Ans

DL/SCI/TM 22T

technical memorandum Daresbury Laboratory

DL/SCI/TM 22T

AN ADAPTATION OF THE ALCHEMY ATOMIC AND MOLECULAR INTEGRALS PACKAGES FOR R-MATRIX ELECTRON-MOLECULAR COLLISIONS

by

J. KENDRICK* and B.D. BUCKLEY+, Daresbury Laboratory

*Now with I.C.I. Limited, Runcorn, Cheshire

+Now with Shell Internationale Petroleum Maatschappij B.V., The Hague, The Netherlands.

MAY, 1980

Science Research Council

Daresbury Laboratory

Daresbury, Warrington WA4 4AD

1. INTRODUCTION

The R-matrix approach to scattering theory was introduced into nuclear physics by Wigner and Eisenbud[1], and extended to electron-atom systems by Burke et al[2]. Recently Schneider et al[3] and Burke et al[4,5] have applied the R-matrix method to electron-molecule collisions with some significant success. It is with the implementation of the approach of Burke et al[4,5] that we concern ourselves in this report.

The basic philosophy of the R-matrix method lies in the division of space into two distinguishable regions (see fig.1). In the external region the scattering electron is far removed from the N-electron target molecule, the exchange interaction is negligibly small, and the collision process may be treated as an electron scattering from a central local potential. In the internal region, however, the scattering electron is in close proximity to the target electrons, exchange is important and the system is more correctly viewed as an (N+1)-electron molecular ion. In this report we present an approach to the solution of the scattering equations in the internal region based on the ALCHEMY[6] suite of bound state molecular codes.

In the R-matrix method, as in many other approaches to low energy electron scattering by atomic/molecular targets, a close coupling expansion of the total (N+1)-electron wavefunction (a solution to the Schrödinger equation) is made, which in quantum chemistry parlance is equivalent to a configuration interaction (CI), expansion. By enclosing the electron-molecule system within a spherical region, of radius a say, the continuum eigenspectrum of the Hamiltonian is replaced by a completely discretised eigenspectrum (see fig. 2). Consequently the solution to the Schrödinger equation in the internal region may be derived in a completely

analogous fashion to the derivation of the eigensolutions of a properly bound system: i.e.(I) we can choose some suitable atomic basis set (in our case we use Slater type orbitals (STO)), and evaluate all the necessary atomic integrals; (II) we determine a set of orthonormal molecular orbitals from the STO basis and transform the atomic integrals to molecular integrals; (III) we generate a set of configurations from the available molecular orbitals and construct the Hamiltonian matrix; and (IV) we diagonalise the Hamiltonian matrix to determine the R-matrix eigenstates and eigenvalues. These give a complete representation of the (N+1)-electron wavefunction on the boundary at r = a, and by matching to the solutions of the Schrödinger equation in the external region we can derive the scattering cross sections.

In the original molecular R-matrix programmee of Buckley et al[5] the evaluation of some of the atomic integrals (particularly those involving contracted STO's on the nuclei) was not as accurate as it might have been, and the transformation of the two electron integrals from atomic to molecular type was exceedingly slow - proportional to n^8, where n is the number of STO's - compared with the most efficient quantum chemistry codes which use n^5 algorithms. For these reasons it was decided to adopt the ALCHEMY[6] suite of codes to evaluate the atomic integrals and perform the integrals transformation described in steps (I) and (II) above.

In section 2 of this report we describe briefly the atomic and molecular integrals packages from ALCHEMY, whilst in section 3 we detail the modifications necessary to generate the integrals required by the R-matrix method. Appendices 1 and 2 outline the input data required by the modified ALCHEMY packages, and an example of both input and output for an electron-nitrogen run is given in appendix 3. Finally appendix 4 provides some

useful information for running the programme on the Daresbury Laboratory IBM 370/165.

2. ALCHEMY

The ALCHEMY suite of codes has been written and developed over many years by the Theoretical Chemistry Group at IBM[6] for the calculation of bound state properties and uses functions of linear molecules (up to 20 nuclei). The facility exists within the codes to perform both self-consistent field (SCF) Hartree-Fock (HF) and/or CI calculations using a basis set of STO's,

$$\phi_s^I = N_s r_I^{n_s-1} e^{-\alpha_s r_I} Y_{\ell_s m_s}(r_I) \qquad (1)$$

where s indexes the STO and I the nucleus on which the STO is centred.

2.1 The Atomic Integral Package

The atomic integrals required for the construction of the Hamiltonian matrix are the one and two electron integrals:

$$\int \phi_s^{I*}(1)\ P\ \phi_t^J(1) d\tau = (I_s|P|J_t) \qquad (2)$$

where the operator P is 1 for the overlap integrals, $-\frac{1}{2}\nabla^2$ for the kinetic energy integrals, and $\Sigma^+ r_I^{-1}$ for the nuclear attraction integrals; and

$$\iint \phi_s^{I*}(1)\ \phi_u^{K*}(2)\ r_{12}^{-1}\ \phi_t^J(1)\ \phi_v^L(2)\ d\tau_1\ d\tau_2 = (I_s J_t|K_u L_v), \qquad (3)$$

(sometimes the notation (s|P|t) and (st|uv) is adopted when the nuclear centres are unimportant.) The two electron integrals may be sub-divided into three different types - the one centre integral $(I_s I_t|I_u I_v)$; the Coulomb-hybrid integrals $(I_s I_t|K_u L_v)$; and the exchange integrals $(I_s J_t|K_u L_v)$ i.e. when both charge distributions are multicentred.

The general structure of the ALCHEMY integrals package is shown in fig.3. The dynamic core allocator determines the amount of core available to the programme, subtracts the space required by the input/output buffers, and allocates the remaining core to a single vector, X, which is passed through as an argument to the driving routine. The driving routine, in turn, allocates space from X for the storage of constants and for useful workspace in the evaluation of the required integrals.

2.2 The Molecular Integrals Package

The molecular integrals package takes a molecular orbital basis, $\{\psi_i\}$, where

$$\psi_i = \sum_s C_{is}\ \phi_{is} \qquad (4)$$

and transforms the atomic integrals to integrals over the molecular orbitals; i.e.

$$(i|P|j) = \sum_{st} C_{is}\ C_{jt}\ (s|P|t) \qquad (5)$$

and

$$(ij|k\ell) = \sum_{stuv} C_{is}\ C_{jt}\ C_{ku}\ C_{\ell v}\ (st|uv) \qquad (6)$$

The summation in eq. (6) contains as it stands, of the order of n^8 arithmetic operations (n being the number of STO's), and would be exceedingly expensive to compute for reasonable basis sets. The number of operations required, however, can be reduced to the order of n^5 if the summation is rearranged in the following manner

$$(it|uv) = \sum_s C_{is}(st|uv)\ , \qquad (7a)$$

$$(ij|uv) = \sum_t C_{jt}(it|uv)\ , \qquad (7b)$$

$$(ij|kv) = \sum_u C_{ku}(ij|uv)\ , \qquad (7c)$$

and finally

$$(ij|k\ell) = \sum_{v} C_{\ell v}\,(ij|kv) \quad . \tag{7d}$$

The intermediate, partially summed integrals in eq. (7) can cause storage problems and it is necessary to use complex integral sorting algorithms[7] to ensure that calculations with reasonably sized basis sets (~ 50) can be performed without unreasonable demands on either CPU time or input/output devices.

3. R-MATRIX CALCULATIONS

The R-matrix method of Burke et al[4,5] for electron scattering by diatomic molecules is based upon a three centre STO expansion of the (N+1)-electron wavefunction in the internal region. The centres are the two nuclei, A and B, and the centre-of-mass, G (see fig.1). Of the STO's, only those centred on G are allowed to be sufficiently diffuse to be non-negligible at the R-matrix boundary. Thus in the external region the scattering equations are based on a single centre, thereby considerably simplifying the solution of the Schrödinger equation in this region. Furthermore, whilst A and B are used as centres for expanding both the bound molecular orbitals (both occupied and unoccupied) and the continuum molecular orbitals, the STO's on G are reserved solely for the continuum molecular orbitals. This, as we shall see below, reduces the number of integrals which need be evaluated and simplifies the modifications required by the R-matrix method.

The difference between what is normally required of the ALCHEMY package, and the R-matrix method is that in the former integration is over all space (o, ∞), whilst in the latter integration is restricted to within a sphere of radius a centred on G.

The approach used in this adaptation of ALCHEMY for R-matrix calculations is (a) to evaluate all the integrals over the normal (0, ∞) range, (b) determine the 'tail' contributions to these integrals coming from the range (a, ∞) and (c) subtract the tails from the (o, ∞) integrals. Step (a) represents a normal run of ALCHEMY. Steps (b) and (c) constitute the modifications made to ALCHEMY and are now described.

3.1 The Tail Integrals

Cognisent of our definitions of bound orbitals (i.e. two centred, A and B) and continuum orbitals (i.e. three centred), the fact that only G centred STO's are non-negligible at r > a, and that in a scattering process only one electron may be in the continuum, then the only required atomic integrals which will have a contribution coming from outside the internal region are the one electron $(G_s|P|G_t)$ integrals, and the two electron coulomb-hybrid $(I_s J_t|G_y G_v)$ integrals, where I and J are either A or B. All the remaining one and two electron integrals possess no contribution from the external region. We note that two electron integrals containing more than two G-centred STO's are not required since only one electron is ever in the continuum.

