XM=XM+DX
*      XM=AMIN1(3.9,3.6/BETA)$          DX=XM/NMAX$XMAX=XM+.5*DX$XM=XM+DX
*      II=TAG(3)$ X(II)=-DX$ W=0$ DO 11 I=1,N2$ X(II)=X(II)+DX
*      CALL CWVFNC(X,NBT ,WAVF,YLM,HNL)$ DEN=DENSD(RO,WAVF,1.,1.,NB,2)
*  11  W=AMAX1(DEN,W)$  DDEN=W/FLOAT(LAY)$ FLIP=.F.$ DZ=6*BETA*DX
       N2=NMAX/2$ FLIP=.F.$ DO10 I=1,3$ Y(I)=(RMS(I)*3.0/BETA)**2
   10  CONTINUE$ XM=AMIN1(3.9,3.6/BETA)$ DX=XM/NMAX$ XMAX=XM+.5*DX
       XM=XM+DX$ DZ=6*BETA*DX
       DO 1 I=1,3$ IF(I.EQ.2.AND.R2)GOTO1$ IF(I.EQ.3.AND.(R1.OR.R3))GOTO1
       I2=MOD(I,3)+1$ I3=MOD(I+1,3)+1$ IF(Y(I3)-Y(I2))3,6,6
  3    K=I2$ I2=I3$ I3=K
  6    YY=SQRT(Y(I2)/Y(I3))$ X(I2)=-DX  $ X(I)=.0$ KK=0
 13    X1=X(I)*X(I)/Y(I)
       PRINT1000,ID$  PRINT1001, DA9
```

```
      DO 2 J=1,NMAX$ X(I2)=X(I2)+DX$ X(I3)=-DX  $  DO 4 K=1,NMAX
    4 ADEN(K)=.0$ X2=X(I2)*X(I2)+YY/Y(I2)$ DO 5 K=1,NMAX$ X(I3)=X(I3)+DX
      IF(X1+X2+X(I3)*X(I3)/Y(I3).GT.1.) GOTO5
      CALL CWVFNC(X,NBT,WAVF,YLM,HNL)$ ADEN(K)=DENSO(RO,WAVF,1.,1.,NB,2)
    5 CONTINUE$ IF(J.EQ.NMAX) FLIP=.T.$ DO12 N=1,NMAX
*  12 AS2(KK+N)=ADEN(N)$ CALL LINPLT1(ADEN,DDEN,LAY,NMAX,FLIP)
   12 AS2(KK+N)=ADEN(N)$CALL LINPLT2(ADEN,UNIT,LAY,NMAX,DAR)
      CALL FOHMT(J,NAME)
    2 KK=KK+NMAX$ PRINT1001,DA9$ PRINT1002,DB9$ PRINT1003,ABSS,XNAME(I3)
      CALL LINPRT1(AS2,DDEN1,NMAX,83)
      IF(I.NE.3.OR.JKX.NE.0)GOTO1$X(I)=JKX=1$X(I2)=-DX$KK=0$ GOTO13
    1 CONTINUE$  RETURN
   51 FORMAT(9A10)
 1000 FORMAT(1H1,18X*DENSITY PLOT FROM EVALIN ( R.Y. CUSSON/ H.C. LEE )
     1C.R.N.L. DATE*A10//)
 1001 FORMAT(9X***9A10***)
 1002 FORMAT(10X,9A10)
 1003 FORMAT(9X,9(F5.3,5X)/44X,A1*-AXIS(FERMI)*)
      END
      SUBROUTINE JWVFNC1(X,NBT,WFJS,YLM,HNL)
*          CALCULATE THE SPIN PROJECTED SINGLE PARTICLE WAVE FUNCTIONS AT X
*          AND STORE IN THE LINEAR ARRAY WFJS. IN THE FIRST 2*NBT LOCATIONS
*          THE SPIN UP COMPONENTS ARE STORED IN THE ORDER GIVEN BY ICODAP AND
*          ICODAN. ICODAP CONTAINS THE POSITIVE PARITY STATES IN CODA AND ICODAN
*          THE NEGATIVE PARITY STATES. FOR A GIVEN J-STATE THE ORDER OF M IS
*          M=J, M=J-1,.....,  M=-J. THE TOTAL NO. OF STATES IS NBT.
*          THE IMAGINARY PART FOLLOWS IMMEDIATETLY AFTER THE REAL PART FOR EACH
*          (J,M) STATE. THE 2*NBT LOCATIONS STARTING FROM 2*NBT+1 STORE THE
*          REAL AND IMAGINARY SPIN DOWN COMPONENTS.
*
      COMMON/CLEBDAT/ALOGF(30),SQRF(30)
      DIMENSION WFJS(1),YLM(1),HNL(1),X(3)
      COMMON /CODAN/ICODAP(16),ICODAN(16),LVP,LVN,IQM(16),NOLT
      COMMON/ADEN1/DTHETA,ITHE,IRAD,MAXX
      DIMENSION CS(4),SN(4),CSL(10),SNL(10)
      COMMON/CODATPR/IBAQ(30)
      LOGICAL XX2,LTHE
      DATA CS,SN/1.,0.,-1.,0.,0.,1.,0.,-1./
      DATA PI,PI2/3.1415 92653 5898,1.5707 96326 7949/
*
*          SCG IS THE C-G COEFF. ( J,M I L,.5,M-S,S ) .   MU=2*(J-L), NU=2*S.
*          X1,X2,X3 ARE THE CARTESIAN COORDINATES X,Y,Z.
      ENTRY CWVFNC
   21 X1=X(1)$ X2=X(2)$ X3=X(3)
      R=X1*X1+X2*X2$ THE=ATAN2(SQRT(R),X3)$ PHI=ATAN2(X2,X1)
      R=SQRT(R+X3*X3)$ IF(ABS(X3).LT.1.E-5) THE=PI2
      IF(ABS(X1).LT.1.E-5) PHI=SIGN(PI2,X2)$ GOTO1
      ENTRY SWVFNC
*
*          X1,X2,X3 ARE THE POLAR COORDINATES R,THETA,PHI.
   22 R=X(1)$ THE=X(2)$ PHI=X(3)
    1 KK=SHIFT(NBT,2)$ DO11 K=1,KK
   11 WFJS(K)=0$ DO 16 L=1,10$ PHL=(L-1)*PHI$ CSL(L)=COS(PHL)
   16 SNL(L)=SIN(PHL)$  XX2=.FALSE.$ KS=2*NBT$ IF(THE-PI/2.)2,2,3
    3 THE =PI-THE$ XX2=.TRUE.
    2 LTHE=(PI*.5-THE).GT.DTHETA$ XT=THE/DTHETA$ K2=XT$ DK2=XT-FLOAT(K2)
      XR=R*MAXX/4.$ K1=XR$ DK1=XR-FLOAT(K1)$ YL=IK=1
      DO 24 I = 1 , NOLT$ IQ= IQM(I)$ IQNT= IBAQ(IQ)$ JPH= IQNT/10000
      IT= MOD(IQNT,10000)$ NT= IT/100$ L= MOD(IT,100)$ M22= SHIFT(JPH,1)
      M2= M22-1$ NQ= NT-1$  MU= M2-SHIFT(L,1)$ SQL= SQRF(2*L+1)
      MDL=MOD(L,4)+1$ SRE=CS(MDL)$ SIM=SN(MDL)$ NQ2=SHIFT(NQ,-1)
      IR=IRAD*K1+(NQ2+1)*NQ2+MOD(NQ,2)*(SHIFT(NQ-1,-1)+1)+SHIFT(L,-1)+1
```

```
      H=HNL(IR)$ H=H+DK1*(HNL(IR+IRAD)-H)$IL=SHIFT(L*(L+1),-1)+ITHE*K2+1
*
*          IR IS THE ADD. OF N,L,R IN HNL. ILM IS THE ADD. OF L,LM,THETA IN YLM.
      DO24 JM=1,M22$ XM=FLOAT(M2)/2$      DO7 NU=1,3,2$ NNU=NU-2$ YM=ZM=1
      LM=SHIFT(M2-NU+2,-1)$ IF(XX2) YL=1-2*MOD(L+LM,2)$ IF(LM)6,5,5
    6 LM=-LM$ YM=1-2*MOD(LM,2)$ ZM=-1.
    5 IF(LM.GT.L) GOTO7$ ILM=IL+LM$ IS=SHIFT(KS*(3-NU),-1)
      KK=.5+MU*NNU*XM$ SQK=SQRF(KK+L)$ SCG=.5*(1+MU-NNU+MU*NNU)*SQK/SQL
      Y=YLM(ILM)$ IF(LTHE) Y=Y+DK2*(YLM(ILM+ITHE)-Y)
      S=YM*YL*Y*H*SCG$ CR=CSL(LM+1)$ SR=SNL(LM+1)*ZM
      WFJS(IS+IK)=(SRE*CR-SIM*SR)*S$ WFJS(IS+IK+1)=(SIM*CR+SRE*SR)*S
    7 CONTINUE$ M2=M2-2
   24 IK=IK+2$  RETURN
      END
      FUNCTION DENSD(RO,WAVF,G1,G2,NB,MUD)
*
*     THIS CODE GENERATES MUD CALLS TO DENSP TO GET TOTAL DENSITY
      DIMENSION RO(1),WAVF(2,1),NB(1)
      M= NBB=1$ X= 0.$ NBT= 0$ DO 11 K= 1 , MUD
   11 NBT= NBT+ NB(K)
      DO 1 K = 1 , MUD$ ND= NB(K)
      X= X+DENSP(RO(M),WAVF(1,NBB),WAVF(1,NBB+NBT),G1,G2,ND)
      M= M+ ND**2
    1 NBB= NBB+ ND$ DENSD= X$ RETURN
      END
      FUNCTION DENSP(RO,FI1,FI2,H1,H2,NB)
*
*     ASSUMING THAT RO(I,J) IS REAL COMPUTE SPATIAL DENSITY USING WAVF .
      DIMENSION RO(NB,1),FI1(1),FI2(1)
      ND= NB$ DENSP= 0.$ II= 1$ DO 1 I = 1 , ND$ JJ= 1$ G1= 2.*H1
      G2= 2.*H2$ DO 3 J = 1 , I$ IF(ABS(RO(I,J)).LT.1.E-6) GO TO 3
      IF(I.GT.J) GO TO 4$ G1= .5*G1$ G2= .5*G2
    4 DENSP= DENSP+ RO(I,J)*(G1*(FI1(JJ)*FI1(II)+FI1(JJ+1)*FI1(II+1))
     1 +G2*(FI2(JJ)*FI2(II)+FI2(JJ+1)*FI2(II+1)))
    3 JJ= JJ+ 2
    1 II= II+2$ RETURN
      END
      SUBROUTINE LINPLT2(A,UNIT,LAY,N,X)
      COMMON/SYMB/LIT(40)$ DIMENSION X(1),A(1)$ INTEGER X
      DATA LIT/1R ,1R1,1R ,1R2,1R ,1R3,1R ,1R4,1R ,1R5,
     1         1R ,1R6,1R ,1R7,1R ,1R8,1R ,1R9,1R ,1RA,
     2         1R ,1RB,1R ,1RC,1R ,1RD,1R ,1RE,1R ,1RF,
     3         1R ,1RG,1R ,1RH,1R ,1RJ,1R ,1RK,1R ,1RL/
*     N1=N-1$ A0=A(1)$ A1=A(2)$ DO3 I=1,13
*   3 X(I)=0B$ DO4 I=2,N1$ A2=A(I+1)$ A(I)=(A0+A1+A2)/3.$ A0=A1
*   4 A1=A2$ M=(5*N)/3$M1=M-1$DEL=.5*UNIT$MS=M1/10+1$ IF(M-100)1,1,99
      DO3 I=1,9
    3 X(I)=0B$ M=(5*N)/3$M1=M-1$DEL=.5*UNIT$MS=M1/10+1$ IF(M-100)1,1,99
    1 K=(A(1)+DEL)/UNIT+1.$ X(1)=SHIFT(LIT(K),54)$ K=(A(N)+DEL)/UNIT+1.
      X(MS)=SHIFT(LIT(K),54-6*MOD(M-1,10))$ DO2 I=2,M1$ I1=I-1
      I2=I1/10$ I1=MOD(I1,10)$ XI=I2*6.+I1*5./9+1.$ IX=XI
      K=(A(IX)+(XI-FLOAT(IX))*(A(IX+1)-A(IX))+DEL)/UNIT+1. $ J=I2+1
    2 X(J)=X(J)+SHIFT(LIT(K),54-6*I1)$  RETURN
   99 PRINT1002$  RETURN
 1002 FORMAT(*  MESSAGE FROM LINPLT. N.GT.60. DECREASE NO. OF POINTS CAL
     1CULATED. N=NMAX. TO LESS THAN 60.*)
      END
      SUBROUTINE FORMT(NL,NAME)
      COMMON/PLOT3/X(3),IAXIS(3),UNIT1,XNAME(3),DA,BETA,ABSS(9),
     1    DAR(9),NPN(2),COMM(2)
      DIMENSION DAR1(6)
      EQUIVALENCE (DAR(1),DAR1(1))
```

```
      DATA DA/10H+---------/, DB/10H1---------/, DC/10HI  NUCLEUS/,
     1     DD/1HI/,           DL/2H--/,          DE/7H(FERMI)/,
     2     DF/5HPLANE/,       DG/7HI  UNIT/,     DH/7HDENSITY/,
     3     DI/9H/F CUBED /,   DJ/10HI  COMMENT/, XNAME/1HX,1HY,1HZ/.
      XX1=BETA*X(IAXIS(1))$ XX2=BETA*X(IAXIS(2))
      IF((NL-45)*(NL-54))10,10,1
    1 IF(NL.EQ.28)GOTO2$ IF(MOD(NL,6).EQ.1)GOTO3
      PRINT1001,DD,DAR,DD$  RETURN
    2 PRINT1002,XNAME(IAXIS(2)),DD,DAR,DD$  RETURN
    3 PRINT1003,XX2,DL,DAR,DL$ ABSS(NL/6+1)=XX2$  RETURN
   10 IF(MOD(NL,2))12,11,12
   11 PRINT1004,DD,DAR1,DD,DD$  RETURN
   12 NI=(NL-43)/2$ GOTO(21,22,23,24,25) NI
   21 PRINT1001,DD,DAR1,DA,DB,DB,DD$  RETURN
   22 PRINT1006,DD,DAR1,DC,NAME,NPN,DD$  RETURN
   23 PRINT1007,XX2,DL,DAR1,XNAME(IAXIS(1)),XX1,DE,DF,DL
      ABSS(NL/6+1)=XX2$  RETURN
   24 PRINT1008,DD,DAR1,DG,DH,UNIT1,DI,DD$  RETURN
   25 PRINT1009,DD,DAR1,DJ,COMM,DD$  RETURN
 1001 FORMAT(9X,A1,9A10,A1)
 1002 FORMAT(1X,A1,'-AXIS  ',A1,9A10,A1)
 1003 FORMAT(3X,F5.3,A2,9A10,A2)
 1004 FORMAT(9X,A1,6A10,A1,29X,A1)
 1006 FORMAT(9X,A1,7A10,1X,A5,'  Z=',I3,' N=',I3,1X,A1)
 1007 FORMAT(3X,F5.3,A2,6A10,'-',2X,A1,'=',F5.3,A7,1X,A5,7X,A2)
 1008 FORMAT(9X,A1,6A10,A7,1X,A7,1X,F5.3,A9,A1)
 1009 FORMAT(9X,A1,9A10,A1)
      END
      SUBROUTINE LINPRT1(RO,DD,N,83)
*          LEFT-RIGHT SYMMETRY OF RO IS NOT CONSERVED. TO CONSERVE SYMMETRY
*          USE SUBROUTINE LINPRT.
      INTEGER A(40)
      DIMENSION RO(N,N)
      DATA M/30/
      S= 100./DD$ SB3= S*B3$ PRINT 1001,SB3$ M1= M-1$ DO 2 J = 1 , N
      A(1)= RO(1,J)*S+.5
      A(M)=RO(N,J)*S+.5$ DO1 K=2,M1$ XK=FLOAT(K*N)/FLOAT(M)$ I=XK
    1 A(K)=(RO(I,J)+(XK-FLOAT(I))*(RO(I+1,J)-RO(I,J)))*S+.5
    2 PRINT1000,(A(I),I=1,M)$ RETURN
 1000 FORMAT(5X,40I3)
 1001 FORMAT('1    DENSITY PRINT. THE INTEGER PRINTED IS= (RHO IN FERM.*
     18H**(-3))*F5.1//')
      END
      OVERLAY(MULTIP,6,0)
      PROGRAM MULTIP
*
*     TO COMPUTE THE MULTIPOLE MOMENTS OF THE PROTON WAVEFUNCTIONS .
      COMMON/BASIS/ ICODE(16),ISTA(16),IPAR(16),INPO(16),INUC(16),
     1 ILC(16),NOLV,NB(2),MUD,IDF,IDSC,IDSJ,JAR(14,2),IDAC(14,14),JMPO,
     2 NBT,HBOM,OSPA2
      COMMON AS1(4000),AS2(4000),AS3(1)
      CALL TIMEPRT(YD)
      PRINT 103
      CALL BUFFIN (4LROPS,AS1,IDSJ)
      DO 1 KPOLL= 1 , 7 , 2$ KPOL= KPOLL-1$ DO 1 MPOLL= 1 , KPOLL,2
      MPOL= MPOLL-1
      CALL MULME(AS2,KPOL,MPOL)
      CALL ROCONV(AS2,AS3,+1)
      HEXAD= TRVEROJ(AS3,AS1,0)*(1-2*MOD(KPOL/2,2))
*     DHEXAD=H**L*(4*PI/2*L+1)**.5*(Y(L,M)+(-1)**M*Y(L,-M))/2,IN F**L
      DHEX= HEXAD*(OSPA2)**(KPOL/2)
    1 PRINT 101,KPOL,MPOL,HEXAD,DHEX
```