3.1.1 The one electron tail integrals

The one electron tail integrals are single centred and are given by

$$(G_s|G_t)_{r>a} = \delta_{\ell_s\ell_t}\delta_{m_s m_t} S_{st}(n_s,n_t) = \delta_{\ell_s\ell_t}\delta_{m_s m_t} N_s N_t \int_a^\infty r^{n_s+n_t} e^{-(\alpha_s+\alpha_t)r}\,dr$$

$$= \delta_{\ell_s\ell_t}\delta_{m_s m_t} N_s N_t \exp(-(\alpha_s+\alpha_t)a) \sum_{k=0}^{(n_s+n_t)} \frac{(n_s+n_t)!}{k!} \frac{a^k}{(\alpha_s+\alpha_t)^{n_s+n_t-k+1}} \tag{8}$$

$$(G_s|-\frac{1}{2}\nabla^2|G_t)_{r>a} = -\frac{1}{2}\delta_{\ell_s\ell_t}\delta_{m_s m_t}\left[(n_t+\ell_t)(n_t-\ell_t-1)S_{st}(n_s,n_t-2)\right.$$
$$\left. - 2n_t\alpha_t S_{st}(n_s,n_t-1) + \alpha_t^2 S_{st}(n_s,n_t)\right] \qquad (9)$$

and

$$(G_s|-Z_A/r_A - Z_B/r_B|G_t)_{r>a}$$
$$= (-1)^{m_s+1}\left[(2\ell_s+1)(2\ell_t+1)\right]^{1/2}\sum_{L=|\ell_s-\ell_t|}^{|\ell_s+\ell_t|}\begin{pmatrix}\ell_s & L & \ell_t\\ -m_s & 0 & m_t\end{pmatrix}\begin{pmatrix}\ell_s & L & \ell_t\\ 0 & 0 & 0\end{pmatrix}$$
$$\times\left[(-1)^L Z_A R_A^L + Z_B R_B^L\right]\delta_{st}(n_s,n_t-L-1) \qquad (10)$$

where N_s is the normalisation constant for the s^{th} STO over the range (o,∞), Z_A and Z_B are the nuclear changes, and R_A and R_B are the displacements of the two nuclei from the centre-of-mass, G.

It is apparent from eqs. (8-10) that the evaluation of the one electron tail integrals is straightforward, requiring only the calculation of overlap integrals of the type S_{st} (eq.8).

3.1.2 The two electron tail integrals

The only two electron tail integrals are of the coulomb-hybrid type

$$(I_s J_t|G_u G_v)_{r_2>a}$$
$$= (-1)^{m_t}\left[(2\ell_s+1)(2\ell_t+1)\right]^{1/2}\sum_{L=|\ell_s-\ell_t|}^{\ell_s+\ell_t}\frac{(L-M)!}{(L+M)!}^{1/2}\begin{pmatrix}\ell_s & L & \ell_t\\ -m_s & M & m_t\end{pmatrix}\begin{pmatrix}\ell_s & L & \ell_t\\ 0 & 0 & 0\end{pmatrix}$$
$$\times(I_s|r_G^L P_{LM}(\cos\theta_G)\, e^{-iM\phi}|J_t)\, S_{uv}(n_u,n_v-L-1) \qquad (11)$$

where I and J are either A or B. Again, the calculation of integrals of the type eq. (11) are relatively simple, requiring only the single centre tail overlaps $S_{uv}(n_u,n_v-L-1)$, and the one or two centre multipole integrals

$$(I_s|r_G^L P_{LM}(\cos\theta_G)\, e^{-iM\phi}|J_t)\ . \qquad (12)$$

Integrals (eq. 12) are over all space and can be evaluated directly within the standard ALCHEMY package.

3.2 Normalisation

In all the integrals so far mentioned the STO's are normalised over all space, i.e. (o,∞), by the factor N_s (see eq.1). In the R-matrix approach, however, normalisation is required within the sphere (O,a), which means in effect that the diffuse G-centred STO's are incorrectly normalised. (The A and B centred STO's are unaffected since they are by definition negligible by the R-matrix boundary). The overlap integral in the internal R-matrix region is given by

$$(G_s|G_t)_a = (G_s|G_t) - S_{st} \qquad (13)$$

so that the STO G_s is correctly normalised by multiplying it by the factor $[1 - S_{ss}]^{-1/2}$.

3.3 Modifications to the ALCHEMY Atomic Integrals Package

As far as possible the modifications required for the tail integrals evaluation were incorporated in the ALCHEMY package so as to mimic the structure of the original code. This was achieved by duplicating original subroutines and modifying them to a form suitable for the tail integrals. Table 1 gives a summary of these, together with their counterparts in the original code. The changes required in the one electron integral subroutines were straightforward, being a direct implementation of eqs.(8-10). The modifications necessary for the two electron tail integrals were more complex.

The first step in evaluating the two electron tail integrals (eq.11) is to calculate the multipole integrals (eq.12). This is done using the

original ALCHEMY package, since the multipole integrals are just a subset of the property integrals which ALCHEMY evaluates. These integrals are stored sequentially in blocks with the same M value but all possible L to facilitate the calculation of the two electron tail integrals (eq.12).

The original ALCHEMY routines have been modified to prevent the calculation of unwanted integrals. These are the one electron nuclear attraction integrals on centre G, and all two electron integrals involving more than two G centred STO's.

The scaling of the atomic integrals required by the renormalisation of the G-centred STO's is performed in subroutine TAILNM. Four sets of integrals are processed: the one electron and two electron integrals and their respective tails. At this stage the tail integrals are further multiplied by - 1 in preparation for their subtraction from the (0, ∞) integrals.

Subroutine TAILNM also reorders the atomic orbital set so that STO's with the same m-quantum number are grouped together, and in the order m = 0, 1, 2 etc. This is the order required by the transformation programme. This reordering is required as it is not always possible to ensure this ordering in the input data to the atomic integrals programme. In this latter case it is imperative that the STO's are ordered such that the target atomic orbitals (i.e. A and B centred STO's) come first, followed by the G-centred functions. This ordering ensures that the multipole integrals (eq.12) required by the two electron tail integrals (eq.13) are stored in the correct manner.

3.3.1 Subroutine LINT

Subroutine LINT is the main driving routine of the programme and has

been extensively modified. It sets up the areas for work space and storage in the large dynamically allocated vector, X, and calls the main routines for the evaluation of the one and two electron integrals. Calls have been inserted to the subroutines

TSTUMX	-	one electron tail integrals
TTEOCW	-	two electron tail integrals
TAILNM	-	integral renormalisation
TRNSDR	-	molecular orbital vector manipulation

(The vector manipulation routine TRNSDR is discussed in section 3.4.)

In addition to these calls use is made of calls to entry points in other subroutines to set up arrays which remain static throughout the later subroutine calls. This use of entry points lowers the overheads of later calls to subroutines as fewer subroutines arguments are needed. However, it does involve the implementation of uniquely IBM type facilities and is incompatible with other machines such as CDC where use of entry points is very restricted.

A flow chart for the modified ALCHEMY atomic integrals package is given in fig.4.

3.4 Molecular Orbital Generation

Part of the usual input to any R-matrix calculation is the target wavefunction(s), be it (they) HF or CI, generated from an orthonormal set of molecular orbitals ($\underset{\sim}{C}_{AB}$, say) which frequently includes additional unoccupied or virtual orbitals. The problem we address in this section is that given these molecular orbitals, how do we generate the additional molecular orbitals from the G-centred STO's. One straightforward approach is by Schmidt orthoganalisation of each of the G-centred STO's in turn to the previous molecular orbitals until the maximum number of molecular orbitals have been generated. In this way a complete set of orthonormal

molecular orbitals is obtained for the (N+1)-electron scattering problem which leaves the molecular orbitals of the N-electron target wavefunction unaltered.

A disadvantage of the above method is that it provides no way of identifying any numerical problems which might exist with the chosen STO basis, other than by visual inspection of the vector coefficients. A more suitable means of ascertaining any numerical problems which may arise is by canonical orthogonalisation of the additional molecular orbitals. Given the target set of molecular orbitals, $\underline{C}_{AB}$, a complete non-orthogonal set $\underline{C}$, which includes all the STO's is

$$\underline{C} = \begin{pmatrix} \underline{C}_{AB} & 0 \\ 0 & \underline{C}_G \end{pmatrix} \tag{14}$$

where $\underline{C}_G$ is any arbitrary vector set, usually chosen to be $\underline{1}$. The individual vectors in $(0\ \underline{C}_G)^+$ can be made orthogonal to $\underline{C}_{AB}$ by Schmidt orthogonalisation (this step is taken to avoid altering $\underline{C}_{AB}$ in what follows), giving us

$$\underline{C}_1' = \begin{pmatrix} \underline{C}_{AB} & \underline{C}_1 \\ 0 & \end{pmatrix} \tag{15}$$

where the vectors in $\underline{C}_1$ are now each orthogonal to $\underline{C}_{AB}$, but not with each other.

Taking S as the atomic overlap matrix, we define a new overlap matrix

$$\underline{S}_1 = \underline{C}_1^+ \underline{S}\ \underline{C}_1 \tag{16}$$

whose eigenvectors $\underline{C}_2$, given by

$$\underline{S}_1\underline{C}_2 = \underline{C}_2\ \underline{E}, \tag{17}$$

are orthogonal to each other. They can be rendered orthonormal by dividing $\underline{C}_2$ by $\underline{E}^{1/2}$, i.e.

$$(\underline{C}_2\ \underline{E}^{-1/2})^+\ \underline{S}_1(\underline{C}_2\ \underline{E}^{-1/2}) = \underline{C}_3^+\ \underline{S}_1\ \underline{C}_3 = 1 \tag{18}$$

Finally we can transform back to the original atomic basis by taking the matrix product of $\underline{C}_1$ and $\underline{C}_3$ to give us

$$\underline{C}_4 = \underline{C}_1\ \underline{C}_3 \tag{19}$$

The final set of orthonormal molecular orbitals is

$$\underline{C}^1 = \begin{pmatrix} \underline{C}_{AB} & \underline{C}_4 \\ 0 & \end{pmatrix} . \tag{20}$$

Numerical problems can arise with the normalisation constant $E^{-1/2}$ when the matrix S_1 has very small eigenvalues. The numerical problems arise at the two and four index transformation stage where the integrals are of the order of E^{-1} and E^{-2} respectively. It has been found from experience that eigenvector corresponding to eigenvalues smaller than 5×10^{-5} often give rise to inaccuracies in the molecular intgrals due to either the word length of the computer, or to errors in the evaluation of the atomic integral due in turn to unavoidable series truncation or poor integration grids. Upon transformation one is frequently left with some error term multiplied by larger coefficients.

3.4.1 Vector manipulation

A set of routines for manipulating the molecular orbitals as described in eqs. (14-20) have been added to the atomic integrals package (see fig. 4). Information for the target wavefunction(s) is read by subroutine TRNSDR, whilst COMMUN reads the data on the additional molecular orbitals to be generated from the G-centred STO's. Subroutine ENLARG performs the actual orthogonalisation.

An additional feature in TRNSDR is the facility to read in the target wavefunction(s) from the ATMOL3 suite of programmes[8].

Table 2 briefly describes the subroutines used in the manipulation of

the molecular orbitals. As far as was possible the data input follows ALCHEMY convention and has the same format and variable names as the HF and transformation packages in ALCHEMY.

3.5 The Integral Transformations

Having evaluated all the required atomic integrals it remains to transform them to molecular integrals over the molecular basis $\{\psi_i\}$. This involves a two and four index transformation respectively of the one and two electron integrals.

3.5.1 The one electron integrals

Since the Hamiltonian for the electron-molecule system is being treated within a sphere and not over all space, the kinetic energy matrix is no longer hermitian and the standard two index transformation routines from the ALCHEMY molecular integral package cannot be used. As the one electron transformation (eq. 5) is not very time consuming it has been included in the subroutine ENLARG in the atomic integrals package.

An alternative procedure would be to add the Bloch operator (see[4,5]) to the Hamiltonian at the atomic integrals stage. The combined $(H+L_b)$ matrix is hermitian and can therefore be transformed in the standard fashion. This approach would remove, however, the flexibility of choice of the Bloch term at an earlier stage in the R-matrix calculation than is sometimes desirable. It has, therefore, not been implemented.

3.5.2 The two electron integrals

The ALCHEMY molecular integrals package has been modified to allow the atomic tail integrals to be subtracted from the renormalised $(0, \infty)$-integrals. The modification required was very simple and involved the changing of two lines of code.

The transformation programme reorders the two electron integrals (using a direct access device if sufficient core is unavailable) so that required subsets of the integrals can be brought into core at different stages in the transformation (eq. 7). The position of these integrals in the integral list is uniquely determined by their orbital indices, and so if two integrals with the same indices are given to the unmodified code (the integral and its tail for instance) then the second integral will overwrite the first.

The modification to the code consists of adding rather than overwriting the integrals in their storage location (which must now be preset to zero for each subset of integrals). The changes are made in the subroutines ORDRB and ORDRC and involve replacing

```
    40 COR(I)   = BOX(N)
```

by

```
    40 COR(I)   = COR(I) + BOX(N)
```

and

```
       COR(INT) = XINT
```

by

```
       COR(INT) = COR(INT) +XINT
```

respectively.

4. INTERFACE WITH THE R-MATRIX CODES

To evaluate the Hamiltonian matrix and subsequently calculate the R-matrix on the boundary of the internal region, the original codes of Buckley et al[5] were used. An interface is required to transform the information on the atomic and molecular basis to the R-matrix codes, and to sort the one and two electron molecular integrals into the order used by these codes. Furthermore the Bloch terms need to be determined to render the Hamiltonian matrix hermitian, and the N-electron multipole integrals transferred to the R-matrix codes for the external region in order to derive the asymptotic interaction potentials. These operations are performed

by the program STG2II.FORT to be found under userid XH. A compiled version of this is kept in XH.LOAD(STG2II), and the required JCL in XH.RMATTI.CLIST(STG2II).

ACKNOWLEDGEMENTS

We should like to acknowledge the help given by Dr. G. Raseev in obtaining the ALCHEMY codes and writeups, and in its initial running. We also thank the IBM Theoretical Chemistry Group for allowing us to use their programmes. Much useful advice and help was given by Dr. U. Lamanna in the writing of certain sections of the codes.

APPENDIX 1

A. The Atomic Integral Data

Date is input into the R-matrix atomic integrals package via three namelists:

&INPUT - atomic integrals data

&GET - target molecular orbital data

&PUT - final molecular orbital data

Each namelist must be terminated by &END, and the first column in each line of data must be left blank. Each piece of data must be separated by a comma. An additional namelist &GET1 is required if the target molecular orbital data are taken from file FT27, having been generated in a previous run of the ALCHEMY SCF programme.

&INPUT

NAME 'title'

NEF total number of atomic orbitals

NNUC number of atomic orbital centres

GEONUC coordinate positions of atomic orbital centres

NLMK the quantum numbers associated with each STO - (n,ℓ,m,k)
n - principle quantum number,
ℓ - azimuthal quantum numbr,
m - magnetic quantum number (i.e. σ, π, δ, etc.)
k - atomic orbital centre,

ZETA Slater orbital exponents

IFLINT a,b,c,d,e,f,g: either 0 or - 1. 0 means calculate the integral, - 1 do not. (Default is 0)
a - Coulomb hybrid
b - one centre two electron
c - exchange
d - one electron
e - property
f - tails and renormalisation
g - vector manipulation

I2PCDE 4,n,i,j,k,ℓ,m,0
to calculate the property integrals of the operator

$$r_k^n \sin^i \theta_k \cos^j \theta_k P_{\ell m}(\cos \theta_k) e^{im\phi}$$

(Note: the property integrals required by the two electron tail integral are calculated by default.)

ICORD Order of atomic orbital centres in Coulomb integral calculations. Should be ordered as to Coulomb change on each centre - largest first. Default is the order specified in GEONUC.

NUCCEN Centre-of-mass

RMATR R-matrix radius

IPRINT a,b,c,d,e,f; either 0 or 1: 0 means no printout of atomic integrals, 1 means print. a-f as in IFLINT.

&END

Note: The numerical integration grid can be modified if desired, but the default grid is believed to be suitable for STO's with exponents within the range (0.5, 50.0).

&GET

ITCVI Controls the input of the target molecular orbital vectors. = 0: vectors read in with VCIN; = 1: vectors are taken from either ATMOL3 dumpfile or from ALCHEMY SCF run in which case namelist &GET1 is required.

IATMOL ED0,ED1..ED7,MT0,MT1..MT7. If vectors are to be read from ATMOL3 dumpfile.

IBLOCK Starting block of the dumpfile on ATMOL3 data set.

ISEC Section of the ATMOL3 dumpfile containing the required vector set.

NSYM The number of symmetries used in the target basis set.

NBF(I) I=1,NSYM: the number of target STO's in each symmetry set.

NOB(I) I=1,NSYM: the number of molecular orbitals in each symmetry set.

VCIN(I,J) {{(I=1,NBF(L)),J=1,NOB(L)),L=1,NSYM}: Molecular orbital vectors. Required when ITVCI=0.

&GET1 (only needed when ITVCI=1 and IATMOL is not specified.)

IVCS Data file containing the target molecular orbital vectors. Default is IVCS=27.

NIND Number of quartets in IND required to define the vectors.

IND (a,b,c,d): a - the set number
b - symmetry type (i.e. σ,π,δ etc)
c - the starting orbital
d - the final orbital

The vectors for orbitals c to d of symmetry b are taken from set a on file IVCS.

NBASE increment all set numbers a in IND by NBASE.

IPAUL Must be set = 0.

&END

Note: IFLSYM should never be used.

&PUT

ITVCI Controls output of the final molecular orbitals: = 0 output molecular orbital vectors onto file MEGU. This is the recommended procedure as it allows a greater flexibility in the selection of molecular orbital vectors. = n where n > 2. Vectors are directed to unit IVCS. (Default ITVCI=0).

MEGU Fortran file used to store final vectors if ITVCI=0. (Default value 27)

IVCS Fortran file used to store final vectors if ITVCI>2. (Default value 21)

ISTOST Control positioning of vectors on MEGU if ITVCI=0.
=0 stored as last set of vectors.
=n stored as n^{th} set of vectors.
As MEGU is a sequential file, all vector sets stored after those being written will be lost. Conventionally the target vectors are stored as set 1 and ISTOST is set equal to 2 for the final set. If ISTOST=0 is used, the input data for the transformation programme will have to be changed for each run.

IORTHO Controls the orthogonalisation scheme used to generate the complete set of molecular orbitals
=0 canonical orthogonalisation (recommended)
=1 Schmidt orthogonalisation

CHARG(I) I=1,NNUC Coulomb charge on the atomic orbital centres.

NSYM, NBF, NOB - as in &GET

&END

B. Atomic Integrals Fortran Files

The fortran file used by the atomic integrals package are summarised

in table 3. The one electron integrals are read in as

```
READ(N) CW,MC,(XC(J),J=1,NC),(INDC(J),J=1,MC)
```

where

```
REAL*8 CW,XC
INTEGER*4 MC
INTEGER*2 INDC
```

CW contains 8 2-byte integers a,b,c,d,e,f,g,h, where a dasignetes the type of integral:

- a=1 kinetic energy
- =2 overlap
- =3 nuclear attraction
- =4 property integral
- =5 kinetic energy tails
- =6 overlap tails
- =7 nuclear attraction tails;

and b-h designate the operator

$$r_e^b \sin^c \theta_e \cos^d \theta_e Y_{fg}(\theta_e,\phi).$$

Atomic orbital centres are denoted by e. The number of integrals in the record is MC, XC are the integrals, and INDC holde 2 1-byte integers (hexadecimal) which are the orbital indices for the integer XC.