```
 101  FORMAT(/* FOR KPOL,MPOL=*2I4,* ,MOMENT=*1PE14.5,* ,IN UNITS OF BSQ
     1R, MOMENT=*E14.5,* ,IN UNITS OF *7HF**KPOL)
 103  FORMAT(////52X*COMPUTATION OF MULTIPOLE MOMENTS*)
      END
      SUBROUTINE MULME(RME,KPOL,MPOL)
*
*     TO COMPUTE THE MATRIX ELEMENTS OF (I*R)**KPOL*C(KPOL,MPOL), IN THE
*     T.R.I. BASIS, WHERE THEY ARE REAL. C(KPOL,MPOL)=(4*PI/(2*K+1))
*     **.5*YLM. IF MPOL.NE.0 COMPUTE .5*(C(K,M)+(-1)**(K-M)*C(K,-M)) SINCE
*     THIS IS THE REAL SYMMETRIC HERMITIAN PART .
      COMMON/BASIS/ ICODE(16),ISTA(16),IPAR(16),INPO(16),INUC(16),
     1 ILC(16),NOLV,NB(2),MUD,IDF,IDSC,IDSJ,JAR(14,2),IDAC(14,14),JMPO,
     2 NBT,HBOM,OSPA2
      DIMENSION RME(1)
      IHA= 1-SHIFT(IABS(KPOL-MPOL).A.1,1)$ RKPO= KPOL
      CNOR= 1./SQRT(1.+2*KPOL)$ DO 2 KK = 1 , IDSC
 2    RME(KK)=0 $ ILS= 0$ DO 1 IPL= 2 , 3$ ND= NB(IPL-1)$ ILAP=0
      DO 3 KP = 1 , NOLV$ IF(((IPL+IPAR(KP)).A.1).NE.0) GO TO 3
      JPHP= ISTA(KP)$ MLMP= SHIFT(JPHP,1)$ LP= ILC(KP)
      NUP1= SHIFT(INPO(KP)-LP+1,-1)$ JMLP2= MLMP-SHIFT(LP,1)-1$ILA= 0
      DO 7 K = 1 , KP  $ IF(((IPL+IPAR(K)).A.1).NE.0) GO TO7$JPH=ISTA(K)
      MLM= SHIFT(JPH,1)$ L= ILC(K)$ IF(L+LP-KPOL.LT.0) GO TO 8
      JML2= MLM-SHIFT(L,1)-1$ NU1= SHIFT(INPO(K)-L+1,-1)
      IPE= IABS(L-LP+KPOL)
      RMX= CNOR*RAD(NUP1,LP,NU1,L,KPOL)*RMECK(LP,JMLP2,KPOL,L,JML2)*
     1 (1-SHIFT(SHIFT(IPE,-1).A.1,1))$ IF(RMX.EQ.0) GO TO 8
      DO 9 MLP= 1 , MLMP$ ILBP= ILAP+MLP$ RFASM= .5
      DO 11 ML= 1 , MLM$ ILB= ILA+ML$ IF(ILB.GT.ILBP) GO TO 11
      MPMM= ML-MLP+JPHP-JPH$ C1=C2=0
      IF(MPOL.EQ.MPMM) C1= CLEBSCH(JPHP-.5,JPH-.5,JPHP+.5-MLP,ML-.5-JPH,
     1 RKPO)
      IF(MPOL.EQ.-MPMM) C2= CLEBSCH(JPHP-.5,JPH-.5,JPHP+.5-MLP,ML-.5-JPH
     1 ,RKPO)*IHA
      RME(ILS+ILBP+ND*(ILB-1))= RFASM*(C1+C2)*RMX
 11   RFASM= -RFASM
 9    CONTINUE
 8    ILA= ILA+ MLM
 7    CONTINUE$ ILAP= ILAP+ MLMP
 3    CONTINUE
 1    ILS= ILS+ ND*ND$ RETURN
      END
      FUNCTION RMECK(LP,JMLP,K,L,JML)
*
*     TO COMPUTE THE REDUCED MATRIX ELEMENTS OF THE UNNORMALISED
*     SPHERICAL HARMONICS C(K), IN J-J COUPLING BASIS, USING
*     FORMULAE 7.1.9 AND 7.1.10 OF EDMUNDS P 113.
*     JMLP AND JML CAN BE ANY REAL OR INTEGER NUMBERS HAVING THE SIGN OF
*     JP-LP AND J-L .THE MOSHINSKY PHASE FOR YLM IS USED THROUGHOUT AND
*     EDMONDS EXPRESSION FOR THE WIGNER-ECKART THEOREM IS ASSUMED .
*     /ALGFN/ MUST HAVE BEEN PRESET TO LOG(GAMMA(N)) .
      COMMON /CLEBDAT/ALOGF(30),SQRF(30)
      COMMON/RMECKOT/I1,I2,I3,I4,I5,I6,I7,I8,IFASN,RJP,RJ
      VAL=0.$ IF(MOD(L+LP+K,2)) 99,1
 1    RSH=1.$ RK=K$ IF(JMLP) 3,99,5
 3    RJP= FLOAT(LP)-.5$ IF(JML) 7,99,9
 5    RJP= FLOAT(LP)+.5$ IF(JML) 11,99,13
 7    RJ= FLOAT(L)-.5$ GO TO 15
 9    RJ= FLOAT(L)+.5$ GO TO 17
 11   RJ= FLOAT(L)-.5$ GO TO 17
 13   RJ= FLOAT(L)+.5
 15   RSH=0.
 17   SM= RJ+RJP+RK$ IFASN=2- 4*MOD(IABS(IFIX(.5*(RJ-RJP+RK+RSH))),2)
```

```
      I1=RJ+RJP-RK+1$ I2= RJP-RJ+RK+1$ I3=RJ+RK-RJP+1$ I4=IFIX(SM)+2
      I5= .5*(SM+3.-RSH)$ I6= .5*(RJP+RJ-RK+1.+RSH)
      I7= .5*(RJ+RK-RJP+2.-RSH)$ I8= .5*(RJP-RJ+RK+2.-RSH)
      I0= MIN0(I1,I2,I3,I4,I5,I6,I7,I8)$ IF(I0.LT.1) GO TO 99
      VAL= IFASN*EXP(.5*(ALOGF(I1)+ALOGF(I2)+ALOGF(I3)-ALOGF(I4))
     1 + ALOGF(I5)- ALOGF(I6)- ALOGF(I7)- ALOGF(I8))
 99   RMECK= VAL$ RETURN
      END
      FUNCTION RAD1(NQ,LQ,NR,LR,LAM)
*
*     USE THE FORMULA OF MORSE-FESHBACH P. 785 WRITTEN IN TERMS OF
*     GAMMA FUNCTIONS.
*     ALGA(R)= LOG(ABS(GAMMA(R))), R= 2*K FAGA(R)= SIGN(GAMMA(R)).
*     IF R= NEG. INT. ALGA,FAGA CONTAIN THE LOG ABS OF THE RESIDUE AT
*     THE POLE, AND THE SIGN OF THE RESIDUE RESPECTIVELY .
*     ALOGF(N) MUST HAVE BEEN PRESET TO LOG(GAMMA(N)) .
*     NQ= NUP+1$ NR= NU+1
*
      COMMON /CLEBDAT/ALOGF(30),SQRF(30)
      ENTRY RAD
      LMPD= LAM+ LQ+ LR$ RLMP= .5*LMPD$ IFAS= 1-2*MOD(NQ+NR,2)
      NP= NQ-1$ N= NR-1$ VAL=0.$ LL= MIN0(NP,LQ,N,LR,LAM+1)
      IF(LL.LT.0) GO TO 100$ RNLP= NP+ LQ$ RNL= N+ LR
      ISU= MIN0(N,NP)+1$ IF(MOD(LMPD,2)) 1, 3
 1    ISL=1$ GO TO 5
 3    LMLP=IFIX(RLMP)-LQ $ LML= IFIX(RLMP)-LR$ ISL=1
      IF(LMLP.GE.0) ISL=1+MAX0(0,NP-LMLP)
      IF(LML.GE.0) ISL= 1+ MAX0(ISL-1,N-LML)
 5    IF(ISU-ISL) 99,7,7
 7    VAL= IFAS*EXP(.5*(ALOGF(NQ)+ ALOGF(NR)-ALGA(RNL+1.5)-ALGA(RNLP
     1 + 1.5))+ ALGA(RLMP-FLOAT(LQ-1))+ALGA(RLMP-FLOAT(LR-1)))*
     2 FAGA(RLMP-FLOAT(LQ-1))*FAGA(RLMP-FLOAT(LR-1))
      SR=0.$ R1=RLMP+1.5$ R2= RLMP-RNLP+1.$ R3= RLMP-RNL+1.
      DO 9 IS= ISL,ISU$ RS= IS-1
 9    SR= SR+EXP(-ALOGF(RS+1.)-ALOGF(FLOAT(N-IS+2))-ALOGF(FLOAT(NP-IS
     1 +2))+ ALGA(R1+RS)-ALGA(R2+RS)-ALGA(R3+RS))*
     2 FAGA(R1+RS)*FAGA(R2+RS)*FAGA(R3+RS)
      VAL= VAL*SR
 99   RAD1=VAL$ RETURN
 100  PRINT 102, NQ,LQ,NR,LR,LAM
 102  FORMAT( * BAD CALL TO RAD  NQ,LQ,NR,LR,LAM= *5I5,
     1 *LOWER LIMITS ARE  1, 0, 1, 0, -1 *)
      GO TO 99
      END
      FUNCTION ALGA(RN)
*     TO COMPUTE LOG(ABS(GAMMA(RN))) , USING /ALGFN/ WHIVH MUST HAVE
*     BEEN FRESET TO LOG(GAMMA(N)) .
*     RN MUST BE + K/2 OR -(K+.5)
*     THE ENTRY FAGA RETURNS WITH THE PHASE OF GAMMA(-K-.5) FOR K
*     NEGATIVE .
*     NONSENSE MAY       BE RETURNED  IF RN.NE. 2*K .
*     IF RN=-N,N=0,1,... ALGA= LOG(ABS(RESIDUE AT THE POLE)),
*     FAGA= SIGN(RESIDUE)
*
      COMMON /CLEBDAT/ALOGF(30),SQRF(30)
      DATA AL2/0.6931 47180 55994/,ALRP/0.5723 64942 92470/
*
      ITN= INT(RN+ RN)$ IF(ITN) 1,3,5
 3    ALGA= -ALOGF(IFIX(-RN)+1)
 17   RETURN
 5    IF(MOD(ITN,2)) 9,7
 7    ALGA= ALOGF(IFIX(RN))$ RETURN
```

```
 9      IF(ITN-1) 13,11
 11     ALGA= ALRP$ RETURN
 13     ALGA= ALRP+ ALOGF(ITN-1)- AL2*(ITN-2)-ALOGF(IFIX(RN-.5))$RETURN
 1      IF(MOD(ITN,2)) 15,3
 15     ALGA= ALRP-AL2*ITN+ ALOGF(IFIX(.5-RN))-ALOGF(1-ITN)$RETURN
        ENTRY FAGA$ ALGA=1$ IF(RN) 19,17,17
 19     ALGA= 1-2*MOD(IFIX(.5-RN),2)$ RETURN
        END
        FUNCTION CLEBSCH(VJ1,VJ2,VM1,VM2,VJ3)
*
*       TO COMPUTE C(J1,J2,J3/M1,M2) , A CLEBSCH-GORDAN COEFFICIENT
*       USING THE FORMULA OF SCHWINGER(REF. QUANTUM THEORY OF ANG. MOM.,
*       BY BIEDENHARN,P246), AND THE REGGE SYMMETRIES(REF.IBID,P.296) .
        COMMON /CLEORD/ ZN(1),VN(3,3),NR(1),NN(3,3)
        COMMON /CLEBDAT/ALOGF(30),SQRF(30)
*
*       PREPARE THE REAL REGGE Q.N. IN VN(I,J) .
 301    RES= 0.$ ZN(2)= VJ1-VJ3+VJ2$ ZN(6)= VJ1-VM1$ ZN(7)= VJ1+VM1
        ZN(9)= VJ2-VM2$ ZN(10)= VJ2+VM2$ RJ= VJ1+VJ2+VJ3
        ZN(3)= RJ-ZN(6)-ZN(9)$ ZN(4)= RJ-ZN(7)-ZN(10)
        ZN(5)= RJ-ZN(6)-ZN(7)$ ZN(8)= RJ-ZN(9)-ZN(10)$ JJ= MOD(INT(RJ),2)
*
*       CHECK THAT THEY ARE NON-NEG. INTEGERS AND FIND NZE.
        NZE= 101$ DO 1 I = 1 , 3$ DO 1 J = 1 , 3$ NT= IABS(INT(VN(I,J)))
        IF(FLOAT(NT).NE.VN(I,J)) GO TO 99$ IF(NT.GE.NZE) GO TO 1
        NZE= NT$ IZE= I$ JZE= J
 1      NN(I,J)= NT$ IFSE= NR(7)+ NR(9)
*
*       C-G EXISTS, BRING NZE TO NN(1,1) POSITION, SAVE THE PHASE .
        IF(IZE.EQ.1) GO TO 3$ IF(JJ.NE.0) IFSE= IFSE+1$ DO 5 J = 1 , 3
        NT= NN(1,J)$ NN(1,J)= NN(IZE,J)
 5      NN(IZE,J)= NT
 3      IF(JZE.EQ.1) GO TO 7$ IF(JJ.NE.0) IFSE=IFSE+1$ DO 9 I = 1 , 3
        NT= NN(I,1)$ NN(I,1)= NN(I,JZE)
 9      NN(I,JZE)= NT
*
*       CONSTRUCT DELTA*NORMALISATION .
 7      IFAS= 1-2*MOD(IFSE+NR(3)+NR(5),2)
        SUM= -ALOGF(INT(RJ)+2)$ DO 11 K = 2 , 10
 11     SUM= SUM+ ALOGF(NR(K)+1)$ SUM= .5*SUM
*
*       DO THE SUM WITHOUT ALOGF .
        SUM= SUM-ALOGF(NR(2)+1)- ALOGF(NR(6)+1)- ALOGF(NR(10)+1)
       1 -ALOGF(NR(9)-NR(2)+1)- ALOGF(NR(7)-NR(2)+1) $ PROD= EXP(SUM)
        SUM= 1.$ NMX= NR(2)$ IF(NMX.LT.1) GO TO 13
        ZA= NR(6)- NMX$ ZB= NR(10)-NMX$ ZC= NMX+1$ ZD= NR(9)+1
        ZE= NR(7)+1$ DO 15 K = 1 , NMX$ ZK= K
 15     SUM=1.-SUM*ZK*(ZA+ZK)*(ZB+ZK)/((ZC-ZK)*(ZD-ZK)*(ZE-ZK))
 13     RES= SUM*PROD*IFAS*SQRT(2.*VJ3+1.)
 99     CLEBSCH= RES$ RETURN
        END
END EVALIN

        BLOCK DATA EVAPREP
C
C            THE SHELL STRUCTURE IS 0S1/2,0P3/2,/P1/2,0D5/2,1S1/2,
C            0D3/2,0F7/2,1P3/2,0F5/2,1P1/2,0G9/2,1D5/2,0G7/2,2S1/2,
C            1D3/2,0H11/2,0H9/2,1F7/2,1F5/2,2P3/2,2P1/2,0I13/2,1G9/2,
C            0I11/2,2D5/2,3S1/2,1G7/2,2D3/2,0J15/2,0J13/2.
C
        COMMON/SHELL/NX(30),LX(30),XJX(30)
        COMMON/UPLOWA/IMIN,IMAX,I1MIN,I1MAX,I2MIN,I2MAX,I3MIN,I3MAX,
```