For the two electron integrals

```
READ(N)MM,NIT,(MJS(K),NIM(K),K=1,MM),
       (XC(K),K=1,NIT),(IC(K),K=1,NIT)
```

where

- MM is the total number of different M quantum number blocks stored in the records
- NIT total number of integrals
- MJS M quantum number of each block
- NIM Number of intgrals in each block
- XC the integrals
- IC 4 1-byte hexadecimal integers denoting the orbital indices for XC

APPENDIX 2

A. The Molecular Integral Data

The input data for the molecular integral package is specified in two namelists, &INTRN and &GET. &INTRN controls the ordering and transformation of the atomic integrals, whilst &GET is used when ITVCI=1.

&INTRN

IORDR =3 for present case. Specifies that all integrals are to be reordered.

ITRNS =3. Specifies that all integrals are to be transformed.

NTDRG Number of records of length LPRDIN on fortran file 20.

LPRDIN Logical record length of file 20.

SYMTYP =2, for STO's.

THINT Any integral less than THINT is treated as zero (default THINT=5×10^{-8}).

NAME 'title'

ITVCI Specifies location of molecular orbitals.
=1 vectors come from IVCS and &GET must be included.
=n, n>1 fortran file 21 contains vectors as n^{th} set. &GET is not required.

NPFLG Controls printing of molecular integrals
=-2 suppress all printing
=-1 print only final vectors
=0 print input data, input and final vectors
=2 print transformed integrals

NOORTH =0 orthogonalise input vectors
≠0 do not orthogonalise input vectors
(default =0)

CHARG Coulomb changes on atomic orbital centres

NSYM, NBF, NOB - as in &GET for atomic integrals package

&END

&GET same as &GET1 for atomic integrals package.

B. Molecular Integrals Fortran Files

The one and two electron atomic integrals required by the transformation programme are on FT16F001 (see table 4) and may be read using

```
1)   READ(16)NAME,NMVL,NBFT,NNUC,NCODT,LTRB1,
     *     (NBF(I),I=1,NMVL),(NPBF(I),I=1,NBFT),
     *     (MBF(I),I=1,NBFT),(CHARG(I),I=1,NNUC),
     *     (GEONUC(I),I=1,NNUC),SCFUSE,LPKFGO,NBLK,
     *     (JUNK(I),I=1,3)
```

where

NAME(33) is the title

NMVL number of symmetries in the molecular orbital set

NBFT number of molecular orbitals

NNUC number of atomic orbital centres

NBF number of molecular orbitals in each symmetry block

MBF symmetry of each symmetry block

CHARG the coulomb charge on each atomic orbital centre

GEONUC the position of each atomic orbital centre

```
2)   DO 15 I=1,2
     DO 15 J=1, JMVL
  15 READ(16) ICA,ICB,NN, (BUF(K),K=1,NN)
```

this reads the overlap integrals, I=1, and the kinetic plus nuclear attraction integrals, I=2.

```
3)   READ(16,END=50)M,M1,M2,M3,M4,IMAX,JMAX,KMAX,LMAX,NPQ,NRS,NCODI,
    *NCODA,NCODB

     L=1
     K=0
     IPQ=NPQ
     DO 20 IRS=1,NRS
     K=K+1
     IF(NCODB.EQ.1.AND.K.LE.KMAX) GO TO 23
     IF(NCODB.EQ.3.AND.K.LE.L) GO TO 23
     K=1
     L=L+1
23   IF(NCODI.EQ.1) IPQ=IRS
     READ(16)(BUF(I),I=1,IPQ)
     J=1
     I=0
     DO 20 IX=1, IPQ
     I=I+1
     IF(NCODA.EQ.1,AND.I.LE.IMAX) GO TO 21
     IF(NCODA.EQ.3.AND.I.LE.J) GO TO 21
     I=1
     J=J+1
21   CONTINUE
20   CONTINUE
```


where

$$M = |\underline{M}1 + \underline{M}2| = |\underline{M}3 + \underline{M}4|$$

and M1,M2,M3,M4 are the absolute magnetic quantum numbers of the four orbitals.

APPENDIX 3

A. Example of JCL and Input Data

Input for calculating the integrals required for an electron nitrogen scattering calculation in the $^2\pi_g$ scattering channel using the minimum STO basis of Ransil[9] for the target wavefunction. The R-matrix radius is at 10 a.u., two additional nuclear centred dπ STO's are included and 6 dπ G-centred STO's used. A Schmidt orthogonalisation (IORTHO=1) of the additional STO's is performed.

EXAMPLE OF INPUT

```
//IDN2RS1 JCB (ACCNO,ID),USER,MSGCLASS=T,MSGLEVEL=1,NOTIFY=ID
//JOBLIB DD DSN=JCK.LOAD,DISP=SHR
// EXEC FGG,LIBRARY='UCK.LOAD',MEMBER=JKTAILS,TIME=(4,49),
//   REGION=360K
//G.FT08F001 DD DSN=ID.V2.ST1N2P7,DISP=(OLD,PASS)
//G.FT08F002 DD DSN=&&JKC82,UNIT=3330,VOL=SER=ONPL33,SPACE=(CYL,(1,1)),
//              DISP=(NEW,PASS),DCB=(RECFM=VBS,BUFNO=1,BLKSIZE=7294)
//G.FT09F001 DD DSN=&&JKC9,UNIT=3330,VOL=SER=ONPL33,SPACE=(CYL,(1,1)),
//              DISP=(NEW,PASS),DCB=(RECFM=VBS,BUFNO=1,BLKSIZE=7294)
//G.FT10F001 DD DSN=&&JKC10,UNIT=3330,VOL=SER=ONPL33,SPACE=(CYL,(1,1)),
//              DISP=(NEW,PASS),DCB=(RECFM=VBS,BUFNO=1,BLKSIZE=7294)
//G.FT11F001 DD DSN=&&JKC11,UNIT=3330,VOL=SER=ONPL33,SPACE=(CYL,(1,1)),
//              DISP=(NEW,PASS),DCB=(RECFM=VBS,BUFNO=1,BLKSIZE=7294)
//G.FT15F001 DD DSN=ID.V2.STIN2P3,DISP=(OLD,PASS)
//G.FT27F001 DD DSN=ID.V2.STON2P,DISP=OLD
//G.FT12F001 DD DSN=ID.V2.ST1N2R1,DISP=(OLD,PASS)
//G.FT13F001 DD DSN=ID.V2.STIN2R2,DISP=(OLD,PASS)
//G.FT21F001 DD DSN=ID.V2.STIN2R6,DISP=(OLD,PASS)
//G.SYSIN DD *
 &INPUT
  NAME ='N2 RANSIL HF WFN WITH PI-D CONTINUUM STOS',
  GEONUC= -1.034, 0.0,  1.034,
  NEF=16,
  NNUC=3,
  NLMK=1,0,0,1,1,0,0,3,
       2,0,0,1,2,0,0,3,
       2,1,0,1,2,1,0,3,
       2,1,1,1,2,1,1,3,
       3,2,1,1,3,2,1,3,
       5,2,1,2,5,2,1,2,
       6,2,1,2,6,2,1,2,
       8,2,1,2,8,2,1,2,
  ZETA=6.7,6.7,1.95,1.95,1.95,1.95,1.95,1.95,
       3.8,3.8,4.0,2.0,1.2,1.2,0.8,0.5,
  NUCCEN=2,
  RMATR=10.0,
  IFLOUT=1,
  IPRINT=1,0,0,1,0,1,1,
  IFLINT=0,0,0,0,0,0,0,
  NOSCF=1,
  IFLSYM=1,
 &END
 &GET
  NSYM=2,
  NBF=6,4,
  NOB=6,4,
  ITVCI=0,
  VCIN= .70447243, .70447243, .00842448, .00842448, .00182260,-.00182260,
       -.16890461,-.16890461, .48828181, .48828181, .23971186,-.23971186,
        .06210650, .06210650, .40579949, .40579949,-.60323294, .60323294,
        .70436650,-.70436650, .01972063,-.01972063, .00856676, .00856676,
       -.16148134, .16148134, .74122040,-.74122040,-.26580588,-.26580588,
       -.10969141, .10969141, 1.2069738,-1.2069738, 1.2162466, 1.2162466,
        .83451689,-.83451689, .0       ,  .0      ,
        .62450235, .62450235, .0       ,  .0      ,
       0.0       , 0.0      , 1.0      , 1.0      ,
       0.0       , 0.0      ,-1.0      , 1.0      ,
 &END
 &PUT
  NSYM=2,
  NBF=6,10,
  NOB=6,10,
  CHARG=7.0,0.0,7.0,
  ITVCI=2,IORTHO=1,
 &END
/*
//TRA EXEC PGM=TRANSR,TIME=(2,59),REGION=360K
//FT06F001  DD  SYSOUT=A
//FT08F001  DD  DSN=ID.V2.ST1N2P7,DISP=SHR
//FT08F002  DD  DSN=ID.V2.STIN231,DISP=SHR
//FT09F001  DD  DSN=ID.V2.STIN232,DISP=SHR
//FT21F001  DD  DSN=ID.V2.STIN2R6,DISP=SHR
//FT20F001  DD  DSN=&&LU20,UNIT=TEMP,SPACE=(TRK,1200),
//   DISP=(NEW,PASS),DCB=BUFNO=1
//FT16F001  DD  DSN=ID.V2.STIN234,DISP=OLD
```

```
//FT17F001  DD DSN=ID.V2,ST1N2RS,DISP=OLD
//FT18F001  DD  DSN=&&LU18,UNIT=TEMP,SPACE=(CYL,(1,1)),
//   DCB=(RECFM=VBS,BLKSIZE=7294)
//FT19F001  DD  DSN=&&LU19,UNIT=TEMP,SPACE=(CYL,(1,1)),
//   DCB=(RECFM=VBS,BLKSIZE=7294)
//FT20F001  DD  DSN=&&LU20,UNIT=TEMP,SPACE=(CYL,(1,1)),
//   DCB=(RECFM=VBS,BLKSIZE=7294)
//FT05F001  DD  *
 &INTRN
   NAME='4-INDEX TRANSF, N2 MINIMAL SET WITH ADDED PI CONTINUUM',
   IORDR=3,ITRNS=3, NTDRG=6, LPRDIN=3250,
   SYMTYP=0, NBF=6,10,     NOB=6,10,
   CHARG=7.0,0.0,7.0, NSYM=2, NPFLG=1,    ITYCI=2,
 &END
/*
// EXEC RELEASE,JOB=IDN2RS2
//
```

EXAMPLE OF OUTPUT

ALCHEMY LINEAR MOLECULAR INTEGRALS PROGRAM

VERSION MODIFIED TO CALCULATE TAIL INTEGRALS

N2 RANSIL HF WFN WITH PI D CONTINUUM STOS

Z-COORDINATES OF ATOMIC NUCLEI

1	-0.1034000C0000000D+01
2	0.0
3	0.1034000C0000000D+01

SLATER TYPE ORBITALS

		N	L	M	K	ZETA
1	1	1	0	0	1	0.6700000000000000D+01
2	2	1	0	0	3	0.6700000000000000D+01
3	3	2	0	0	1	0.1950000000000000D+01
4	4	2	0	0	3	0.1950000000000000D+01
5	5	2	1	0	1	0.1950000000000000D+01
6	6	2	1	0	3	0.1950000000000000D+01
7	1	2	1	1	1	0.1950000000000000D+01
8	2	2	1	1	3	0.1950000000000000D+01
9	3	3	2	1	1	0.3800000000000000D+01
10	4	3	2	1	3	0.3800000000000000D+01
11	5	5	2	1	2	0.4000000000000000D+01
12	6	5	2	1	2	0.2000000000000000D+01
13	7	5	2	1	2	0.1200000000000000D+01
14	8	8	2	1	2	0.1200000000000000D+01
15	9	8	2	1	2	0.8000000000000000D+00
16	10	8	2	1	2	0.5000000000000000D+00

ATOMIC INTEGRATION PARAMETERS

FILE STRUCTURE FOR ATOMIC INTEGRALS

FT08F001 RESTART INFO
FT08F002 2-E INTEGRALS 0 TO INF
FT09F001 1 E INTEGRALS 0 TO INF INCLUDING PROPERTIES
FT10F001 2-E TAIL INTEGRALS
FT11F001 1-E TAIL INTEGRALS
FT12F001 2-E MERGED RENORMALISED AND RELABELLED INTEGRALS
FT13F001 1-E INTEGRALS NORMALISED O-A AND RELABELLED

```
FT15F001 1-E TRANSFORMED MOLECULAR INTEGRALS
FT21F001 OUTPUT FILE FOR VECTORS
FT27F001 INPUT  FILE FOR VECTORS

NO. OF NON-CONTINUUM ORBITALS  10

COULOMB AND HYBRID INTEGRALS NORMALISED FROM 0 TO INF
-----------------------------------------------------

        3       246         0       108         1       102         2        36

 0B070101   -0.6051445895D+00   0B090101   -0.1324031350+00
 0C070101    0.3244878100+00    0C090101    0.1009257840+00
 0DC70101   -0.1037966580+00    0D090101    0.3893952830-01
 0E070101   -0.1039055860-01    0E090101    0.2935888150-02
 0F070101    0.1257864010 02    0F090101    0.2682241440-03
 100701 01  -0.7615331730-04    10090101    0.1140745560-04
 0B070301   -0.1342260670+00    0B090301   -0.2895827310-01
 0C070301    0.7256725320 01    0C090301    0.2276860710-01
 0D070301   -0.2350043860-01    0D090301    0.8806923390-02
 0E070301   -0.2366433190 02    0E090301    0.6673156700-03
 0F070301    0.2557060320 03    0F090301    0.6102410950-04
 10C70301   -0.1735400310-04    10090301    0.2596746760-05

REORDERING OF BASIS FUNCTIONS ACCORDING TO M VALUE
--------------------------------------------------

NEW OLD   N   L   M   K          ZETA
  1   1   1   0   0   1   0.6700000000000000D+01
  2   2   1   0   0   3   0.6700000000000000D+01
  3   3   2   0   0   1   0.1950000000000000D+01
  4   4   2   0   0   3   0.1950000000000000D+01
  5   5   2   1   0   1   0.1950000000000000D+01
  6   6   2   1   0   3   0.1950000000000000D+01
  7   7   2   1   1   1   0.1950000000000000D+01
  8   8   2   1   1   3   0.1950000000000000D+01
  9   9   3   2   1   1   0.3800000000000000D+01
 10  10   3   2   1   3   0.3800000000000000D+01
 11  11   5   2   1   2   0.4000000000000000D+01
 12  12   5   2   1   2   0.2000000000000000D+01
 13  13   5   2   1   2   0.1200000010000000D+01
 14  14   8   2   1   2   0.1200000000000000D+01
 15  15   8   2   1   2   0.8000000000000000D+00
 16  16   8   2   1   2   0.5000000000000000D+00

1 AND 2-E INTEGRALS NORMALISED OVER THE RANGE 0 TO      10.000
-------------------------------------------------------------

000100C0000000000000000000000000        70

 0101   2.2445000000D+01    0103   -7.1263393730-01
 0303   6.3375000000D-01    0505    1.9012500000+00
 0707   1.9012500000+00     0909    7.2200000000+00
 0B0B   3.0222222222D+00    0B0C    6.2783045110-01
 0B00   4.5180356620-02     0B0E   -1.3347106850-02
 0B0F  -2.6136307100-03     0B10   -5.7276857790-04
 0C0C   7.5555556780 01     0C0D    2.8312686910-01
 0C0E   4.4635007500-04     0C0F   -1.3122464330-02

OLD BASIS SET
-------------

NSYM =   2
ITVC1=   0
LTAB = 449
NBF=   6   4
NOB=   6   4
SYMMETRY  0 BASIS FUNCTIONS  1  2  3  4  5  6
SYMMETRY  1 BASIS FUNCTIONS  7  8  9 10

NEW BASIS SET DATA
------------------

NSYM=   2
ITVC1=   2
LTAB = 449
TVCS =  21 NFONE=  13 NFTRAN=  15 LREC=7200 IORTHO=   0 TPROJ= 0.1000000D-04
NBF=   6  10
NOB=   6  10
CHARG=        7.000       0.0         7.000
MAP =  1  2  3  4  5  6  1  2  3  4
SYMMETRY  0 BASIS FUNCTIONS  1  2  3  4  5  6
SYMMETRY  1 BASIS FUNCTIONS  7  8  9 10 11 12 13 14 15 16
```

NON-ORTHOGONAL VECTORS WITH ADDED CONTINUUM ORBITALS

SYM= 1

	1	2	3	4	5	6
1	0.70447243	0.16890461	0.06210650	0.70436650	-0.16148134	-0.10969141
2	0.70447243	-0.16890461	0.06210650	-0.70436650	0.16148134	0.10969141
3	0.00842448	0.48828101	0.40579949	0.01972063	0.74122040	1.20697380
4	0.00842448	0.48828101	0.40579949	-0.01972063	-0.74122040	-1.20697380
5	0.00182260	0.23971186	-0.60323294	0.00856676	-0.26580588	1.21624660
6	-0.00182260	0.23971186	0.60323294	0.00856676	0.26580588	1.21624660

SYM= 2

	1	2	3	4	5	6
1	0.83451689	0.62450235	0.0	0.0	0.0	0.0
2	0.83451689	0.62450235	0.0	0.0	0.0	0.0
3	0.0	0.0	1.00000000	-1.00000000	0.0	0.0
4	0.0	0.0	1.00000000	1.00000000	0.0	0.0
5	0.0	0.0	0.0	0.0	1.00000000	0.0
6	0.0	0.0	0.0	0.0	0.0	1.00000000
7	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0

	7	8	9	10
1	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0
7	1.00000000	0.0	0.0	0.0
8	0.0	1.00000000	0.0	0.0
9	0.0	0.0	1.00000000	0.0
10	0.0	0.0	0.0	1.00000000

S**-0.5 ORTHONORMALISED VECTORS WITH S-MATRIX EIGENVALUES

SYM= 1

	1	2	3	4	5	6
	1.00000000	1.00000000	1.00000000	1.00000000	1.00000000	1.00000000
1	0.70447243	-0.16890461	-0.06210650	0.70436651	-0.16148134	-0.10969141
2	0.70447243	-0.16890461	-0.06210650	-0.70436651	0.16148134	0.10969141
3	0.00842448	0.48828181	0.40579949	0.01972063	0.74122041	1.20697384
4	0.00842448	0.48828181	0.40579949	0.01972063	-0.74122041	-1.20697384
5	0.00182260	0.23971186	-0.60323294	0.00856676	-0.26580588	1.21624663
6	0.00182260	-0.23971186	0.60323294	0.00856676	-0.26580588	1.21624663