```
     1I4MIN,I4MAX,JMIN,JMAX,ICODE
      COMMON/NUMB/NUM,NUMIN,IPARITY
      COMMON/IPRI/ IPRINT$ LOGICAL IPRINT
      COMMON /TCON/NAME,HBWA(30),IUNIT(30),IADV(30)
*
*     DATA NAME,HBWA/10H234-016,6L ,13.5,10.5,13.5,10.5,0/
*     DATA IUNIT/30*5/                                                    234-016
*     DATA IADV/30*1/                                                     234-016
*     DATA NAME,HBWA/10H021SP2,6LV,13.5,17.5,20.0,10.5,25.0,8.0,30.0,
      DATA NAME,HBWA/10H020SP2,16L,13.5,17.5,20.0,10.5,25.0,8.0,30.0,
     1 15.0,0.00/
      DATA IUNIT/5,7*6/
      DATA IADV/3,7*1/
*
      DATA NX,LX,XJX/
     1 4*0,1,2*0,1,0,1,0,1,0,2,0,1,0,1,0,2,1,2,0,1,0,2,1,3,2,0,
     20,2*1,2,0,2,3,1,3,1,4,2,4,0,5,2,5,3,6,1,3,1,6,4,7,2,4,0,2,7,
     30.5,1.5,0.5,2.5,0.5,1.5,3.5,1.5,2.5,0.5,4.5,2.5,3.5,0.5,5.5,
     41.5,4.5,3.5,6.5,1.5,2.5,0.5,5.5,4.5,7.5,2.5,3.5,0.5,1.5,6.5/
C
*     ICODE=0, NO GA PRINTOUT. ICODE=2, FOR GA MATR. PRINTOUT.
      DATA IMAX,ICODE/16,2/
*
      DATA NUM,NUMIN,IPARITY/0,0,1/
*
      DATA IPRINT/.T./
      END
      PROGRAM EVAFME(OUTPUT,TAPE1,TAPE2,TAPE5,TAPE9=1,TAPE6)
      DIMENSION GA(15000),MAGIC(8),ICODE(16),GPH(2),FCL(5),TLINT(150)
      COMMON /TCON/NAME,HBWA(30),IUNIT(30),IADV(30)
      COMMON/SHELL/NX(30),LX(30),XJX(30)
      COMMON/UPLOWA/IMIN,IMAX,I1MIN,I1MAX,I2MIN,I2MAX,I3MIN,I3MAX,
     1I4MIN,I4MAX,JMIN,JMAX,IPUNCH
      COMMON/LBL3/IPH(30,30),IJTV(10,2),IIP(100),IIH(100)
      COMMON/LBL4/IPMIN,IPMAX,IHMIN,IHMAX,IPHD,IJTD,JJMAX
      COMMON/MOSHTN/ NOEN,TABM(1000),NQMS(1000)
      COMMON/IPRI/ IPRINT$ LOGICAL IPRINT
      COMMON/NUMB/NUM,NUMIN,IPARITY
      COMMON/PHGCON/NMAXP,NOENP,IPASS$ LOGICAL IPASS
      DATA MAGIC,MAG/2,8,20,28,40,50,82,126,9/
      DATA NTYP/2/
      CALL ERRSET(IERROR,50)
      REWIND 2$ REWIND 5$ REWIND 6
      PRINT 101
  101 FORMAT(1HR)
      IPMIN=IHMIN=I1MIN=I2MIN=I3MIN=I4MIN=IMIN=1
      IPMAX=IHMAX=I1MAX=I2MAX=I3MAX=I4MAX=NV=IMAX
      J1P1=1$ DO 21 KP= 1 , NV$ K= KP+IMIN-1
   21 J1P1= MAX0(J1P1,IFIX(2.*XJX(K))+1)
      CALL TALMI(TLINT,NTYP)
      PRINT1
    1 FORMAT(1H0,5X,45H***************SHELL STRUCTURE***************
     1//18X,*N*,7X,*L*,7X,*J*,5X,*NOSC=2N+L */)
      DO 3 I=1,30
      J=31-I
      NOSC=2*NX(J)+LX(J)
      IF(J.NE.22.AND.J.NE.16.AND.J.NE.11.AND.J.NE.10.AND.J.NE.7
     1.AND.J.NE.6.AND.J.NE.3.AND.J.NE.1) GO TO 3
      MAG=MAG-1
      PRINT2,MAGIC(MAG)
    2 FORMAT(18X,38H************************************** ,I3)
    3 PRINT1002,J,NX(J),LX(J),XJX(J),NOSC
```

```
1002 FORMAT(5X,I2,I12,I8,F9.1,I10)
1003 FORMAT(8X,012,5E20.10,I2)
     DO 49 IBL= 1 , 31$ IF(IBL.EQ.1) GO TO 23$ IB= IBL-1$ WH=HBWA(IB)
     NMAXP=NOENP=0$ IPASS= .F.
     IF(WH.EQ.0) GO TO 99$ CALL PREPSA(IADV(IB),0,IUNIT(IB))
23   REWIND 1$ WRITE(1,1010) NAME
1010 FORMAT(A10)
     IF(IBL.GT.1) CALL LABELF(5,GA)
     JF=JG=F3=F4=F5=IFAS= 0$ F1=F2=NV
     PRINT1003,JF,F1,F2,F3,F4,F5,IFAS
     WRITE(1,1003)JF,F1,F2,F3,F4,F5,IFAS
     DO 7 KP = 1 , NV$ K = KP+IMIN-1$ F1=F2=K$ F3=F4=F5= 0.
     ICODE(KP)= K
     PRINT1003,JF,F1,F2,F3,F4,F5,IFAS
     WRITE(1,1003)JF,F1,F2,F3,F4,F5,IFAS
   7 CONTINUE
     IFL= NV+2
     IPI= 1$ DO 4 J1= 1 , J1P1$ J= J1-1$ VJ= J$CALL LABEL3(IPI,J,IDM,1)
     PRINT 1007$ ICNT= 0
1007 FORMAT    (///* THE FIRST 100 F MATRIX ELEMENTS FOR THIS J.*/)
     F1=F2=64*J+IDM$ F3=F4=F5=JF=0
     PRINT1003,JF,F1,F2,F3,F4,F5,IFAS
     WRITE(1,1003)JF,F1,F2,F3,F4,F5,IFAS
     IFL= IFL+1$ IF(IDM.LT.1) GO TO 4
     DO 5 M=1,IDM
        IC=IIP(M)
        ID=IIH(M)
     DO 5 N=M,IDM
        IA=IIP(N)
        IB=IIH(N)
*       JF=ID+64*(IC+64*(IB+64*(IA    +64*64*J)))                          T=0
        JF=ID+64*(IC+64*(IB+64*(IA+128+64*64*J)))                          N=N
        IF(MOD(LX(IA)+LX(IB)+LX(IC)+LX(ID),2))12,11,12
  12    F1=F2=0.$ GO TO 13
*    F1= FS, F2= FD, F3= FCS,
 11  IF(IBL.EQ.1) GO TO 111
     CALL PHGPHC(IA,IB,IC,ID,J,GPH,GA)$ F1= .5*(GPH(1)-GPH(2))
     F2= .5*(GPH(1)-GPH(2))$ F3=F4=F5=IFAS=0.$ GO TO 13
 111 F1=F2=0.
     CALL FMTRX1(IA,IB,IC,ID,J,TLINT,FCL,NTYP,IFAS)
     F3= FCL(3)$ F4= FCL(4)$ F5= FCL(2)
*       F3=FCL(3)*E2B$ F4=FCL(4)$ F5=FCL(2)
*        FCL(1) IS COUL DIRECT, FCL(3) IS COUL DRCT-EXCH,
*        FCL(2) IS R**2 DRCT, FCL(4) IS R**2 DRCT-EXCH.
*    E2B=SQRT(.0241145*WH)*1.44                                            E**2/B
 13  ICNT= ICNT+1$ IF(ICNT.LT.101.A.IPRINT)
    1          PRINT1003,JF,F1,F2,F3,F4,F5,IFAS
 15  WRITE(1,1003)JF,F1,F2,F3,F4,F5,IFAS
     IFL= IFL+1
   5    CONTINUE
   4    CONTINUE
     ENDFILE 1
     IFL= IFL-1
     PRINT 1011, IFL
1011 FORMAT (//* NUMBER OF F MATRIX ELEMENTS COMPUTED*I6)
     PRINT2004,NOEN
2004 FORMAT(/*    NUMBER OF TALMAN COEFF. TABULATED=*I5///)
     CALL REDFA12(GA,NV,ICODE,NAME,IFL,15000,NUM,WH,IBL)
 49  CONTINUE
 99  ENDFILE 2
     REWIND 2$ REWIND 5$ REWIND 6
     PRINT 777,IERROR
```

```
 777  FORMAT(* NUMBER OF INPUT ERRORS= *I3)
      END
      SUBROUTINE TALMI(TLINT,N)
      DIMENSION TLINT(1)
          TLINT(1)=SQRT(2./3.141592653)
          TLINT(31)=3.$              TLINT(61)=1./SQRT(8.)
      DO 1 I=2,30$ XI=I-1$ TLINT(I)=XI*TLINT(I-1)/(XI+.5)
          TLINT(30+I)=(XI+1.5)*TLINT(29+I)/(XI+.5)
          TLINT(60+I)=TLINT(59+I)/2.
    1 CONTINUE$ NN=N*30$ PRINT1006,(TLINT(I),I=1,NN)$   RETURN
 1006 FORMAT(/* TALMI INTEGRALS*//1P(E17.5,9E12.5))
      END
      SUBROUTINE PPMEXT(J1,J2,J3,J4,KJ,KT,GGG,IPR)
*
*     TO OBTAIN FROM MASS STORAGE PREPARED BY PREPSA, THE MATRIX
*     ELEMENTS(J1,J2,KJ,KT/GGG/ J3,J4,KJ,KT) .
*     THE SAUNIER MES ARE UNNORMALISED AND USE THE BROD. MOSH. PHASE .
      COMMON /SHELL/ NX(30),LX(30),XJX(30)
      COMMON/TEMP/ IQNA(250),ZA(250),IDIM,IDIM2
      COMMON /INDN/ INDEX(71),IFIRS(71),IMS
      DATA ICBL/0/
      GGT=0.$ IFAS= IABS(LX(J1) + LX(J2)- LX(J3)- LX(J4))/2
      IA= MAX0(J1,J2)$ IB= MAX0(J3,J4)$ IF(IA- IB) 20,11,10
 11   IA= MIN0(J1,J2)$ IB= MIN0(J3,J4)$ IF(IA-IB) 20,10,10
 20   I1= J3$ I2= J4$ I3= J1$ I4= J2$ GO TO 15
 10   I1= J1$ I2= J2$ I3= J3$ I4= J4
 15   IF(I2.LE.I1) GO TO 30
      IA= I1$ I1= I2$ I2= IA$ IFAS= IFAS+ IFIX(XJX(I1)+ XJX(I2))+ KJ+ KT
 30    IF(I4.LE.I3) GO TO 40
      IA= I3$ I3= I4$ I4= IA$ IFAS=IFAS+IFIX(XJX(I3)+XJX(I4))+KJ+KT
 40   I= I4+ (I3*(I3-1))/2+ ((I2-1)*(I1-1)*I1+ I2))/2
     1 + ((I1-1)*I1*(2-I1+ I1**2))/8
      IQN= SHIFT(I,5)+ SHIFT(KJ,1)+ KT$     IF(IQN.LT.1) GO TO 99
*
*     NOW FIND POSITION OF IQN IN BLOCKS .
      LM= IMS-1$ DO 50 L = 1 , LM$ IF(IQN.LT.IFIRS(L+1)) GO TO 51
 50   CONTINUE $ GO TO 99
 51   IF(L.EQ.ICBL) GO TO 52
      ICBL= L$ CALL READMS(9,IQNA,IDIM2,L)
 52   IR= INBINS(IQNA,1,IDIM,IQN)$ IF(IR.LT.1) GO TO 99
      GGT= (1- 2*MOD(IFAS,2))*ZA(IR)
 99   GGG= GGT$ RETURN
      END
      SUBROUTINE LABEL3(IPARITY,IJ,IDIM,ISYM)
C
C          IF IPARITY = 1,      TWO PARTICLE STATES HAVE POSITIVE PARITY,
C          IF IPARITY =-1,      TWO PARTICLE STATES HAVE NEGATIVE PARITY,
C          IF IPARITY = 0,      NO RESTRICTION ON PARITY.
C
      COMMON/SHELL/NX(30),LX(30),XJX(30)
      COMMON/LBL3/IPH(30,30),IJTV(10,2),IIP(100),IIH(100)
      COMMON/LBL4/IPMIN,IPMAX,IHMIN,IHMAX,IPHD,IJTD,JMAX
      VJ=IJ
      DO 1 I=1,30
      DO 1 J=1,30
    1 IPH(I,J)=.0
      DO 2 I=1,10
      DO 2 J=1,2
    2 IJTV(I,J)=.0
      N=0$ DO 7 I=1,2$ DO 7 IH=IHMIN,IHMAX
      IF(MOD(LX(IH),2).NE.I-1)GOTO7$ IPMINP=IPMIN$ IF(ISYM.EQ.1)GOTO6
      IF(IHMIN.EQ.IPMIN.AND.IHMAX.EQ.IPMAX) IPMINP=IH
```

```
    6 DO 3 IP=IPMINP,IPMAX
      IF(IJ.LT.0) GOTO4
      IF(XJX(IP)+XJX(IH).LT.VJ.OR.ABS(XJX(IP)-XJX(IH)).GT.VJ) GOTO3
    4 IF(IPARITY.EQ.0) GOTO5
      IF((-1)**(LX(IP)+LX(IH)).NE.IPARITY) GOTO3
    5 N=N+1
      IPH(IP,IH)=N
      IIH(N)=IH
      IIP(N)=IP
    3 CONTINUE
    7 CONTINUE$ IDIM=N
      IF(IPASS.NE.125394) PRINT1002,((IPH(J,I),J=1,30),I=1,30)
      IPASS=125394
 1002 FORMAT(30I3)
      RETURN
      END
      FUNCTION RACAH(A,B,C,D,E,F)
*     FUNCTION RACAT(A,B,C,D,E,F)
*
*     TO COMPUTE RACAH COEFFICIENTS USING THE REGGE SYMMETRIES
*     AND TO BUFFER THE PREVIOUSLY COMPUTED VALUES .
      LOGICAL INITIAL
      COMMON/BUFFN/IQN(1000),SIXJT(1000),NOTA,IOVF,IR,ICU
      COMMON /FACLN/FACLO(100)
      COMMON /ORDTRK/ IZ(1),IX1,IX2,IX3,IY1,IY2,IY3,IY4,
     1 RV(1),X1,X2,X3,Y1,Y2,Y3,Y4,SIXJ
      DATA INITIAL/.TRUE./,ICU,NOTA/2*0/,IOVF/500/
      SIXJ=0.$ IFAS=0
      X1=A+B+C+D$ X2=A+D+E+F$ X3= B+C+E+F
      Y1= A+B+E$ Y2= C+D+E$ Y3= A+C+F$ Y4= B+D+F
*
*     FIRST CHECK FOR ERRORS IN CALLING SEQUENCE
      DO 1 K = 2 , 8
      Z0=RV(K)$ IT= ABS(Z0)$ IF(FLOAT(IT).NE.Z0) GO TO 999
 1    IZ(K)= IT
*
*     THEN SORT THE REGGE QUANTUM NUMBERS.
      IFAS=IX1$ IT= IX1$ IE=2$ IF(IX1-IX2) 25,23,23
 25   IT= IX2$ IE=3
 23   IF(IT-IX3) 29,29,27
 27   IZ(IE)= IX3$ IX3= IT
 29   IF(IX1-IX2) 31,31,33
 33   IT= IX1$ IX1=IX2$ IX2= IT
 31   IX0= IX1$ IT= IY1$ IE= 5$ IF(IY1-IY2) 35,37,37
 35   IT= IY2$ IE= 6
 37   IF(IT-IY3) 39,41,41
 39   IT= IY3$ IE= 7
 41   IF(IT-IY4) 45,45,43
 43   IZ(IE)= IY4$ IY4= IT
 45   IT= IY1$ IE=5$ IF(IY1-IY2) 47,49,49
 47   IT= IY2$ IE=6
 49   IF(IT-IY3) 53,53,51
 51   IZ(IE)= IY3$ IY3= IT
 53   IF(IY1-IY2) 57,57,55
 55   IT= IY1$ IY1=IY2$ IY2= IT
 57   IY0= IY4
*
*     NEXT TEST FOR TRIANGLE CONDITION .
      IF(IX0.LT.IY0) GO TO 999 $ IF(IX0.LT.1) GO TO 997
*
*     DO A TABLE SEARCH .
      IQ= IX0-IY0$ DO 59 K= 3 , 8
```

```
 59   IQ= SHIFT(IQ,6)+ IZ(K)
      IR= INBINS(IQN,1,NOTA,IQ)$ IF(IR.LT.1) GO TO 61
      SIXJ= SIXJT(IR) $ GO TO 999
*
*     SO,  COMPUTE DEL3
 61   IF(.NOT.INITIAL) GO TO 3$ INITIAL= .FALSE.$ FACLO(1)= 0.
      DO 5 K= 2 , 100
 5    FACLO(K)= FACLO(K-1)+ ALOG(FLOAT(K-1))
 3    DEL3=0.$ CN= FACLO(2+IY0)$ DO 7 K = 1 , 4
      DEL3= DEL3- FACLO(2+IZ(K+4))$ CN= CN-FACLO(1+IY0-IZ(K+4))
      DO 7 L = 1 , 3 $ DEL3= DEL3+ FACLO(1+IZ(L+1)- IZ(K+4))
 7    CONTINUE
*
*     THEN INITIALISE THE SUM.
      DO 9 I= 2 , 4
 9    CN= CN-FACLO(1+IZ(I)-IY0)
      DEL3CN= (1-2*MOD(IY0,2))*EXP(CN+ .5*DEL3)
      SM=1.$ IRAN= IX0-IY0+1$ IF(IRAN.LT.2) GO TO 13
      Z0=IX0+3$ NX0=IX0+1$ X1=IX1-NX0$ X2= IX2-NX0$ X3= IX3-NX0
      NX0= IX0+2$ Y1= NX0-IY1$ Y2= NX0-IY2$ Y3=NX0-IY3$ Y4=NX0-IY4
*
*     THE SUM IS DONE RECURSIVELY WITHOUT FACLO
      DO 15 K = 2 , IRAN$ XK= K
 15   SM= 1.-SM*(Z0-XK)*(X1+XK)*(X2+XK)*(X3+XK)/
     1 ((Y1-XK)*(Y2-XK)*(Y3-XK)*(Y4-XK))
*
*     SIXJ IS THE USUAL 6-J SYMBOL OF WIGNER .
 13   SIXJ= DEL3CN*SM
*
*     SAVE THIS VALUE IN ICN .
      NOTA= MIN0(NOTA+1,1000)$ ICU= ICU+1$ ICN= NOTA
      IF(NOTA.EQ.1000) ICN= IOVF$ IQN(ICN)= IQ$ SIXJT(ICN)= SIXJ
      CALL SORTAG1(IQN,1,NOTA,SIXJT,ICN)
      GO TO 999
 997  SIXJ=1.
 999  RACAH= SIXJ*(1-2*MOD(IFAS,2))
*999  RACAT= SIXJ*(1-2*MOD(IFAS,2))
      RETURN
      END
      SUBROUTINE FMTRX1(IA,IB,IC,ID,J,TLINT,FCL,NTYP,IFAS)
*
*          ROUTINE CALCULATES THE ANTI-SYMMETRIZED F-MATRIX ELEMENTS
*          UTILIZING THE SUBROUTINE FMTRX.
*          FCL(I) I=1, ... ,NTYP, STORES THE DIRECT M.E.,
*          FCL(K), K=NTYP+1, ... ,2*NTYP, STORES THE DIRECT MINUS EXCHANGE M.E.
*
      COMMON/SHELL/NX(30),LX(30),XJX(30)
      DIMENSION FCL(1),TLINT(1),FCL1(10)
          VA=XJX(IA)$ VB=XJX(IB)$ VC=XJX(IC)$ VD=XJX(ID)$ VJ=J
      IFAS= 1-2*MOD(NX(IA)+NX(IB)+NX(IC)+NX(ID)+IABS(LX(IA)+LX(ID)
     1      -LX(IB)-LX(IC))/2,2)
          JM=AMIN1(VA+VC,VB+VD)+1.$ JN=AMAX1(ABS(VA-VC),ABS(VB-VD))+1.
          JN=JN+MOD(LX(IA)+LX(IC)+JN+1,2)$       I=VA+VD$ PH=(-1)**(I+J)
          CALL FMTRX(IA,IB,IC,ID,J,TLINT,FCL,NTYP)$  DO 1 K=1,NTYP
     1    FCL(K+NTYP)=FCL(K)
      DO 2 JX=JN,JM,2$ JP=JX-1$ VJP=JP
          X=  (VJP+VJP+1.)*(-1)**JP*RACAH(VA,VB,VC,VD,VJ,VJP)
          CALL FMTRX(IA,IC,IB,ID,JP,TLINT,FCL1,NTYP)
      DO 2 K=1,NTYP$ NK=K+NTYP
     2    FCL(NK)=FCL(NK)-X*FCL1(K)*PH$  RETURN
      END
      SUBROUTINE FMTRX(I1,I2,I3,I4,J,TLINT,FCL,NTYP)
```

```
*
*       FUNCTION GENERATES THE DIMENSIONLESS TWO BODY MATRIX
*       ELEMENTS USING TALMANS FORMULEA FOR CENTRAL FORCES.
*       THE TALMI INTEGRALS ARE NORMALIZED BY THE INTEGRAL
*       (0.INF.)R**(2M+2)*EXP(-R**2).
*       THUS RTAL(S1+1,S2+1,L,N+1) IS EQUAL TO TALMANS R-COEFFICIENT
*       R(S1,S2,L,N) TIMES THE NORMALIZATION FACTOR.
*       THE EXTERNALS TALMAN AND RACAH ARE NEEDED.
*       C(N+1,L+1) ARE THE NORMALISATION CONSTANT OF THE OSC. W.F.(N,L)
*
      COMMON/SHELL/NX(30),LX(30),XJX(30)
      DIMENSION FCL(1),TLINT(1),T(5),S(5)
      DIMENSION FAC(30),DFAC(30),C(5,8)
      LOGICAL IPASS
      DATA IPASS,RTPI/.FALSE.,1.77245385/
          RTAL(N1,N2,L,M)=      -YY*FLOAT((-1)**(N1+N2+M)*(L+L+1))
     1     *EXP(DFAC(2*(L+N1+N2-1))+FAC(L+N1+N2-1)-FAC(N1)-FAC(N2)
     2     -DFAC(2*(L+N2))-FAC(M)-FAC(L+N1+N2-M)
     3     -DFAC(2*(L+N1)))/2.**(N1+N2)
          IF(IPASS)       GOTO2$ FAC(1)=FAC(2)=DFAC(1)=DFAC(2)=0
      DO 1 N=3,30$ X1=N-1$ X1=ALOG(X1)$ FAC(N)=X1+FAC(N-1)
     1    DFAC(N)=X1+DFAC(N-2)$ RX=SQRT(2./RTPI**3)$ XL=-3
      DO 7 L=1,8$ XL=XL+2.$ X=L+L-1$ C(1,L)=RY=RX=SQRT(ABS(XL)/2.)*RX
      DO 7 N=2,5$ XN=N-1$ X=X+2.
     7    C(N,L)=RY=-SQRT(XN*X/2.)*RY$ YY=4.*RTPI**6$ IPASS=.TRUE.
      N= NTYP
     2  DO 14 I=1,N
    14    FCL(I)=0$ IA=I1$ IB=I2$ IC=I3$ ID=I4
          LA=LX(IA)$LB=LX(IB)$LC=LX(IC)$LD=LX(ID)
           IF(MOD(LA+LB+J,2).NE.0.OR.MOD(LC+LD+J,2).NE.0) RETURN
          IF((LA+LB-J)*(IABS(LA-LB)-J).GT.0) RETURN
          IF((LC+LD-J)*(IABS(LC-LD)-J).GT.0) RETURN
          NA=NX(IA)$NB=NX(IB)$NC=NX(IC)$ND=NX(ID)
          I=XJX(I1)-XJX(I3)$ PH=(-1)**(I+(LC+LB-LA-LD)/2)
          N1MAX=(2*(NA+NB)+LA+LB-J)/2+1$N2MAX=(2*(NC+ND)+LC+LD-J)/2+1
          IF(N2MAX-N1MAX)5,5,6
     6    K=N1MAX$N1MAX=N2MAX$N2MAX=K$K=LA$LA=LC$LC=K$K=LB$LB=LD$LD=K
      IA=I3$IB=I4$IC=I1$ID=I2$ NA=NX(IA)$ NB=NX(IB)$ NC=NX(IC)$ND=NX(ID)
     5    VLA=LA$ VLB=LB$ VLC=LC$ VLD=LD
          VA=XJX(IA)$VB=XJX(IB)$VC=XJX(IC)$VD=XJX(ID)$VJ=J$ DO 9 I=1,N
     9    S(I)=0
      DO 11 N1= 1,N1MAX$ C1=TALMAN(NA,LA,NB,LB,N1-1,J)
      DO 11 N2=1,N2MAX$ C2=TALMAN(NC,LC,ND,LD,N2-1,J)*C1$MAX=N1+N2+J-1
      DO 8 I=1,N
     8    T(I)=0$ DO 12 M=1,MAX$ MI=-30$ DO 12 I=1,N$ MI=MI+30
    12    T(I)=T(I)+RTAL(N1,N2,J,M)*TLINT(MI+M)$   DO 11 I=1,N
    11    S(I)=S(I)+C2*T(I)
          X=C(NA+1,LA+1)*C(NB+1,LB+1)*C(NC+1,LC+1)*C(ND+1,LD+1)
     1     *SQRT((VA+VA+1.)*(VB+VB+1.)*(VC+VC+1.)*(VD+VD+1.))/(VJ+VJ+1.)
     2     *RACAH(VLA,VLB,VA,VB,VJ,.5)*RACAH(VLC,VLD,VC,VD,VJ,.5)*PH
      DO 10 I=1,N
    10    FCL(I)=X*S(I)$  RETURN
      END
      FUNCTION TALMAN(N1,L1,N2,L2,NC,LC)
      COMMON/MOSHTN/NDUM,TAB(1000),NAB(1000)
      DATA NMAX/0/,NOEN/0/
          TALMAN=0$ IF(MOD(L1+L2+LC,2).NE.0) RETURN
          IF((2*NC+LC).GT.(2*(N1+N2)+L1+L2)) RETURN
          NA=N1$ NB=N2$ LA=L1$ LB=L2$ IF(NA-NB)4,5,5
     4    NB=N1$ NA=N2$ LB=L1$ LA=L2
     5    IQ=SHIFT(NA,30)+SHIFT(LA,24)+SHIFT(NB,18)+SHIFT(LB,12)
```

```
     1      *SHIFT(NC,6)+LC$ IR=INBINS(NAB,1,NOEN,IQ)$ IF(IR)2,2,7
     2      Z=TALMAN1(NA,LA,NB,LB,NC,LC)$ TALMAN=Z$ IF(NMAX-1000)1,9,9
     1      NMAX=NMAX+1$ NOEN=MAX0(NMAX,NOEN)$ NDUM=NOEN
     3      TAB(NMAX)=Z$ NAB(NMAX)=IQ$ CALL SORTAG1(NAB,1,NOEN,TAB,NMAX)
            RETURN
     7      TALMAN=TAB(IR)$ RETURN
     9      PRINT1000$ NMAX=500$ GOTO3
  1000 FORMAT(*      TALMAN TABLE IS FULL... LENGTHIS 1000  *)
       END
       FUNCTION TALMAN1(NA,LA,NB,LB,NC,LC)
*
*           THIS IS THE C-COEFFICIENT OF TALMAN(SEE NUCL.PHYS. A141
*           (1970)273, EQ(15)). THE FUNCTION SUBROUTINE CG000(L1,L2,LC)
*           FOR THE 3-J SYMBOLS IS THE ONLY EXTERNAL NEEDED.
*           FAC(N+1) IS THE LOG OF FACTORIAL N.
*           NA IS ASSUMED TO BE GT NB. IF OTHERWISE CALCULATION
*           WILL BE SLOWER BUT RESULT WILL BE CORRECT.
*
            DIMENSION FAC(30),DFAC(30)
       LOGICAL IPASS
       DATA IPASS/.FALSE./
            IF(IPASS)       GOTO2$ FAC(1)=FAC(2)=DFAC(1)=DFAC(2)=0
       DO 1 N=3,30$ X1=N-1$ X1=ALOG(X1)$ FAC(N)=X1+FAC(N-1)
     1      DFAC(N)=X1+DFAC(N-2)$ IPASS=.TRUE.
     2      NX=LA+LB+LC-2$ NY=(LA+LB-LC)/2-1$ NZ=(LA+LB-LC)/2-NC-1
            S=0$ LAM=LA+LA$ NAA=NA+1$ NBB=NB+1
       DO 6 M=1,NAA$NMIN=MAX0(1,NC-M+1-NY)$LAM=LAM+2$IF(NMIN.GT.NBB)GOTO6
            NS=NY+M$ NT=NZ+M$           NR=NX+M+M+NMIN+NMIN-2
            LBN=LB+LB+NMIN+NMIN-2$ S1=FAC(M)+FAC(NAA-M+1)+DFAC(LAM)
       DO 7 N=NMIN,NBB$LBN=LBN+2$ NR=NR+2
     7      S=S+(1-2*IABS(MOD(M+N,2)))*EXP(DFAC(NR)+FAC(NS+N)
     1       - FAC(N)-FAC(NBB-N+1)-DFAC(LBN)-FAC(NT+N)-S1)
     6 CONTINUE$  TALMAN1=(-1)**(NA+NB)*(2*LA+1)*(2*LB+1)*2**(NY+1)
     1      *CG000(LA,LB,LC)*S$ RETURN
*    1      *CG000(LA,LB,LC)*S
            PRINT8, NA,LA,NB,LB,NC,LC,TALMAN1$ RETURN
     8      FORMAT(10X,6I3,E20.10)
       END
       SUBROUTINE SORTAG1(A,II,JJ,TAG,IOUT)
*      THE ELEMENTS A(II) TO A(JJ) ARE ASSUMED IN INCREASING
*      ORDER EXCEPT FOR A(IOUT), WHICH IS TO BE POSITIONED .
*      TAG IS PERMUTED ALONG WITH A .
       DIMENSION A(1), TAG(1)
       INTEGER A
       IO= IOUT$ I=II$ J= JJ
       IF(J-I-1) 99,1,1
  1    IF(IO-I) 99,13,5
  5    IF(J-IO) 99,11,9
  9    IF(A(IO)-A(IO+1)) 11,99,3
  11   IF(A(IO-1)-A(IO)) 99,99,7
  7    T= TAG(IO-1)$ TAG(IO-1)= TAG(IO)$ TAG(IO)= T
       IT= A(IO-1)$ A(IO-1)= A(IO)$ A(IO)= IT$ IO= IO-1
       IF(IO-I) 99,99,11
  3    T= TAG(IO+1)$ TAG(IO+1)= TAG(IO)$ TAG(IO)= T
       IT= A(IO+1)$ A(IO+1)= A(IO)$ A(IO)= IT$ IO= IO+1
       IF(J-IO) 99,99,13
  13   IF(A(IO)-A(IO+1)) 99,99,3
  99   RETURN
       END
       FUNCTION CG000(IA,IB,IC)
C      SUBROUTINE TO CALCULATE THE 3-J SYMBOL (A,B,C/0,0,0)
C      THE TRIANGULAR INEQUALITIES ARE ALREADY SATISFIED.
```

```
C      ALSO (-1)**(A+B+C)=+1 HAS BEEN SATISFIED.
       DIMENSION FL(130)
       DATA IPASS/0/
           IF(IPASS.EQ.1) GOTO2$ FL(1)=0$DO1 K=2,130$ A=K-1
     1 FL(K)=FL(K-1)+ALOG(A)$ IPASS=1
     2 CG000=.0
       IF(IC.GT.(IA+IB).OR.IC.LT.IABS(IA-IB)) RETURN
       IF((-1)**(IA+IB+IC).NE.1) RETURN
       IAB=IA+IB
    40 IG2=IAB+IC
       IG2P=IG2+2
       IG=IG2/2
       ICP=IC+1
       IACMBP=IA-IB+ICP
       IBCMAP=IB-IA+ICP
       IABMCP=IAB-IC+1
    70 IGP 1=IG+1
       IGMAP1=IGP1-IA
       IGMBP1=IGP1-IB
       IGMCP1=IGP1-IC
       IGD2=IG/2
       NSIG=IG-IGD2-IGD2
       SIG=1-NSIG-NSIG
       X=FL(IGP1)-FL(IGMAP1)-FL(IGMBP1)-FL(IGMCP1)
       DLG=.5*(FL(IACMBP)+FL(IBCMAP)+FL(IABMCP)-FL(IG2P))
       YLG=X+DLG
       Y=EXP(YLG)
       TJ=SIG*Y
       CG000=TJ
       RETURN
       END
       SUBROUTINE PREPSA(IBLK,IREW,KU)
*
*      TO PREPARE THE SAUNIER TAPE FOR USE BY PPMEX .
*      IF(IREW.EQ.0) DO NOT REWIND 5 ....
       COMMON/TEMP/ IQNA(250),ZA(250),IDIM,IDIM2
       COMMON /INDN/ INDEX(71),IFIRS(71),IMS
       DIMENSION Z(5),IRCT(8)
       LOGICAL INITIAL
       DATA IRC,IRCT/0,17661,17662,17663,17744,17745,17750,17761,17762/
       DATA INITIAL /.T./
       DATA IDIM,IDIM2/250,500/
       IF(IREW.NE.0) REWIND KU
       IMS= ILA= ILB= 1 $ IB= IBLK-1$ IF(.NOT.INITIAL) GO TO 11
       INITIAL= .F.$ CALL OPENMS(9,INDEX,71,0)
  11   IF(IB.LT.1) GO TO 5$ DO 1 K = 1 , IB$ DO 1 L = 1 , 8364
   1   READ(KU,102) IDUMP
 102   FORMAT(I6)
*
*      TAPE IS NOW POSITIONED AT CORREXT BLOCK .
   5   DO 7 L = 1 , IDIM2
   7   IQNA(L)= 0
       DO 3 L = 1 , 8364$ IF(KU.EQ.6) IRC= IRC+1
  30   READ(KU,1013) I1,I2,I3,I4,J,(Z(K),K=1,5)
       DO 201 JC= 1 , 8$ IF(IRC.NE.IRCT(JC)) GO TO 201
       PRINT 1015,I1,I2,I3,I4,J,(Z(K),K=1,5),IRC
  201   CONTINUE
 1015 FORMAT(1X5I2,5E14.6,I5/)
 1013 FORMAT(5I2,5E14.6)
       I= I4+ (I3*(I3-1))/2+ ((I2-1)*((I1-1)*I1+ I2))/2
      1 + ((I1-1)*I1*(2-I1+ I1**2))/8
       IQN= SHIFT(I,5)+ SHIFT(J,1)
```