SYM= 2

0	1	2	3	4	5	6
	1.00000000	1.00000000	1.00000000	1.00000000	3.21013744	1.79673134
1	0.83451689	0.62450235	0.18242597	0.08401834	0.08071001	0.69890135
2	-0.83451689	0.62450235	-0.18242597	0.08401834	-0.08071001	-0.69890135
3	0.0	0.0	0.79891269	-0.65703550	0.10473404	0.37888340
4	0.0	0.0	0.79891269	0.65703550	0.10473404	0.37888340
5	0.0	0.0	0.0	0.0	0.35546119	1.36935265
6	0.0	0.0	0.0	0.0	-0.22470720	-0.60998225
7	0.0	0.0	0.0	0.0	-0.26126200	-0.26519169
8	0.0	0.0	0.0	0.0	0.28898455	0.09890928
9	0.0	0.0	0.0	0.0	-0.27174644	0.25947016
10	0.0	0.0	0.0	0.0	-0.23962339	0.29081081

	7	8	9	10
	0.63582439	0.31536544	0.04045301	0.00148837
1	-3.10404045	2.51015176	0.53336069	0.08660333
2	3.10404045	-2.51015176	0.53336069	-0.08660333
3	0.83191565	0.43705331	0.09828425	0.21855531
4	-0.83191565	0.43705331	0.09828425	0.21855531
5	-3.41128720	2.02269630	0.21663314	0.82220850
6	-0.31750937	1.71335755	2.39007784	2.83811920
7	-0.55834454	0.46452037	3.26686037	5.91021150
8	0.25489006	0.81116106	2.33624723	-12.50629674
9	0.15847602	0.14249053	-1.36077531	18.97196454
10	0.42354830	0.73909515	2.21523689	-10.95637660

0001000000000000000000000000000000 136

0101	2.226008303D+01	0102	5.822372058D+00	0103	-2.386296036D+00	0104	7.632362425D-18
0105	8.667147468D-16	0106	1.551276468D-15	0201	-5.822372058D+00	0202	2.276733244D+00
0203	9.716916532D-02	0204	4.350971081D-16	0205	-2.220446049D-16	0206	-5.273559367D-16
0301	-2.386296036D+00	0302	9.716916532D-02	0303	1.985768356D+00	0304	4.461322533D-16
0305	-2.220446049D-16	0306	-4.440892099D-16	0401	-2.115922184D-16	0402	6.730727087D-16
0403	2.194425197D-16	0404	2.223351253D+01	0405	-5.824527940D+00	0406	-4.643504628D+00
0501	8.878531590D-16	0502	2.636779683D-16	0503	-1.804112415D-16	0504	-5.824527940D+00
0505	2.499892471D+00	0506	9.246299351D-01	0601	1.487525381D-15	0602	-4.996003611D-16
0603	-4.440892099D-16	0604	-4.643504628D+00	0605	9.246299351D-01	0606	4.914935860D+00
0707	2.413527161D+00	0708	0.0	0709	2.761423338D-01	070A	-1.331400268D-15
070B	-4.103024396D-01	070C	-7.014135130D-01	070D	6.920180892D-01	070E	1.292975614D+00
070F	-6.824510681D-01	0710	-3.442470009D-01	0807	0.0	080B	1.614368275D+00
0809	1.207367539D-15	080A	6.243410087D-02	080B	-5.551115123D-16	080C	-2.303712776D-15
080D	5.107025913D-15	080E	-2.706168623D-15	080F	-5.273559367D-16	0810	-1.304512054D-15
0907	2.761423338D-01	0908	-1.207367539D-15	0909	8.752233414D+00	090A	-2.220446049D-16
090B	3.940462532D-01	090C	1.432164299D+00	090D	-3.067090959D+00	090E	1.585060545D+00
090F	2.891787293D-01	0910	6.201791590D-01	0A07	-1.332267630D-15	0A08	6.243410087D-02
0A09	0.0	0A0A	6.538923618D+00	0A0B	-4.857225733D-16	0A0C	-2.484124018D-15
0A0D	7.327471963D-15	0A0E	-4.440892099D-15	0A0F	-3.191891196D-16	0A10	-1.790234627D-15
0B07	4.103024396D-01	0B08	-5.672764189D-16	0B09	3.940462532D-01	0B0A	-4.817136665D-16
0B0B	2.947447418D-01	0B0C	6.110405993D-01	0B0D	-5.874765673D-01	0B0E	5.510594983D-02
0B0F	2.444268303D-01	0B10	2.351548761D-01	0C07	-7.014135130D-01	0C08	-2.152450711D-15
0C09	1.432164299D+00	0C0A	-2.346895557D-15	0C0B	6.221936450-01	0C0C	1.859194531D+00
0C0D	2.568431933D+00	0C0E	8.232393558D-01	0C0F	5.660153359D-01	0C10	6.914194369D-01
0D07	-6.920180892D-01	0D08	5.030076139D-15	0D09	-3.067090959D+00	0D0A	7.091125784D-15
0D0B	-5.575385484D-01	0D0C	-2.586691161D+00	0D0D	7.281485964D+00	0D0E	-4.577719389D+00
0D0F	1.269423206D-01	0D10	-8.692115631D-01	0E07	1.292975614D+00	0E08	-2.719994476D-15
0E09	1.585C6C545D+00	0E0A	-4.629289398D-15	0E0B	1.128591033D-01	0E0C	7.8200779790-01

0E0D	-4.593554333D+00	0E0E	4.186585442D+00	0E0F	-1.027303092D+00	0E10	1.561744597D-02
0F07	6.824510681D 01	0F08	6.049010204D-16	0F09	2.891787293D-01	0F0A	-2.989276999D-16
0F0B	3.993749765D-01	0F0C	4.383082246D-01	0F0D	3.721532571D 02	0F0E	1.116368517D+00
0F0F	1.372459173D+00	0F10	7.8037415890-01	1007	-3.442470009D-01	1008	-1.307089157D-15
1009	6.201791590D 01	100A	1.737964719D 15	100B	-1.903725674D-01	100C	1.065799524D+00
100D	-5.588524881D-01	100E	3.835710172D-01	100F	1.111333421D+00	1010	1.300220397D+00

0002000000000000C000000000000000 76

0101	1.0000000000+00	0201	-1.908195824D-17	0202	1.0000000000+00	0301	2.949029909D 17
0302	4.163336342D-17	0303	1.0000000000+00	0401	4.391018799D 18	0402	9.540979118D-18
0403	5.204170428D-18	0404	1.0000000000+00	0501	1.214306433D-17	0502	0.0
0503	0.0	0504	-8.239936511D-17	0505	1.0000000000+00	0601	1.387778781D-17
0602	0.0	0603	0.0	0604	0.0	0605	1.942890293D-16
0606	1.0000000000+00	0707	1.0000000000+00	0807	0.0	0808	1.0000000000+00
0907	0.0	0908	0.0	0909	1.0000000000+00	0A07	0.0
0A08	-2.775557562D-17	0A09	1.387778781D-17	0A0A	1.0000000000+00	0B07	-7.085261197D-17
0B08	0.0	0B09	3.433871568D-17	0B0A	0.0	0B0B	1.0000000000+00
0C07	1.831217469D-16	0C08	0.0	0C09	3.753033583D-17	0C0A	0.0
0C0B	5.551115123D 17	0C0C	1.0000000000+00	0D07	3.031429274D-16	0D08	0.0
0D09	8.010898802D-17	0D0A	0.0	0D0B	5.551115123D-17	0D0C	3.191891196D 16
0D0D	1.0000000000+00	0F07	1.270142845D-16	0E08	0.0	0E09	1.505414716D-16
0E0A	0.0	0F0B	6.938893904D-17	0E0C	8.326672685D-17	0E0D	1.665334537D 16
0E0E	1.0000000000+00	0F07	-2.541369892D-16	0F08	0.0	0F09	1.175817256D-16
0F0A	0.0	0F0B	4.440892099D 16	0F0C	-4.440892099D-16	0F0D	1.665334537D-16
0F0E	-5.551115123D-17	0F0F	1.0000000000+00	1007	-3.807718030D-16	1008	0.0
1009	7.941780913D-17	100A	0.0	100B	3.774758284D-15	100C	-8.881784197D-16
100D	1.776356839D-15	100E	-6.661338148D-16	100F	-1.872113575D-14	1010	1.0000000000+00

0101	5.0088970180+01	0201	6.2945738280+00	0202	-1.1859160250+01	0301	2.6535755780+00
0302	6.0804735920-01	0303	-9.7788631110+00	0401	3.4677122280-15	0402	-1.4571677200-16
0403	-2.9490299090-17	0404	-5.0060848610+01	0501	-6.3837823920-16	0502	2.2204460490-16
0503	-2.2204460490-16	0504	6.3382066930+00	0505	-1.0242122080+01	0601	-6.6613381480 16
0602	0.0	0603	0.0	0604	4.9057871540+00	0605	1.3853759540+00
0606	-1.3307934600+01	0707	-1.0269774480+01	0807	0.0	0808	-9.6973670550+00
0907	7.0175832730 01	0908	0.0	0909	-1.2922829570+01	0A07	0.0
0A08	6.0227744040-01	0A09	0.0	0A0A	-1.2051058730+01	0B07	8.5320988480 01
0B08	3.7656414080-16	0B09	-1.1869947220-01	0B0A	8.2110494210-17	0B0B	-2.5768242930+00
0C07	1.4836307270+00	0C08	1.6550908490-15	0C09	-6.8479834930-01	0C0A	4.0548715760 16
0C0B	-1.4685138690+00	0C0C	-4.8306424250+00	0D07	1.1925050440+00	0D08	-4.7058726430-15
0D09	2.1455675160+00	0D0A	-1.2051361280-15	0D0B	1.5248115130-01	0D0C	2.0720181990+00
0D0D	-8.8921718610+00	0E07	-2.4178522110+00	0E08	2.9763601370-15	0E09	1.3918132290+00
0E0A	6.9573452780-16	0F0B	3.2224098200-01	0E0C	4.0466468530-01	0E0D	4.8911123090+00
0E0E	-7.5070316130+00	0F07	1.2894068730+00	0F08	1.1991296890 16	0F09	2.9732770770-02
0F0A	2.7781823900-16	0F0B	-3.6897774440-01	0F0C	-7.9896712660-01	0F0D	-5.3646649700-01
0F0E	2.3484405970+00	0F0F	-3.8333786860+00	1007	6.4855790640-01	1008	6.8246234470 16
1009	-2.7051563950-01	100A	6.6621272030-16	100B	-2.5297844220-01	100C	-6.5095223590-01
100D	2.4132455100-01	100E	6.6573171990-01	100F	-1.4202728190+00	1010	2.9867737460+00