```
      DO 3 IT= 1 , 2$ IF(Z(IT).EQ.0.) GO TO 3
      IQNA(ILB)= IQN+ IT-1$ ZA(ILB)= Z(IT)$ ILB= ILB+1
      IF(ILB.GT.IDIM.OR.( L .GE.8364.AND.IT.EQ.2)) GO TO 150
 3    CONTINUE$ IMS= IMS-1$ RETURN
 150  IFIRS(IMS)= IQNA(1)$ CALL WRITMS(9,IQNA,IDIM2,IMS)
      IMS= IMS+1$ ILB=1$ DO 9 LT= 1 , IDIM2
 9    IQNA(LT)=0$ GO TO 3
      END
      SUBROUTINE LABELF(M,GA)
C
C          ROUTINE PUTS TWO PARTICLE M.E.S (IA,IB,2J,2TIGIIC,ID,2J,2T)
C          INTO A ONE-DIMENSIONAL ARRAY.
C          IF M=0,      M.E.S ARE READ IN ON DATA CARDS,
C          IF M=2,      M.E.S ARE CALCULATED BY CALLING PPME.
C          THE GENERAL RULE OF THE ORDER OF THE INPUT M.E.S IS
C          IB.GE.IA.AND.ID.GE.IC.  IN PARTICULAR IF THE INPUT ARE
C          PPME TO BE CONVERTED TO PHME. THEN IN ADDITION WE HAVE
C          IC.GE.IA, AND ID.GE.IB IFF IC.EQ.IA.
C
      COMMON/SHELL/NX(30),LX(30),XJX(30)
          LOGICAL IABCD,IPAULI,I12,I34
      COMMON/LBL5/IA,IB,IC,ID,JJ2,JT2,N,IP2
      COMMON/UPLOWA/IMIN,IMAX,I1MIN,I1MAX,I2MIN,I2MAX,I3MIN,I3MAX,
     1I4MIN,I4MAX,JMIN,JMAX,ICODE
      COMMON/NUMB/NUM,NUMIN,IPARITY
      DIMENSION II12(200),GA(1)
      COMMON/GMTRXB/GB(32)
      LOGICAL IPRINT,IWRITE
      PRINT 105,M$ ICNT= 0
      ENTRY LABELC
      ENTRY LABELD
      ENTRY LABELE
 105  FORMAT(//* THE FIRST 200 G MAT ELEM OBTAINED BY LABELING ROUTINE,M
     1=*I3//)
      IPRINT=ICODE.EQ.2.OR.ICODE.GE.4$ IWRITE=ICODE.EQ.3.OR.ICODE.GE.4
      IQAB=SHIFT(IA,6)+IB$ IQCD=SHIFT(IC,6)+ID
      IQABCD=SHIFT(IQAB,12)+IQCD$ DO 25 K=1,32
  25      GB(K)=0$ N=NUMIN$ K2=I2MAX-I2MIN+1$ K0=-K2
      DO 24 I1=I1MIN,I1MAX$ K0=K0+K2$ II1=I1-I2MIN-1
      IF(II1.LE.0) II1=0$ K1=(II1*(II1+1))/2$ I2MN=MAX0(I1,I2MIN)
      DO 24 I2=I2MN,I2MAX$ IQ12=SHIFT(I1,6)+I2$ L12=LX(I1)+LX(I2)
      IMN=K0-K1+I2-I2MN+1
      V1=XJX(I1)$ V2=XJX(I2)$ J12=V1+V2$ J21=ABS(V1-V2)$ I12=I1.EQ.I2
      IF(M.NE.1) GOTO103$ IF(IQAB-IQ12)24,50,24
  50  N=II12(IMN)$ GOTO104
 103      II12(IMN)=N
 104  DO 22 I3=I1,I3MAX$ IF(I3-I1) 34,12,11
  12      I4MN=MAX0(I2,I4MIN)$ GOTO10
  11      I4MN=MAX0(I3,I4MIN)
  10  DO 22 I4=I4MN, I4MAX$ IQ34=SHIFT(I3,6)+I4$ L34=LX(I3)+LX(I4)
          IF(MOD(L12-L34,2)) 22,26,22
  26  V3=XJX(I3)$ V4=XJX(I4)$ J34=V3+V4$ J43=ABS(V3-V4)$ I34=I3.EQ.I4
      J2MAX=MIN0(J12,J34)+1$ J2MIN=MAX0(J21,J43)+1
          IF(J2MAX-J2MIN)22,13,13
  13  IQ1234=SHIFT(IQ12,12)+IQ34$ IABCD=IQ1234.EQ.IQABCD
      DO 23 IT1=1,2$ KT=IT1-1$ KJT=KT*16$ IT2=2*KT
      DO 23 JJ1=J2MIN,J2MAX$ KJT=KJT+1$ J=JJ1-1$ IJ2=2*J
          IPAULI=MOD(J+KT,2).EQ.0
      IF(I12.AND.IPAULI)GOTO23$  IF(I34.AND.IPAULI)GOTO23
      N=N+1$ IF(M.EQ.2)GOTO4$ IF(M.EQ.0)GOTO2$ IF(M.EQ.3) GOTO23
      IF(M.EQ.5)  GO TO 501
          IF(IABCD.AND.M.EQ.1) 102,23
```

```
    4 CALL PPME(I1,I2,I3,I4,J,KT,GGG,0)
*           THE TIME-REVERSAL PHASE IS IN PPME.
 504  ICNT= ICNT+1$ IF(ICNT.LT.201.A.IPRINT)
     1 PRINT5000,N,I1,I2,I3,I4,J,KT,GGG
         IF(IWRITE) WRITE(8,5000)N,I1,I2,I3,I4,J,KT,GGG
         GOTO1
    2 READ(8,5000)NN,IA,IB,IC,ID,J,KT,GGG
      JJ2=J*2$ JT2=KT*2
      IQABCD=ID+SHIFT(IC,6)+SHIFT(IB,12)+SHIFT(IA,18)
      JNJT=IQABCD+SHIFT(JJ2,24)+SHIFT(JT2,30)
      INJT=IQ1234+SHIFT(IJ2,24)+SHIFT(IT2,30)
C   2 READ 5000,  NN,IA,IB,IC,ID,JJ2,JT2,GGG,JP1
 5000 FORMAT(I5,5X,6I5,5X,E20.8,I5)
      IF(NN.NE.N.OR.INJT.NE.JNJT) GOTO34
    1 GA(N-NUMIN)=GGG
         GOTO23
  102    IF(JJ2-IJ2) 41,42,41
   42    IF(JT2-IT2) 41,33,41
   41    GB(KJT)=GA(N-NUMIN)
   23    CONTINUE
         IF(IABCD.AND.M.EQ.1) RETURN
   22 CONTINUE
   24    CONTINUE$ IF(M-1)27,28,27
   28    N=0$   RETURN
   27    NUM=N$ IF(IWRITE) ENDFILE8$   RETURN
   34 PRINT2000,N,I1,I2,I3,I4,IJ2,IT2,NN,IA,IB,IC,ID,JJ2,JT2
      RETURN
 2000 FORMAT(2X,49H**********      ERROR IN CARD INPUT      **********/
     15X,14I5)
   33    GB(KJT)=GA(N-NUMIN)$ RETURN
 501  CALL PPMEXT(I1,I2,I3,I4,J,KT,GGG,0)$ GO TO 504
      END
      SUBROUTINE REDFA12(AL1,NOLV,CODA,FLNAME,IDF,IDM,IDG,HBOM,IBL)
*
*     READ FROM SCRATCH TAPE1, INTO TAPE2 THE 7 ARRAYS .
*     LOOK FOR FLNAME, THEN READ TO ENDFILE MARK.
*     NOLV= F(1)= NUMBER OF LEVELS .
*     CODA(NOLV)= CODENUMBER ARRAY= F(2,...,NOLV+2) .
*     IDF= NUMBER OF F ELEMENTS READ .
*     IDM= DIM OF F ARRAY
*     WRITE ON 2 OUT OF TA FLNAME,NOLV,CODA(16),IDF,IDG,HBOM.
*
      DIMENSION AL1(1),CODA(1),NAA(7),FT(5)
      DIMENSION TA(25)
      INTEGER CODA
      DATA NAA/3LFPR,3LFNE,3LFDN,3LRTS,3LRTD,3LPTS,3LPTD/
      REWIND 1
      TA(1)= FLNAME$ TA(2)= NOLV$ DO 7 K = 1 , NOLV
 7    TA(K+2)= CODA(K)
      TA(21)= IDF$ TA(22)= IDG$ TA(23)= HBOM
      WRITE(2) TA $ PRINT 107, TA
 107  FORMAT(//* TA CONTROL BLOCK*//1P5(5E20.6//)/)
      IF(IBL.GT.1) WRITE(2)(AL1(I),I=1,IDG)
      DO 1 L = 1 ,7$IF((IBL.EQ.1.A.L.LT.3).OR.(IBL.GT.1.A.L.GT.2))GOTO1
      READ(1,1010) TITLE$ IDF=0
 1010 FORMAT (A10)
      IF(TITLE.NE.FLNAME) GO TO 101
      READ(1,1003) JF,FT,IPH$ IF(FT(1).LT.1) GO TO 101
 1003 FORMAT(8X,O12,5E20.10,I2)
      NOLV= FT(1)$ IFL=2$ AL1(1)= FT(L)$ IF(L.GT.5) AL1(1)=FT(L-2)
      DO 31 K = 1 , NOLV$ READ(1,1003) IFD,FT,IPH
      CODA(K)= FT(1)$ AL1(IFL)= FT(L) $ IF(L.GT.5)AL1(IFL)=FT(L-2)
```

```
   31 IFL= IFL+1
 33   IF(IFL.GT.IDM) GO TO 101$ READ(1,1003) IFD,FT,IPH
      AL1(IFL)= FT(L)$IF(L.GT.5)  AL1(IFL)= IPH*FT(L-2)
 5    IFL= IFL+1$ IF(EOF(1)) 35,33
   35 IDF= IFL-2
      REWIND 1
      WRITE(2)(AL1(I),I=1,IDF)
 1     CONTINUE
  105 RETURN
  101 PRINT 103
  103 FORMAT (/* F ARRAY NOT ON TAPE1 */)
      GO TO 105
      END
      SUBROUTINE PHGPHD(IU,IV,IW,IX,J,GPH,GA)
*
*          CALCULATES THE F-MATRIX ELEMENT F(A,B,C,D,J) FROM THE FORMULA
*          F(A,B,C,D,J,T)=-SUM(JP)(.5*(2*JP+1)*RACAH(A,D,B,C,JP,J)*
*                          (-G(0)+3*G(1))*D(T,0) + (G(0)+G(1)))
*          WHERE    G(T) IS THE G-MATRIX G(D,A,B,C,JP,T),
*          AND      D(T,TP) IS THE KRONEKER DELTA-FUNCTION.
*          GPH(T+1)=F(A,B,C,D,J,T)
*
      COMMON/SHELL/NX(30),LX(30),XJX(30)
      COMMON/PHGCON/NMAX ,NOEN ,IPASS
      DIMENSION GA(1),GPH(1),GX(2),TAB(200,2),NAB(200)
      COMMON/GMTRXB/GB(32)$ INTEGER TAG(200)$ LOGICAL IPASS
      COMMON/LBL5/IP,IQ,IR,IS,JJ2,JDUM,N,IDUM
      DATA NMAX,NOEN/0,0/,IPASS/.F./
      ENTRY PHGPHC
      IF(IPASS)GOTO3$ DO 4 I=1,200
    4     TAG(I)=I $  IPASS=.T.
    3 IF(IU-IV)30,31,31
   30 IA=IV$ IB=IU$ IC=IX$ ID=IW$  FAT=1$ IF(IA-IC)33,32,32
   31 IA=IU$ IB=IV$ IC=IW$ ID=IX$  FAT=0$ IF(IA-IC)33,32,32
   33 II=IA$ IA=IC$ IC=II$ II=IB$ IB=ID$ ID=II
   32 II=SHIFT(J,24)+SHIFT(IA,18)+SHIFT(IB,12)+SHIFT(IC,6)+ID
         VC=XJX(IC)$ VD=XJX(ID)$ VJ=J$  VA=XJX(IA)$ VB=XJX(IB)
      IFAT=FAT*(VA+VB+VC+VD)$FAT=1-2*MOD(IFAT,2)$ JJ2=-1$GPH(1)=GPH(2)=0
      IF(NOEN.LT.1)GOTO8$ JJ=INBINS(NAB,1,NOEN,II)$ IF(JJ)8,8,17
   17 GPH(1)=TAB(TAG(JJ),1)*FAT$ GPH(2)=TAB(TAG(JJ),2)*FAT$ RETURN
    8 JAD=VA+VD$ JBC=VB+VC$ JDA=ABS(VA-VD)$ JCB=ABS(VB-VC)$ J2MAX=MIN0(
     1JAD,JBC)+1$ J2MIN=MAX0(JDA,JCB)+1$ IF(J2MAX-J2MIN)11,10,10
   10     IP=MIN0(IA,ID)$ IQ=MAX0(IA,ID)$ IR=MIN0(IB,IC)$ IS=MAX0(IB,IC)
          IF(IP.GT.IR.OR.(IP.EQ.IR.AND.IQ.GT.IS))6,7
    6     JJ=IP$ IP=IR$ IR=JJ$ JJ=IQ$ IQ=IS$ IS=JJ
    7 CALL LABELC(1,GA)$ IF(N) 1,1,12
   12 DO 2 J2=J2MIN,J2MAX$ JP=J2-1$ VJP=JP$ DO 5 JT1=1,2$ JT=JT1-1
          IFAS=0$ IF(ID.GT.IA) IFAS=IFAS+JP+JT+JAD
          IF(IB.GT.IC) IFAS=IFAS+JP+JT+JBC$ PH=1-2*MOD(IFAS,2)
    5     GX(JT1)=GB(JT*16+J2-J2MIN+1)*PH
          S0=(-GX(1)+3.*GX(2))/2.$ S1=(GX(1)+GX(2))/2.
          XX=RACAH(VA,VD,VB,VC,VJP,VJ)*(VJP+VJP+1.)
      GPH(1)=GPH(1)-XX*S0*FAT
    2 GPH(2)=GPH(2)-XX*S1*FAT$ IF(NMAX-200)20,21,21
   20     NMAX=NMAX+1$ NOEN=MAX0(NMAX,NOEN)
   22     NAB(NMAX)=II$ DO 19 I=1,2
   19     TAB(TAG(NMAX),I)=GPH(I)*FAT$ CALL SORTAG1(NAB,1,NOEN,TAG,NMAX)
          RETURN
   21     PRINT2002$ NMAX=1$ GOTO22
 2002     FORMAT(*    MESSAGE FROM PHGPHD. TABLE TAB(200) IS FULL.*)
    1     PRINT2001,IA,IB,IC,ID,J
 2001     FORMAT(*   MESSAGE FROM PHGPHD. G-MATRIX ELEMENTS NEEDED FOR
```