INTEGRAL TYPE	NO. COMPUTED	NO. STORED	CPU TIME	TIME / INTEGRAL
COULOMB & HYBRID	2397	2355	61.843 SEC	0.02580 SEC
ONE CENTER	86	86	0.050 SEC	0.00058 SEC
EXCHANGE	4227	0	79.688 SEC	0.01885 SEC
TOTAL TWO ELECTRON	6710	2441	141.581 SEC	0.02110 SEC
ONE ELECTRON	286	286	7.123 SEC	0.02491 SEC
ONE ELECTRON TAILS			0.213 SEC	
TWO ELECTRON TAILS			3.623 SEC	
INTEGRAL RENORMALISATION			2.573 SEC	
VECTOR RENORMALISATION AND TRANSFORMATION			0.640 SEC	

APPENDIX 4

INSTALLATION DETAILS

A. Atomic Integrals

The following is a list of files relevant to the program available on the IBM 370/165 at Daresbury. Filenames refer to userid QCK.

Load library	ALCHEMY.TAILS.LOAD
Load module	LOAD(JKTAILS)
FORTRAN source	ALCHMY.TAILS.FORT

CLIST(OLDFHCL) compiles a routine and places it in the load library, e.g.

```
OLDFHCL MAIN MAIN REG(240K)
```

compiles ALCHMY.TAILS.FORT(MAIN) and places the object module in ALCHEMY.TAILS.LOAD(MAIN).

ALCHMY.CNTL(LINK) links together the load library and places the executable load module in LOAD(JKTAILS)

B. Molecular Integrals

LOAD Module	LOAD(TRANSR)
FORTRAN source	RMATRANS.FORT

ALCHMY.CNTL(TRANSF) compiles this program.

REFERENCES

1. E.P. Wigner, Phys. Rev. 70 (1946) 15; ibid 606; E.P. Wigner and L. Eisenbud, Phys. Rev. 72 (1947) 29.

2. P.G. Burke, A. Hibbert and W.D. Robb, J. Phys. B: Atom. Molec. Phys. 4 (1971) 153.

3. B.I. Schneider, Chem. Phys. Lett. 31 (1975) 237; Phys. Rev. A11 (1975) 1957; B.I. Schneider and P.J. Hay, J. Phys. B: Atom. Molec. Phys. 9 (1976) L165; Phys. Rev. A13 (1976) 2049; B.I. Schneider, M. Le Dourneuf and P.G. Burke, J. Phys. B: Atom. Molec. Phys. 12 (1979), L365; B.I. Schneider, M. Le Dourneuf and Vo Ky Lan, submitted to Phys. Rev. Lett. (1979).

4. P.G. Burke, I. Mackay and I. Shimamura, J. Phys. B: Atom. Molec. Phys. 12 (1977) 2497.

5. B.D. Buckley, P.G. Burke and Vo Ky Lan, Comp. Phys. Commun. 17 (1979) 175.

6. A.D. McLean in: Proc. Conf. on Potential Energy Surfaces in Chemistry ed. W.A. Lester Jr. (IBM Research Laboratory, San Jose, California, 1971), p.87.

7. P.S. Bagus, B. Liu, A.D. McLean and Y.M. Yoshimine, IBM Research Report RJ1129 (1972); IBM Research Report RJ1143 (1973).

8. V.R. Saunders and M.F. Guest. ATMOL3 User Manual (Rutherford Laboratory, 1976).

9. B.J. Ransil, Rev. Mod. Phys. 32 (1960) 245.

TABLE 1

SUBROUTINES USED IN THE EVALUATION OF TAIL INTEGRALS

New Subroutine	Related original subroutine	Comment
TSTUMX	STUMX	A driving routine for the overlap, kinetic energy and nuclear attraction integrals
TSTPQ	STPQ	Overlap and kinetic energy integrals
TTTPQ		Calculates the rest of the kinetic energy integrals needed to form the full square matrix
TTCUPQ	OTCUPQ	Nuclear attraction integrals
TOEINT	WOEINT	Controls output and printing of the one electron integrals
TTEOCW	TEOCW	Driving routine for the coulomb-hybrid two electron integrals
TCHBLK	CHBLKS	Generates indices for the required coulomb-hybrid integrals
TCPQRQ	OCPQRQ	
TWRINT	WRINT	Calculation and storage of the coulomb-hybrid integrals
TCHIN1	Stores the solid harmonic, property integrals ready for the calculation of a block of coulomb-hybrid tail integrals	
TCOF	Evaluates $\int\int Y^*_{\ell_1 m_1} Y_{LM} Y_{\ell_2 m_2} \sin\theta d\theta d\phi$	
TFAC	Evaluates factorials up to 31!	
CLEBGN	Evaluates 3-j coefficients	

TABLE 2

VECTOR MANIPULATION SUBROUTINES

TRNSDR	Reads in target wavefunction data
COMMUN	Reads in total wavefunction data
DIAG SORTQ FMOVER	Jacobi diagonalisation routine
MULT2 MULT1 TRNSLD TRNSQ	Matrix multiplication routines
ENLARG	Orthogonalises molecular orbitals and performs the transformation of the one electron integrals
GETQ	Obtains molecular orbitals from an ATMOL3 dumpfile
ORTHOG	Schmidt orthogonalisation routine
PTCPS	Prints ALCHEMY molecular orbitsl coefficients
STOVC RMEGU STOJ GETVC RPAULC RPAUL WMEGU PRINT FIND WRTCP	ALCHEMY routines for reading and writing vectors from add to files

TABLE 3

ATOMIC INTEGRAL FILES

Unit No.	Comments
FT08F001*	Basis set information
FT08F002	2-electron integrals
FT09F002	1-electron integrals
FT10F001	2-electron tail integrals
FT11F001	1-electron tail integrals
FT12F001+	2-electron integrals, merged, renormalised and relabelled
FT13F001*	1-electron integrals, merged, renormalised and relabelled
FT21F001)*] FT27F001)	Used for storing molecular orbitals

* These files are passed onto the transformation program.

TABLE 4

MOLECULAR INTEGRAL FILES

Unit No.	Comments	Unit No. used by Atomic Integral Package
FT08F001	Atomic orbital basis set data	FT08F001
FT08F002	2-electron atomic integrals	FT12F001
FT09F001	1-electron atomic integrals	FT13F001
FT21F001	Molecular orbital information, only one	FT21F001
FT27F001	of these is required.	FT27F001
FT16F001	Transformed 1- and 2-electron integrals	FT27F001
FT20F001		
FT18F001		
FT19F001	Temporary, intermediate files	
FT28F001		
FT17F001		
FT26F001		

FIGURE CAPTIONS

Fig. 1 The division of configuration space in the R-matrix method.

Fig. 2 The eigenspectrum of the electron-molecule Hamiltonian in all space ($0 \leq$ all $r < \infty$), and in the R-matrix sphere ($0 \leq$ all $r \leq a$).

Fig. 3 Flow chart for the ALCHEMY atomic integrals package.

Fig. 4 Flow chart for the modified ALCHEMY atomic integrals package.

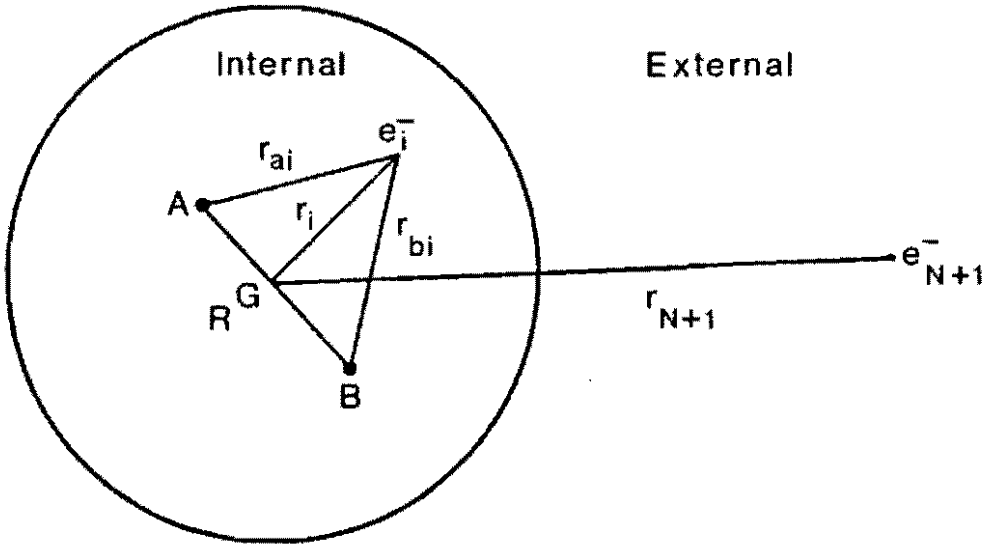
Internal
External
r_{ai}
e_i^-
A
r_i
r_{bi}
e_{N+1}^-
G
R
r_{N+1}
B