```
     1THE CONSTRUCTION OF THE F-MATRIX ELEMENT*/*    F(*513*) DO NOT
     2EXIST IN THE ARRAY GA OF SUBROUTINE LABELD*)
  11      RETURN
      END
```

APPENDIX II

SAMPLE RESULTS FOR ^{4}He

This case was run with the S.P. No. 2 potential, N = 0-2, $\hbar\omega$ = 13.5 MeV. Only 2 iterations were performed to illustrate the output format.

TRIAL WAVE FUNCTION COMPUTATION FOR HE 4

ELLIPSOIDAL MODIFIED HARMONIC OSCILLATOR COMPUTATION

DEF. PAR. OF POTENTIAL ARE, BETA= .001000, GAMMA= 0.000

MAJOR SHELL N	SPIN-ORBIT STRENGTH CHI	LSQR STRENGT D
0	.0700	0.0000
1	.0700	0.0000
2	.0700	0.0000
3	.0700	.3500
4	.0500	.4500
5	.0500	.4500
6	.0500	.4500

STATE ORDERING ACCORDING TO INCREASING ENERGY EIG. VAL., FOR SUMMING PURPOSES

INCR. ORD.NO. J	1	2	3	4	5	6	7	8	9	10	11	12
NATU. ORD.NO. I	1	2	4	3	5	6	8	10	7	9	13	12
EAV. OF JTH ST.	1.125000	1.839774	1.840226	1.945001	2.554636	2.554910	2.555451	2.625002	2.729664	2.730338	3.122557	3.122734
INCR. ORD.NO. J	13	14	15	16	17	18	19	20	21	22	23	24
NATU. ORD.NO. I	15	19	11	20	16	14	18	17	22	23	25	28
EAV. OF JTH ST.	3.123082	3.123619	3.315088	3.315911	3.367535	3.367887	3.368575	3.420510	3.799455	3.799591	3.799864	3.800272
INCR. ORD.NO. J	25	26	27	28	29	30	31	32	33	34	35	36
NATU. ORD.NO. I	34	27	30	32	24	26	29	33	35	21	31	42
EAV. OF JTH ST.	3.800816	4.006932	4.007358	4.008205	4.024429	4.024659	4.025117	4.025795	4.124944	4.132057	4.133014	4.411872
INCR. ORD.NO. J	37	38	39	40	41	42	43	44	45	46	47	48
NATU. ORD.NO. I	45	40	44	49	54	37	47	50	52	41	39	43
EAV. OF JTH ST.	4.411980	4.412195	4.412518	4.412947	4.413484	4.664320	4.664592	4.665135	4.665948	4.686735	4.687002	4.687335
INCR. ORD.NO. J	49	50	51	52	53	54	55	56	57	58	59	60
NATU. ORD.NO. I	48	53	60	62	68	70	69	75	82	46	38	51
EAV. OF JTH ST.	4.687833	4.688493	5.001798	5.001886	5.002061	5.002324	5.002675	5.003113	5.003638	4.826891	4.828090	4.839319
INCR. ORD.NO. J	61	62	63	64	65	66	67	68	69	70	71	72
NATU. ORD.NO. I	55	56	36	72	71	61	76	83	63	66	65	64
EAV. OF JTH ST.	4.839853	4.840847	4.902523	5.299224	5.299418	5.299806	5.300387	5.301159	5.326751	5.326879	5.327137	5.327522
INCR. ORD.NO. J	73	74	75	76	77	78	79	80	81	82	83	84
NATU. ORD.NO. I	74	81	58	59	79	77	67	73	80	57	84	78
EAV. OF JTH ST.	5.328034	5.328673	5.506713	5.507303	5.508472	5.524201	5.524526	5.525168	5.526112	5.624889	5.631934	5.633208

ENERGY EIG.VAL., JZ AVER. AND MAJOR SHELL Q.N. IN THE NATURAL ORDER(ORDER OF COMPUTATION)

NATU. ORD.NO. I	1	2	3	4	5	6	7	8	9	10	11	12
ENE. OF ITH ST.	1.500000	2.429706	2.640001	2.430294	3.359528	3.359883	3.709547	3.360587	3.710456	3.500002	4.380461	3.995672
JZA. OF ITH ST.	.50000	.50000	.50000	-1.50000	.50000	-1.50000	.50000	2.50000	-1.50000	.50000	.50000	-1.50000
NMSH OF ITH ST.	0	1	1	1	2	2	2	2	2	2	3	3
NATU. ORD.NO. I	13	14	15	16	17	18	19	20	21	22	23	24
ENE. OF ITH ST.	3.995453	4.485854	3.996109	4.485400	4.591010	4.486744	3.996764	4.381538	5.514394	4.849332	4.849499	5.299274

JZA. OF ITH ST.	.50000	-1.50000	2.50000	.50000	.50000	2.50000	-3.50000	-1.50000	.50000	.49999	-1.9996	.50000
NMSH OF ITH ST.	3	3	3	3	3	3	3	3	4	4	4	4
NATU. ORD.NO. I	25	26	27	28	29	30	31	32	33	34	35	36
ENE. OF ITH ST.	4.849833	5.299567	5.264263	4.850333	5.300148	5.264816	5.515678	5.265915	5.301012	4.851000	5.499944	6.555023
JZA. OF ITH ST.	2.49997	-1.50000	.50000	-3.50000	2.50000	-1.50000	-1.50000	2.50000	-3.50000	4.50000	.50000	.50000
NMSH OF ITH ST.	4	4	4	4	4	4	4	4	4	4	4	5
NATU. ORD.NO. I	37	38	39	40	41	42	43	44	45	46	47	48
ENE. OF ITH ST.	6.079131	6.405783	6.124378	5.574635	6.124170	5.574248	6.124794	5.575021	5.574377	6.404197	6.079479	6.125416
JZA. OF ITH ST.	.50000	-1.50000	-1.50000	2.50000	.50000	.50000	2.50000	-3.50000	-1.50000	.50000	-1.50000	-3.50000
NMSH OF ITH ST.	5	5	5	5	5	5	5	5	5	5	5	5
NATU. ORD.NO. I	49	50	51	52	53	54	55	56	57	58	59	60
ENE. OF ITH ST.	5.575536	6.080173	6.429112	6.081212	6.126241	5.576179	6.429801	6.431105	7.499891	7.263973	7.264743	6.254180
JZA. OF ITH ST.	4.50000	2.50000	.50000	-3.50000	4.50000	-5.50000	-1.50000	2.50000	.50000	.50000	-1.50000	.50000
NMSH OF ITH ST.	5	5	5	5	5	5	5	5	6	6	6	6
NATU. ORD.NO. I	61	62	63	64	65	66	67	68	69	70	71	72
ENE. OF ITH ST.	6.849756	6.254283	6.904084	6.905027	6.904556	6.904241	7.299389	6.254488	6.255205	6.254795	6.849269	6.849025
JZA. OF ITH ST.	2.50000	-1.50000	.49999	-3.49999	2.49997	-1.49997	-1.50000	2.50000	4.50000	-3.50000	-1.50000	.50000
NMSH OF ITH ST.	6	6	6	6	6	6	6	6	6	6	6	6
NATU. ORD.NO. I	73	74	75	76	77	78	79	80	81	82	83	84
ENE. OF ITH ST.	7.300215	6.905654	6.255716	6.850487	7.298972	7.515935	7.266273	7.301433	6.906436	6.256330	6.851459	7.514207
JZA. OF ITH ST.	2.50000	4.50000	-5.49999	-3.49999	.50000	-1.50000	2.50000	-3.50000	-5.50000	6.50000	4.50000	.50000
NMSH OF ITH ST.	6	6	6	6	6	6	6	6	6	6	6	6

COMP. OF 1TH L. ORB IN SPIN-CART. DEFORM. BASIS.[MU,NX,NY,NZ] .
QN= [0]
CI= 1.0000000
EXPANSION COEF. IN [NT,J+K,J-K] SPHERICAL BASIS
QN= [100] [20201] [20302]
CI= 1.0000000 .0001727 -.0002116
QN= [1] [20102] [20203]
CI= 1.0000000 -.0001727 -.0002116

```
 TIMEPRT CALLED BY NUHART  AT LINE   23, CP TIME=   16.568 SEC, SYSTEM TIME  18.34.56.

HARTREE-FOCK ITERATIONS WITH THE PARAMETER LIST,      RONI      ROPI      FPR       FNE       FDN       SPE
                          MASS STORAGE ARRAY NAMES,   RONI      ROPI      FPF       FNF       FDF

ITERATION NO  1 FOR  2 PROTONS AND  2 NEUTRONS, NUCLEUS= HE  4

                          ********** THE LOWEST  0 NEUTRON(S) STATES, ALL COMP. .GT.  .003 **********

REMAINING S.P. ORBITS AT ENERGIES(MEV)

  -11.30798  -11.30798   15.31936   15.31936   15.32285   15.32285   17.87555   17.87555   24.29343   24.29343
   27.13201   27.13201   27.13283   27.13283   27.13447   27.13447   29.14350   29.14350   29.14501   29.14501

                          ********** THE LOWEST  0 PROTON(S)  STATES, ALL COMP. .GT.  .003 **********

REMAINING S.P. ORBITS AT ENERGIES(MEV)

  -10.62198  -10.62198   16.30268   16.30268   16.30618   16.30618   18.85888   18.85888   25.21901   25.21901
   28.03887   28.03887   28.03968   28.03968   28.04131   28.04131   30.05036   30.05036   30.05185   30.05185

                                        **********

 TOTAL BINDING ENERGY=   -21.17943 MEV, TOTAL PROTON POTENTIAL ENERGY=  -10.26194 MEV, TOTAL NEUTRON POTENTIAL ENERGY=  -10.91749
                    DELTAPN ENERGY=      .65555 MEV, TOTAL ONE-BODY ENERGY=     0.00000 MEV
                                        **********

ITERATION NO  2 FOR  2 PROTONS AND  2 NEUTRONS, NUCLEUS= HE  4

      MULTIPOLE ANALYSIS OF NEUTRON(S) H-F POT. V AND DENS. MATR. D, WITH LISTING OF ALL COMPS WITH EN. CONTR. GT  .0090

GN= [ 1010000] [ 10500001] [ 50100001] [ 50500001]
VR=  -15.98249   -8.92312   -8.92312   34.49191
VI=    0.00000    0.00000    0.00000    0.00000
DR=    1.39870     .14729     .14729     .01551
DI=    0.00000    0.00000    0.00000    0.00000
FOR MULTIPOL. LAM=  0, THE TOTAL CONTRIB. TO THE POT. ENERGY IS    -12.22419 MEV
```

```
FOR MULTIPOL. LAM=  1, THE TOTAL CONTRIB. TO THE POT. ENERGY IS      -.00000 MEV
FOR MULTIPOL. LAM=  2, THE TOTAL CONTRIB. TO THE POT. ENERGY IS      -.00000 MEV
FOR MULTIPOL. LAM=  3, THE TOTAL CONTRIB. TO THE POT. ENERGY IS      -.00000 MEV
FOR MULTIPOL. LAM=  4, THE TOTAL CONTRIB. TO THE POT. ENERGY IS      -.00000 MEV
FOR MULTIPOL. LAM=  5, THE TOTAL CONTRIB. TO THE POT. ENERGY IS      -.00000 MEV

                          ********** THE LOWEST 14 NEUTRON(S) STATES, ALL COMP. .GT.  .003 **********

STATE NO.  1, WITH S.P. EN.= -12.383931 MEV., PAR.=  1, OCCUPIED BY A NEUTRON(S)
QN= [0, 1, -1] [2, 1, -1]
RP=   .9855973   .1691094
IP=  0.0000000  0.0000000

STATE NO.  2, WITH S.P. EN.= -12.383931 MEV., PAR.=  1, OCCUPIED BY A NEUTRON(S)
QN= [0, 1,  1] [2, 1,  1]
RP=   .9855973   .1691094
IP=  0.0000000  0.0000000

STATE NO.  3, WITH S.P. EN.=  17.094385 MEV., PAR.= -1, OCCUPIED BY A
QN= [1, 3, -1]
RP=   .9999999
IP=  0.0000000

STATE NO.  4, WITH S.P. EN.=  17.094085 MEV., PAR.= -1, OCCUPIED BY A
QN= [1, 3,  1]
RP=   .9999999
IP=  0.0000000

STATE NO.  5, WITH S.P. EN.=  17.095684 MEV., PAR.= -1, OCCUPIED BY A
QN= [1, 3,  3]
RP=  1.0000000
IP=  0.0000000

STATE NO.  6, WITH S.P. EN.=  17.095684 MEV., PAR.= -1, OCCUPIED BY A
QN= [1, 3, -3]
RP=  1.0000000
IP=  0.0000000

STATE NO.  7, WITH S.P. EN.=  20.115797 MEV., PAR.= -1, OCCUPIED BY A
QN= [1, 1,  1]
RP=   .9999999
IP=  0.0000000

STATE NO.  8, WITH S.P. EN.=  20.115797 MEV., PAR.= -1, OCCUPIED BY A
QN= [1, 1, -1]
RP=   .9999999
IP=  0.0000000

STATE NO.  9, WITH S.P. EN.=  25.472066 MEV., PAR.=  1, OCCUPIED BY A
QN= [0, 1,  1] [2, 1,  1]
RP=  -.1691094   .9855973
IP=  0.0000000  0.0000000
```

```
STATE NO. 10, WITH S.P. EN.=   25.472066 MEV., PAR.=  1, OCCUPIED BY A
QN= [0, 1, -1] [2, 1, -1]
RP=  -.1691094    .9855973
IP=  0.0000000  0.0000000

STATE NO. 11, WITH S.P. EN.=   29.200555 MEV., PAR.=  1, OCCUPIED BY A
QN= [2, 5,  1]
RP=  1.0000000
IP=  0.0000000

STATE NO. 12, WITH S.P. EN.=   29.200555 MEV., PAR.=  1, OCCUPIED BY A
QN= [2, 5, -1]
RP=  1.0000000
IP=  0.0000000

STATE NO. 13, WITH S.P. EN.=   29.200912 MEV., PAR.=  1, OCCUPIED BY A
QN= [2, 5,  3]
RP=  1.0000000
IP=  0.0000000

STATE NO. 14, WITH S.P. EN.=   29.200912 MEV., PAR.=  1, OCCUPIED BY A
QN= [2, 5, -3]
RP=  1.0000000
IP=  0.0000000

REMAINING S.P. ORBITS AT ENERGIES(MEV)

    29.20162    29.20162    31.40131    31.40131    31.40192    31.40192
WAVEFUNCTION PUNCHED WITH HEADER= HE  4        2   0  NEUTRON(S)021SP2.6LV    1     13.50 11/09/70  18.35.01.

        MULTIPOLE ANALYSIS OF PROTON(S)  H-F POT., V AND DENS. MATR. D, WITH LISTING OF ALL COMPS WITH EN. CONTR. GT  .0090

QN= [ 1010000] [ 1050000] [ 5010000] [ 5050000]
VR=  -15.03117   -8.76113   -8.76113   35.87213
VI=    0.00000    0.00000    0.00000    0.00000
DR=    1.40001     .14103     .14103     .01421
DI=    0.00000    0.00000    0.00000    0.00000
FOR MULTIPOL. LAM=  0, THE TOTAL CONTRIB. TO THE POT. ENERGY IS    -11.50264 MEV

FOR MULTIPOL. LAM=  1, THE TOTAL CONTRIB. TO THE POT. ENERGY IS      -.00000 MEV

FOR MULTIPOL. LAM=  2, THE TOTAL CONTRIB. TO THE POT. ENERGY IS      -.00000 MEV

FOR MULTIPOL. LAM=  3, THE TOTAL CONTRIB. TO THE POT. ENERGY IS      -.00000 MEV

FOR MULTIPOL. LAM=  4, THE TOTAL CONTRIB. TO THE POT. ENERGY IS      -.00000 MEV

FOR MULTIPOL. LAM=  5, THE TOTAL CONTRIB. TO THE POT. ENERGY IS      -.00000 MEV

                        ********** THE LOWEST 14 PROTON(S)  STATES, ALL COMP. .GT.  .003 **********

STATE NO.  1, WITH S.P. EN.=  -11.665048 MEV., PAR.=  1, OCCUPIED BY A PROTON(S)
QN= [0, 1, -1] [2, 1, -1]
RP=   .9862931    .1650030
IP=  0.0000000  0.0000000
```

```
STATE NO.  2, WITH S.P. EN.=  -11.665048 MEV., PAR.=  1, OCCUPIED BY A PROTON(S)
QN= [0, 1,  1] [2, 1,  1]
RP=   .9862931    .1650030
IP=  0.0000000  0.0000000

STATE NO.  3, WITH S.P. EN.=   18.098119 MEV., PAR.= -1, OCCUPIED BY A
QN= [1, 3, -1]
RP=   .9999999
IP=  0.0000000

STATE NO.  4, WITH S.P. EN.=   18.098119 MEV., PAR.= -1, OCCUPIED BY A
QN= [1, 3,  1]
RP=   .9999999
IP=  0.0000000

STATE NO.  5, WITH S.P. EN.=   18.099721 MEV., PAR.= -1, OCCUPIED BY A
QN= [1, 3, -3]
RP=  1.0000000
IP=  0.0000000

STATE NO.  6, WITH S.P. EN.=   18.099721 MEV., PAR.= -1, OCCUPIED BY A
QN= [1, 3,  3]
RP=  1.0000000
IP=  0.0000000

STATE NO.  7, WITH S.P. EN.=   21.113161 MEV., PAR.= -1, OCCUPIED BY A
QN= [1, 1,  1]
RP=   .9999999
IP=  0.0000000

STATE NO.  8, WITH S.P. EN.=   21.113161 MEV., PAR.= -1, OCCUPIED BY A
QN= [1, 1, -1]
RP=   .9999999
IP=  0.0000000

STATE NO.  9, WITH S.P. EN.=   26.401832 MEV., PAR.=  1, OCCUPIED BY A
QN= [0, 1,  1] [2, 1,  1]
RP=  -.1650030    .9862931
IP=  0.0000000  0.0000000

STATE NO. 10, WITH S.P. EN.=   26.401832 MEV., PAR.=  1, OCCUPIED BY A
QN= [0, 1, -1] [2, 1, -1]
RP=  -.1650030    .9862931
IP=  0.0000000  0.0000000

STATE NO. 11, WITH S.P. EN.=   30.128003 MEV., PAR.=  1, OCCUPIED BY A
QN= [2, 5,  1]
RP=  1.0000000
IP=  0.0000000

STATE NO. 12, WITH S.P. EN.=   30.128003 MEV., PAR.=  1, OCCUPIED BY A
QN= [2, 5, -1]
RP=  1.0000000
IP=  0.0000000

STATE NO. 13, WITH S.P. EN.=   30.128354 MEV., PAR.=  1, OCCUPIED BY A
QN= [2, 5,  3]
RP=  1.0000000
Ip=  0.0000000

STATE NO. 14, WITH S.P. EN.=   30.128354 MEV., PAR.=  1, OCCUPIED BY A
```

```
QN= [2, 5, -3]
RP=  1.0000000
IP=  0.0000000

REMAINING S.P. ORBITS AT ENERGIES(MEV)

     30.12906    30.12906    32.32431    32.32431    32.32491    32.32491
WAVEFUNCTION PUNCHED WITH HEADER= HE  4        2   0  PROTON(S) 0215P2.6LV    1     13.50 11/09/70  18.35.02.

                                        **********

    TOTAL BINDING ENERGY=   -23.72683 MEV, TOTAL PROTON POTENTIAL ENERGY=   -11.50264 MEV, TOTAL NEUTRON POTENTIAL ENERGY=   -12.22419
                          DELTAPN ENERGY=      .72155 MEV, TOTAL ONE-BODY ENERGY=     0.00000 MEV
                                        **********

COMPUTATION OF EXPECTATION VALUE OF THE 21 SYMPLECTIC BILINEAR OPERATORS, IN DIMENSIONLESS UNITS, OSC. PAR. BSQUARE=   3.081 FERMISQ

          NAME          PROTON(S)          NEUTRON(S)          NUCLEON(S)
          NL1            0.00000             0.00000             0.00000
          NL2            0.00000             0.00000             0.00000
          NL3           -.00000             -.00000             -.00000
          NL4            0.00000             0.00000             0.00000
          NL5            0.00000             0.00000             0.00000
          NL6            0.00000             0.00000             0.00000
          NL7            0.00000             0.00000             0.00000
          NL8            0.00000             0.00000             0.00000
          NL9            0.00000             0.00000             0.00000
          AP4             .00000              .00000              .00000
          AP5           -.00010             -.00010             -.00020
          AP6            0.00000             0.00000             0.00000
          AP7            0.00000             0.00000             0.00000
          AP8            0.00000             0.00000             0.00000
          AP9            3.10890             3.11439             6.22330
          AM4            0.00000             0.00000             0.00000
          AM5           -.00036             -.00035             -.00071
          AM6            0.00000             0.00000             0.00000
          AM7            0.00000             0.00000             0.00000
          AM8            0.00000             0.00000             0.00000
          AM9             .79727              .81653             1.61380

R.M.S. S.P. RADIUS FOR   2 PROTON(S) =    1.88723 F.,  BET =  .00009,  GAM =     .00 DEG.

R.M.S. S.P. RADIUS FOR   2 NEUTRON(S)=    1.88160 F.,  BET =  .00009,  GAM =     .00 DEG.

R.M.S. S.P. RADIUS FOR   4 NUCLEON(S)=    1.88442 F.,  BET =  .00009,  GAM =     .00 DEG.

********** TOTAL R.M.S. RADIUS W.R.T. C.M.=   1.63195 F.,   PROTON R.M.S. RADIUS W.R.T. TOTAL C.M.=   1.63358 F.  **********

        COULOMB ENERGY=      .74307 MEV.
```

HARTREE-FOCK ITERATIONS WITH THE PARAMETER LIST, RONI ROPI FPR FNE FDN SPE

MASS STORAGE ARRAY NAMES, RON ROP FPR FNE FDN KEN

ITERATION NO 1 FOR 2 PROTONS AND 2 NEUTRONS, NUCLEUS= HE 4

MULTIPOLE ANALYSIS OF NEUTRON(S) H-F POT, V AND DENS, MATR, D, WITH LISTING OF ALL COMPS WITH EN, CONTR, GT .0090

QN=	(1010000)	(1050000)	(5010000)	(5050000)
VR=	-40.87687	-17.87368	-17.87368	-15.67061
VI=	0.00000	0.00000	0.00000	0.00000
DR=	1.37377	.23571	.23571	.04044
DI=	0.00000	0.00000	0.00000	0.00000

FOR MULTIPOL, LAM= 0, THE TOTAL CONTRIB, TO THE POT, ENERGY IS -32.60764 MEV

FOR MULTIPOL, LAM= 1, THE TOTAL CONTRIB, TO THE POT, ENERGY IS .00000 MEV

FOR MULTIPOL, LAM= 2, THE TOTAL CONTRIB, TO THE POT, ENERGY IS -.00000 MEV

FOR MULTIPOL, LAM= 3, THE TOTAL CONTRIB, TO THE POT, ENERGY IS .00000 MEV

FOR MULTIPOL, LAM= 4, THE TOTAL CONTRIB, TO THE POT, ENERGY IS -.00000 MEV

FOR MULTIPOL, LAM= 5, THE TOTAL CONTRIB, TO THE POT, ENERGY IS .00000 MEV

********** THE LOWEST 14 NEUTRON(S) STATES, ALL COMP, .GT, .003 **********

STATE NO, 1, WITH S.P, EN.= -19.377974 MEV., PAR.= 1, OCCUPIED BY A NEUTRON(S)
QN= [0, 1, -1] [2, 1, -1]
RP= .9907529 .1356784
IP= 0.0000000 0.0000000

STATE NO, 2, WITH S.P, EN.= -19.377974 MEV., PAR.= 1, OCCUPIED BY A NEUTRON(S)
QN= [0, 1, 1] [2, 1, 1]
RP= .9907529 .1356784
IP= 0.0000000 0.0000000

STATE NO, 3, WITH S.P, EN.= 3.057962 MEV., PAR.= -1, OCCUPIED BY A
QN= [1, 3, 1]
RP= 1.0000000
IP= 0.0000000

STATE NO, 4, WITH S.P, EN.= 3.057962 MEV., PAR.= -1, OCCUPIED BY A
QN= [1, 3, -1]
RP= 1.0000000
IP= 0.0000000

STATE NO, 5, WITH S.P, EN.= 3.058400 MEV., PAR.= -1, OCCUPIED BY A
QN= [1, 3, -3]

```
RP=  1.0000000
IP=  0.0000000

STATE NO.  6, WITH S.P. EN.=     3.058400 MEV., PAR.= -1, OCCUPIED BY A
QN= [1, 3,  3]
RP=  1.0000000
IP=  0.0000000

STATE NO.  7, WITH S.P. EN.=     6.351016 MEV., PAR.= -1, OCCUPIED BY A
QN= [1, 1, -1]
RP=  1.0000000
IP=  0.0000000

STATE NO.  8, WITH S.P. EN.=     6.351016 MEV., PAR.= -1, OCCUPIED BY A
QN= [1, 1,  1]
RP=  1.0000000
IP=  0.0000000

STATE NO.  9, WITH S.P. EN.=    13.142871 MEV., PAR.=  1, OCCUPIED BY A
QN= [0, 1,  1] [2, 1,  1]
RP=  -.1356784    .9907529
IP=  0.0000000  0.0000000

STATE NO. 10, WITH S.P. EN.=    13.142871 MEV., PAR.=  1, OCCUPIED BY A
QN= [0, 1, -1] [2, 1, -1]
RP=  -.1356784    .9907529
IP=  0.0000000  0.0000000

STATE NO. 11, WITH S.P. EN.=    17.439012 MEV., PAR.=  1, OCCUPIED BY A
QN= [2, 5,  1]
RP=  1.0000000
IP=  0.0000000

STATE NO. 12, WITH S.P. EN.=    17.439012 MEV., PAR.=  1, OCCUPIED BY A
QN= [2, 5, -1]
RP=  1.0000000
IP=  0.0000000

STATE NO. 13, WITH S.P. EN.=    17.439154 MEV., PAR.=  1, OCCUPIED BY A
QN= [2, 5,  3]
RP=  1.0000000
IP=  0.0000000

STATE NO. 14, WITH S.P. EN.=    17.439154 MEV., PAR.=  1, OCCUPIED BY A
QN= [2, 5, -3]
RP=  1.0000000
IP=  0.0000000

REMAINING S.P. ORBITS AT ENERGIES(MEV)

     17.43944    17.43944    19.73509    19.73509    19.73533    19.73533


        MULTIPOLE ANALYSIS OF PROTON(S)  H-F POT. V AND DENS. MATR. D, WITH LISTING OF ALL COMPS WITH EN. CONTR. GT  .0090


QN= [ 1010000] [ 1050000] [ 5010000] [ 5050000]
VR= -39.90723  -17.78474  -17.78474  -14.32646
VI=   0.00000    0.00000    0.00000    0.00000
DR=   1.37571     .23015     .23015     .03850
DI=   0.00000    0.00000    0.00000    0.00000
```

```
FOR MULTIPOL. LAM=  0, THE TOTAL CONTRIB. TO THE POT. ENERGY IS    -31.81938 MEV

FOR MULTIPOL. LAM=  1, THE TOTAL CONTRIB. TO THE POT. ENERGY IS       .00000 MEV

FOR MULTIPOL. LAM=  2, THE TOTAL CONTRIB. TO THE POT. ENERGY IS      -.00000 MEV

FOR MULTIPOL. LAM=  3, THE TOTAL CONTRIB. TO THE POT. ENERGY IS       .00000 MEV

FOR MULTIPOL. LAM=  4, THE TOTAL CONTRIB. TO THE POT. ENERGY IS      -.00000 MEV

FOR MULTIPOL. LAM=  5, THE TOTAL CONTRIB. TO THE POT. ENERGY IS       .00000 MEV


                    ********** THE LOWEST 14 PROTON(S)  STATES, ALL COMP. .GT.  .003 **********


STATE NO.  1, WITH S.P. EN.=  -18.670839 MEV., PAR.=  1, OCCUPIED BY A PROTON(S)
QN= [0, 1, -1] [2, 1, -1]
RP=   .9911472   .1327677
IP=  0.0000000  0.0000000

STATE NO.  2, WITH S.P. EN.=  -18.670839 MEV., PAR.=  1, OCCUPIED BY A PROTON(S)
QN= [0, 1,  1] [2, 1,  1]
RP=   .9911472   .1327677
IP=  0.0000000  0.0000000