Fig.1

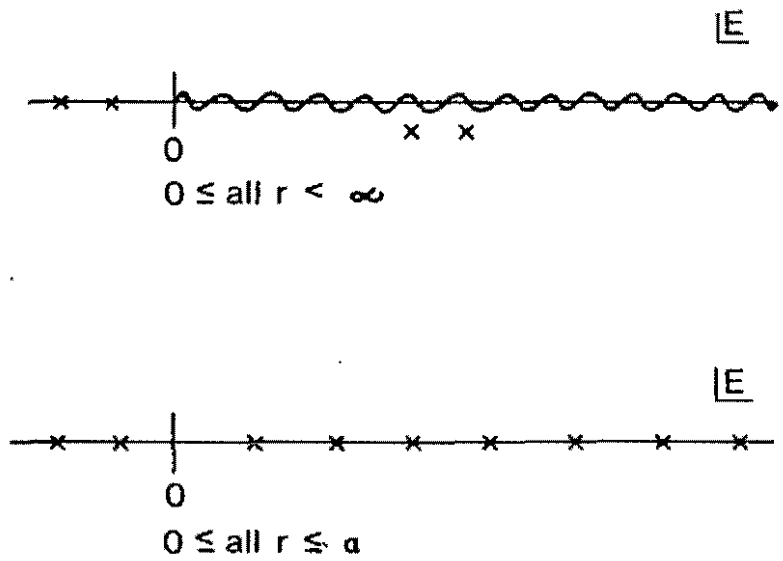
E
0
0 ≤ all r < ∞
E
0
0 ≤ all r ≤ a

Fig.2

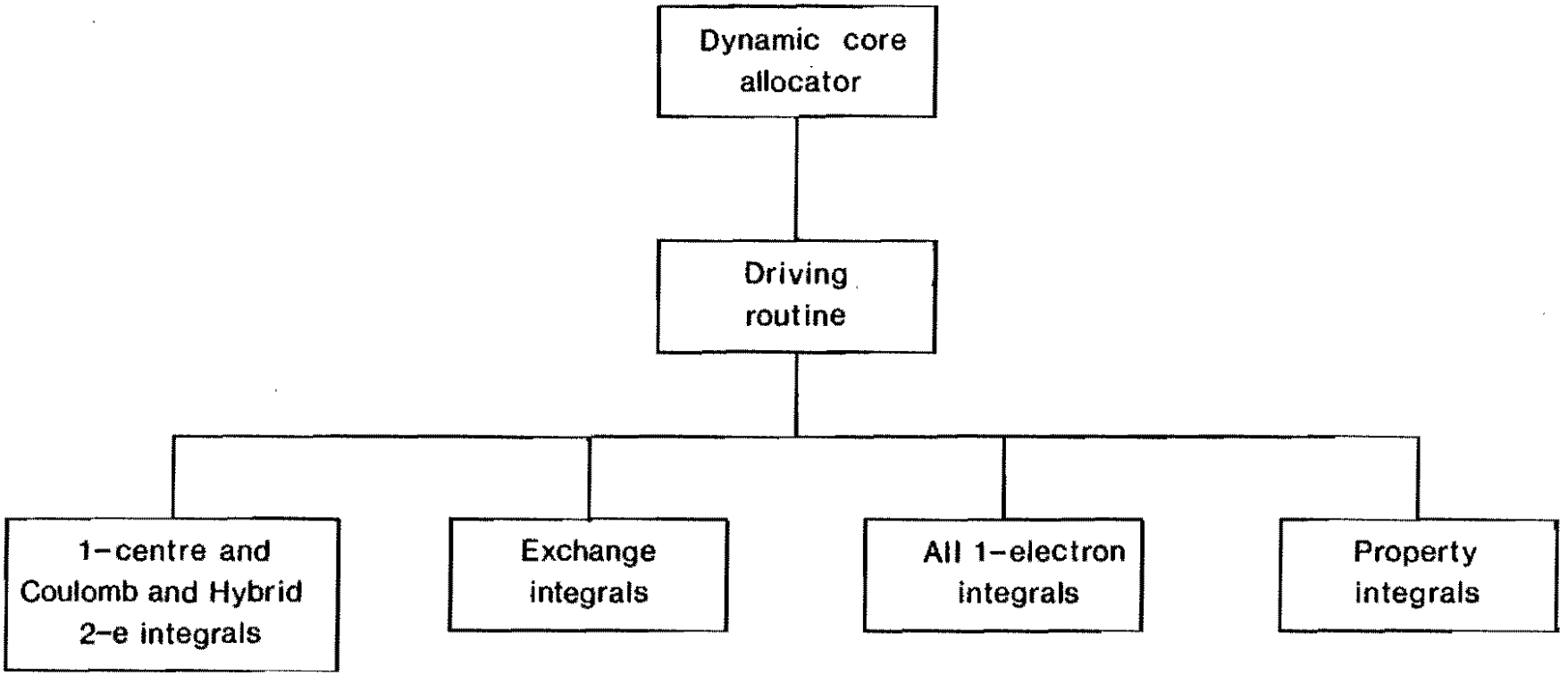
Dynamic core allocator
Driving routine
1-centre and Coulomb and Hybrid 2-e integrals
Exchange integrals
All 1-electron integrals
Property integrals

Fig.3

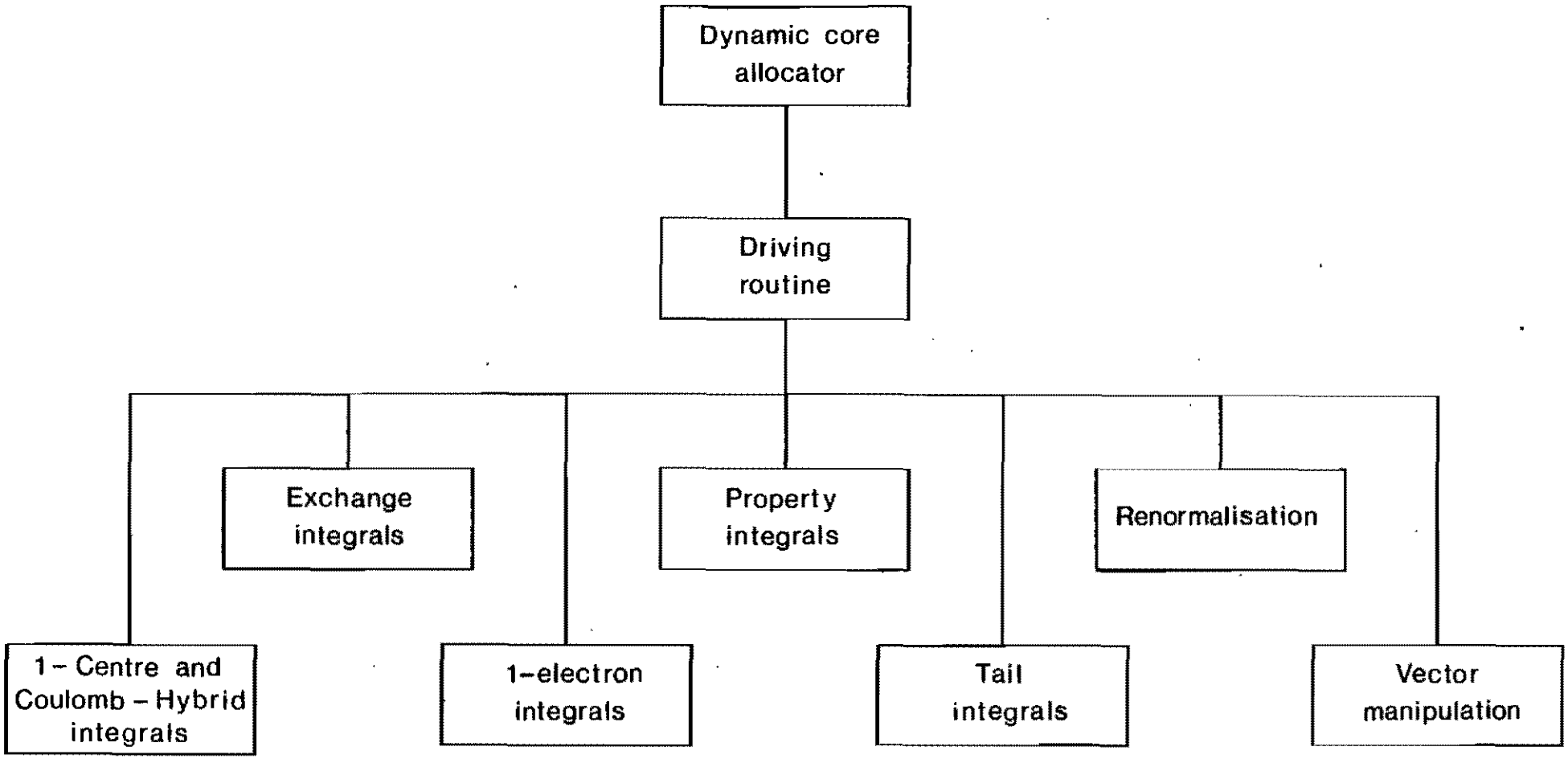
Dynamic core allocator
Driving routine
Exchange integrals
Property integrals
Renormalisation
1 - Centre and Coulomb - Hybrid integrals
1-electron integrals
Tail integrals
Vector manipulation

Fig. 4