STATE NO.  3, WITH S.P. EN.=    4.091875 MEV., PAR.= -1, OCCUPIED BY A
QN= [1, 3,  1]
RP=  1.0000000
IP=  0.0000000

STATE NO.  4, WITH S.P. EN.=    4.091875 MEV., PAR.= -1, OCCUPIED BY A
QN= [1, 3, -1]
RP=  1.0000000
IP=  0.0000000

STATE NO.  5, WITH S.P. EN.=    4.092309 MEV., PAR.= -1, OCCUPIED BY A
QN= [1, 3, -3]
RP=  1.0000000
IP=  0.0000000

STATE NO.  6, WITH S.P. EN.=    4.092309 MEV., PAR.= -1, OCCUPIED BY A
QN= [1, 3,  3]
RP=  1.0000000
IP=  0.0000000

STATE NO.  7, WITH S.P. EN.=    7.379135 MEV., PAR.= -1, OCCUPIED BY A
QN= [1, 1, -1]
RP=  1.0000000
IP=  0.0000000

STATE NO.  8, WITH S.P. EN.=    7.379135 MEV., PAR.= -1, OCCUPIED BY A
QN= [1, 1,  1]
RP=  1.0000000
IP=  0.0000000

STATE NO.  9, WITH S.P. EN.=   14.071823 MEV., PAR.=  1, OCCUPIED BY A
QN= [0, 1, -1] [2, 1, -1]
RP=  -.1327677   .9911472
IP=  0.0000000  0.0000000
```

STATE NO. 10, WITH S.P. EN.= 14.071823 MEV., PAR.= 1, OCCUPIED BY A
QN= [0, 1, 1] [2, 1, 1]
RP= -.1327677 .9911472
IP= 0.0000000 0.0000000

STATE NO. 11, WITH S.P. EN.= 18.374921 MEV., PAR.= 1, OCCUPIED BY A
QN= [2, 5, 1]
RP= 1.0000000
IP= 0.0000000

STATE NO. 12, WITH S.P. EN.= 18.374921 MEV., PAR.= 1, OCCUPIED BY A
QN= [2, 5, -1]
RP= 1.0000000
IP= 0.0000000

STATE NO. 13, WITH S.P. EN.= 18.375062 MEV., PAR.= 1, OCCUPIED BY A
QN= [2, 5, 3]
RP= 1.0000000
IP= 0.0000000

STATE NO. 14, WITH S.P. EN.= 18.375062 MEV., PAR.= 1, OCCUPIED BY A
QN= [2, 5, -3]
RP= 1.0000000
IP= 0.0000000

REMAINING S.P. ORBITS AT ENERGIES(MEV)

18.37534 18.37534 20.66728 20.66728 20.66751 20.66751

****

TOTAL BINDING ENERGY= -11.52663 MEV, TOTAL PROTON POTENTIAL ENERGY= -31.81938 MEV, TOTAL NEUTRON POTENTIAL ENERGY= -32.60764

DELTAPN ENERGY= .78826 MEV, TOTAL ONE-BODY ENERGY= 52.90038 MEV

****

COMPUTATION OF EXPECTATION VALUE OF THE 21 SYMPLECTIC BILINEAR OPERATORS, IN DIMENSIONLESS UNITS, OSC. PAR. BSQUARE= 3.081 FERMISQ

NAME	PROTON(S)	NEUTRON(S)	NUCLEON(S)
NL1	0.00000	0.00000	0.00000
NL2	0.00000	0.00000	0.00000
NL3	.00000	.00000	.00000
NL4	0.00000	0.00000	0.00000
NL5	0.00000	0.00000	0.00000
NL6	0.00000	0.00000	0.00000
NL7	0.00000	0.00000	0.00000
NL8	0.00000	0.00000	0.00000
NL9	0.00000	0.00000	0.00000
AP4	.00000	.00000	.00000
AP5	-.00002	-.00002	-.00004
AP6	0.00000	0.00000	0.00000
AP7	0.00000	0.00000	0.00000
AP8	0.00000	0.00000	0.00000

AP9	3.07051	3.07363	6.14414
AM4	0.00000	0.00000	0.00000
AM5	-.00008	-.00008	-.00017
AM6	0.00000	0.00000	0.00000
AM7	0.00000	0.00000	0.00000
AM8	0.00000	0.00000	0.00000
AM9	.64467	.65854	1.30321

R.M.S. S.P. RADIUS FOR 2 PROTON(S) = 1.93329 F., BET = .00002, GAM = .00 DEG.

R.M.S. S.P. RADIUS FOR 2 NEUTRON(S)= 1.92900 F., BET = .00002, GAM = .00 DEG.

R.M.S. S.P. RADIUS FOR 4 NUCLEON(S)= 1.93114 F., BET = .00002, GAM = .00 DEG.

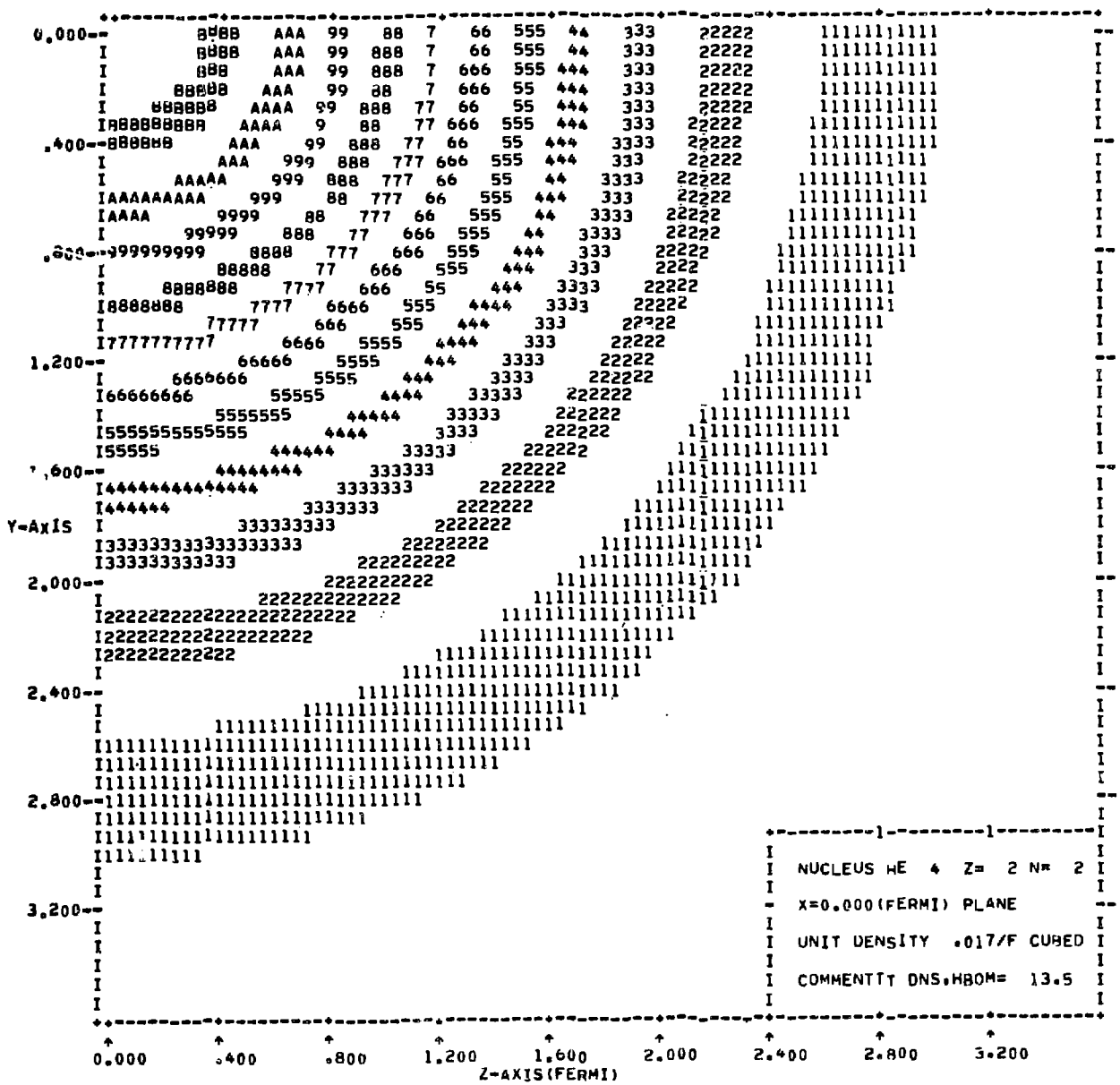
TIMEPRT CALLED BY MULTIP AT LINE 9, CP TIME= 19.055 SEC, SYSTEM TIME 18.35.10.

COMPUTATION OF MULTIPOLE MOMENTS

FOR KPOL,MPOL=	0	0	,MOMENT=	2.00000E+00	,IN UNITS OF BSQR,	MOMENT=	2.00000E+00	,IN UNITS OF F**KPOL
FOR KPOL,MPOL=	2	0	,MOMENT=	1.31379E-04	,IN UNITS OF BSQR,	MOMENT=	4.04842E-04	,IN UNITS OF F**KPOL
FOR KPOL,MPOL=	2	2	,MOMENT=	0.	,IN UNITS OF BSQR,	MOMENT=	0.	,IN UNITS OF F**KPOL
FOR KPOL,MPOL=	4	0	,MOMENT=	2.51408E-08	,IN UNITS OF BSQR,	MOMENT=	2.38725E-07	,IN UNITS OF F**KPOL
FOR KPOL,MPOL=	4	2	,MOMENT=	0.	,IN UNITS OF BSQR,	MOMENT=	0.	,IN UNITS OF F**KPOL
FOR KPOL,MPOL=	4	4	,MOMENT=	0.	,IN UNITS OF BSQR,	MOMENT=	0.	,IN UNITS OF F**KPOL
FOR KPOL,MPOL=	6	0	,MOMENT=	0.	,IN UNITS OF BSQR,	MOMENT=	0.	,IN UNITS OF F**KPOL
FOR KPOL,MPOL=	6	2	,MOMENT=	0.	,IN UNITS OF BSQR,	MOMENT=	0.	,IN UNITS OF F**KPOL
FOR KPOL,MPOL=	6	4	,MOMENT=	0.	,IN UNITS OF BSQR,	MOMENT=	0.	,IN UNITS OF F**KPOL
FOR KPOL,MPOL=	6	6	,MOMENT=	0.	,IN UNITS OF BSQR,	MOMENT=	0.	,IN UNITS OF F**KPOL

TIMEPRT CALLED BY DENSITY AT LINE 21. CP TIME= 19.832 SEC, SYSTEM TIME 18.35.12.

DENSITY PLOT FROM EVALIN (R.Y. CUSSON/ H.C. LEE) C.R.N.L. DATE 11/09/70



DENSITY PRINT. THE INTEGER PRINTED IS= (RHO IN FERM.**(-3))*588.2

111	110	107	104	99	94	88	81	74	66	59	52	45	39	33	28	23	19	15	12	9	7	6	4	3	2	2	0	0	0
111	110	107	104	99	94	87	81	74	66	59	52	45	39	33	28	23	19	15	12	9	7	6	4	3	2	2	0	0	0
110	109	107	103	99	93	87	80	73	66	59	52	45	39	33	27	23	19	15	12	9	7	6	4	3	2	2	0	0	0
109	108	106	102	98	92	86	79	72	65	58	51	45	38	32	27	23	18	15	12	9	7	6	4	3	2	2	0	0	0
108	107	104	101	96	91	85	78	72	65	57	51	44	38	32	27	22	18	15	12	9	7	5	4	3	2	2	0	0	0
106	105	103	99	95	90	84	77	70	63	57	50	43	37	32	26	22	18	14	11	9	7	5	4	3	2	2	0	0	0
104	103	101	97	93	88	82	76	69	62	55	49	42	36	31	26	21	18	14	11	9	7	5	4	3	2	2	0	0	0
102	101	98	95	91	86	80	74	67	61	54	48	41	36	30	25	21	17	14	11	9	7	5	4	3	2	1	0	0	0
99	98	96	93	89	84	78	72	66	59	53	46	40	35	29	25	20	17	13	11	8	6	5	4	3	2	1	0	0	0
96	95	93	90	86	81	76	70	64	57	51	45	39	34	28	24	20	16	13	10	8	6	5	4	3	2	1	0	0	0
93	92	90	87	83	78	73	68	62	55	49	43	38	32	27	23	19	16	13	10	8	6	5	3	3	2	1	0	0	0
90	89	87	84	80	76	70	65	59	53	48	42	36	31	26	22	18	15	12	10	8	6	4	3	2	2	1	0	0	0
86	85	83	80	77	72	68	62	57	51	46	40	35	30	25	21	18	14	12	9	7	6	4	3	2	2	1	0	0	0
82	81	79	77	73	69	65	60	54	49	44	38	33	29	24	20	17	14	11	9	7	5	4	3	2	2	1	0	0	0
78	77	76	73	70	66	62	57	52	47	42	37	32	27	23	19	16	13	10	8	7	5	4	3	2	2	0	0	0	0
74	74	72	69	66	63	58	54	49	44	39	35	30	26	22	18	15	12	10	8	6	5	4	3	2	1	0	0	0	0
70	70	68	66	63	59	55	51	46	42	37	33	28	24	21	17	14	12	9	7	6	5	3	3	2	1	0	0	0	0
66	66	64	62	59	56	52	48	44	39	35	31	27	23	19	16	13	11	9	7	5	4	3	2	2	0	0	0	0	0
62	62	60	58	56	52	49	45	41	37	33	29	25	22	18	15	13	10	8	7	5	4	3	2	2	0	0	0	0	0
58	58	56	54	52	49	46	42	38	35	31	27	23	20	17	14	12	10	8	6	5	4	3	2	2	0	0	0	0	0
54	54	52	51	48	46	43	39	36	32	29	25	22	19	16	13	11	9	7	6	4	3	3	2	1	0	0	0	0	0
50	50	49	47	45	42	40	36	33	30	27	23	20	17	15	12	10	8	7	5	4	3	2	2	1	0	0	0	0	0
47	46	45	44	42	39	37	34	31	28	25	22	19	16	14	11	9	8	6	5	4	3	2	2	0	0	0	0	0	0
43	43	42	40	38	36	34	31	28	25	23	20	17	15	12	10	9	7	6	4	3	3	2	2	0	0	0	0	0	0
40	39	38	37	35	33	31	29	26	23	21	18	16	14	11	10	8	6	5	4	3	2	2	1	0	0	0	0	0	0
36	36	35	34	32	30	28	26	24	21	19	17	14	12	10	9	7	6	5	4	3	2	2	1	0	0	0	0	0	0
33	33	32	31	29	28	26	24	22	19	17	15	13	11	9	8	7	5	4	3	3	2	2	0	0	0	0	0	0	0
30	30	29	28	27	25	23	22	20	18	16	14	12	10	9	7	6	5	4	3	2	2	1	0	0	0	0	0	0	0
27	27	26	25	24	23	21	19	18	16	14	12	11	9	8	6	5	4	3	3	2	2	1	0	0	0	0	0	0	0
24	24	24	23	22	20	19	18	16	14	13	11	10	8	7	6	5	4	3	2	2	1	0	0	0	0	0	0	0	0
22	22	21	20	19	18	17	16	14	13	11	10	9	7	6	5	4	3	3	2	2	1	0	0	0	0	0	0	0	0
20	19	19	18	17	16	15	14	13	11	10	9	8	7	6	5	4	3	2	2	1	0	0	0	0	0	0	0	0	0
17	17	17	16	15	15	14	12	11	10	9	8	7	6	5	4	3	3	2	2	1	0	0	0	0	0	0	0	0	0
15	15	15	14	14	13	12	11	10	9	8	7	6	5	4	4	3	2	2	1	0	0	0	0	0	0	0	0	0	0
14	13	13	13	12	11	11	10	9	8	7	6	5	5	4	3	3	2	2	1	0	0	0	0	0	0	0	0	0	0
12	12	12	11	11	10	9	9	8	7	6	5	5	4	3	3	2	2	1	0	0	0	0	0	0	0	0	0	0	0
11	10	10	10	9	9	8	8	7	6	5	5	4	3	3	2	2	2	1	0	0	0	0	0	0	0	0	0	0	0
9	9	9	9	8	8	7	7	6	5	5	4	4	3	3	2	2	1	0	0	0	0	0	0	0	0	0	0	0	0
8	8	8	7	7	7	6	6	5	5	4	4	3	3	2	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0
7	7	7	6	6	6	5	5	4	4	4	3	3	2	2	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0
6	6	6	6	5	5	5	4	4	3	3	3	2	2	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	5	5	5	4	4	4	4	3	3	3	2	2	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	4	4	4	4	4	3	3	3	2	2	2	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	4	4	3	3	3	3	3	2	2	2	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	3	3	3	3	3	2	2	2	2	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	3	3	2	2	2	2	2	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	2	2	2	2	2	2	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	2	2	2	2	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

APPENDIX III

DEBUGGING & ERREXIT MODES

A COMPASS multiple entry point routine (FTNUTIL) is included in the code for debugging purposes. The statement call TIMEPRT(TIME) will print the CP-time, the system time of day and return the CP-time in TIME. The CALL TRCEBCK statement works out the calling sequence in nested subroutine calls. The CALL TRCEDMP(p_1,p_2,p_3,$\cdots$,p_k), $k \leq 20$ will print the parameter list p_i either as single variables or as arrays with 3 formats, 1PE25.14, I16, and 020. In the array mode of print out 3 parameters must be given, p_i, p_{i+1}, p_{i+2} with p_i being the location where printing begins, p_{i+1} = 2HFL and p_{i+2} is an integer specifying the number of consecutive locations to be printed. The last entry point is reached with CALL ERREXIT(p_1,$\cdots$,p_k). This will first trace back then trace dump as above then dump the xfer package and 4000_8 locations around the call. Finally EXIT is called. All exits from EVALIN result from malfunctions of the program. The trace part of ERREXIT will supply the name of the subroutine and the line number. A list of these exits is given below and the problem which caused it is explained.

SUBROUTINE	*LINE*	*PROBLEMS*
EVALIN	58	TA control block on input tape was misread by REDFAT1. F array does not exist.
EVALIN	70	ICON array of EVALI should not point to a non positive RNU data set.
ADLONG	7	F array appears to be more than 16000 long, need more core.
ROCONIP	19	RO is not hermitian, suspect bad symmetry properties of the F array or erroneous matrix diagonalisation during previous iteration.
READIN	33	Last entry of current card set should be the cartesian dimension of IDSC; bad cards, or cards punched for a different basis than the present assumed one.
READIN	35	The present basis is being overwritten, see 33 above.
READIN	37	Too many or bad wavefunction cards.
CARTOJ	100	Too many states for the size of the STUCO array.
POLTRA	17	need a bigger array to handle this state.
TRIMUP	8	bad calling sequence. ND should have been larger than N.
NOLMIN	15	array CP too small.
LINCOM	12	array C appears to be too small.
SCALN2	17	array CP appears to be too small.
DISTRAE	29	array CP appears to be too small.
CNNJKTE	16	array CP appears to be too small.
EXFI1BP	41	The required state was not found, bad calling sequence to INBINS.
EXFI1BP	48	Missing state or bad matrix elements.

SUBROUTINE	*LINE*	*PROBLEMS*
HARFIT	34	RO**2.NE.RO, bad density matrix suspect diagonalisation or F array.
TRVEROJ	27	TRVEROJ and IDACSET disagree on the computed dimension of a J subspace. Suspect bad calling sequences.
VEEBAR	11,12,18	Bad F array, suspect mass storage read operation or the construction of F.
CALHNL	17	number of nodes +1 must be .GE.1.

APPENDIX IV

CONTROL OPTIONS IN BLOCK DATA

All the data cards of EVALI can be set to specify the flow of the program. IPUNCH may be set to .F. to inhibit punching of the wavefunction cards by WAVPUN. The card header is also printed when cards are punched, to facilitate identification of the cards.

The variable EPS may be increased to .1 if only the major components of the wavefunction are to be printed. The amount and detail of the printout can be increased by raising IPR up to 10. The cranking loop in NWHART will be implemented by writing the names of the one body operators etc... (see NUHART) in EXFID. ITER and ITCR give the number of iterations in the main call and in the cranking calls. TNAME, HBWA and IBLK specify the force parameters. TNAME is compared with the name read from tape. HBWA(I) should have the value of $\hbar\omega$ for the Ith block on the tape while IBLK specifies which block is to be used.

The arrays ICON(2,40) and RNU(5,40) specify the sequence of nuclei which will be treated in one run of the program. RNU(5,K) contains 5 pieces of initial data belonging to the Kth nucleus. The first 2 items are the number of protons and neutrons stored as real numbers. The next one contains the name of the nucleus, up to 10 characters left adjusted, in internal BCD mode. The last 2 items

are the real values of gamma, the non-axiality angle in degree, and β_ω, the potential deformation parameter. They are used by the overlay ELMOSE to determine the initial trial wavefunction from Nilsson's deformed ellipsoidal oscillator model.

The sequence in which these initial nuclei will be considered is contained in ICON(2,J). For example ICON(1,1) = K1,ICON(1,2) = K2, ICON(1,3) = 0 will process nuclei K1,K2 in the list of RNU, then stop. If ICON(2,KI) = 1, the initial wavefunction is computed using ELMOSE, but if ICON(2,KI) = 0 this wavefunction is read from cards by READIN. Other control values could clearly be added to obtain different sequences.

APPENDIX V

MAGNETIC TAPE CARD IMAGES OF THE CODES

A copy of a 600 foot reel of ½" magnetic tape (AECL #234-005) may be obtained by sending a similar blanked reel of tape to the Technical Information Branch, Atomic Energy of Canada Limited, Chalk River Nuclear Laboratories, Chalk River, Ontario, Canada.

The returned tape will consist of 3414 physical records of 80 external 7-track BCD characters each, recorded at a density of 556 bpi. Each record is separated by a 3/4" interecord gap and contains one card image. The listing of appendix I was made from this tape.

TABLE 1

THE SINGLE PARTICLE BASIS USED IN EVALIN

I	n	ℓ	j	N = 2n+ℓ	$Z_{cumulative}$
					82
16	1	2	3/2	4	
15	0	5	11/2	5	
14	2	0	1/2	4	
13	0	4	7/2	4	
12	1	2	5/2	4	
					50
11	0	4	9/2	4	
					40
10	1	1	1/2	3	
9	0	3	5/2	3	
8	1	1	3/2	3	
					28
7	0	3	7/2	3	
					20
6	0	2	3/2	2	
5	1	0	1/2	2	
4	0	2	5/2	2	
					8
3	0	1	1/2	1	
2	0	1	3/2	1	
					2
1	0	0	1/2	0