October 8, 2008

Discussion about some parts of the CRYSTAL code - Work in progress

Version 1.0.1

Raffaella Demichelis and Roberto Dovesi

Contents

LEGEND

$\mathbf{a}_i$	Direct lattice basis vectors
$\mathbf{b}_i$	Reciprocal lattice basis vectors
l, **g**, **g'**, **h**	Direct lattice vectors
r	Direct lattice point
K	Reciprocal lattice vector
k, **k'**	Reciprocal lattice points
$\mathbf{r}_1$, $\mathbf{r}_2$	Electron coordinates
$\mathbf{R}_A$, $\mathbf{S}_\mu$	Atomic coordinates
$\mathbf{R}_\mu$	Center of the μ^{th} atomic orbital (atomic position)
V_{BZ}	First Brillouin zone volume
GTF	Gaussian Type Functions
AO	Atomic Orbital
MO	Molecular Orbital
BF	Bloch Function
CO	Crystalline orbital
χ_μ, χ_ν..	Atomic functions
σ,ω,μ,ν	Atomic function labels
ψ_i	Monoelectronic function
$\mathrm{Y}\ell m$	Spherical harmonic
α	Gausian exponent
c	Linear combination coefficient
C	Coefficient matrix
γ_λ	Adjoined Gaussian
ρ	Charge distribution
$\rho(\mathbf{r})$	Electronic Density
λ	Shell
n,ℓ,m	Quantum numbers
S	Overlap matrix
X	Exchange term
Z	Nuclear-electron repulsive term
C	Coulomb term
T	kinetik term
F	Fock matrix
H	Hamiltonian (or set of **h** vectors)
V	Potential term
N	Nuclear term or normaization factor
V_{ee}	KS electron-electron contribution
E_{xc}	Exchange and correlation functional
$v(\mathbf{r})$	Potential interacting with electrons
$\mathrm{A}_{\mu\nu}$	General term of the matrix A
$\mathcal{H}$	Hessian matrix
HF	Hartree-Fock
KS	Kohn-Sham
LDA	Local Density Approximation (exchange)
GGA	Generalized Gradient Approximation (exchange)
PBE	GGA; Perdew-Beke-Ernzerhof (exchange)
B3LYP	Beke 3 LYP (exchange-correlation)
LYP	GGA; Lee-Yang-Parr (correlation)
VWN	LSD; Voske-Wilk-Nusair (correlation)
T_i	Thresholds in the integral evaluation
N_{occ}	Number of occupied states
E_F	Fermi energy
E_n	Energy of the n state
s	Shell or atom label
$\hat{\mathrm{R}}$, $\hat{\mathrm{Q}}$	Symmetry operators
O	Lattice vector bringing back a shell (or an atom) in the unit cell after the application of a symmetry operator
FWHM	Full Width Half Maximum.

Chapter 1

Theoretical Framework

1.1 Basic equations

We summarize in the following the basic equations of the variational approach as implemented in CRYSTAL. CRYSTAL is a periodic *ab-initio* code, where the molecular Hartree-Fock (HF) and Density Functional (DFT) methods have been extended to periodic infinite systems. Localized functions are used as a basis set.

1.1.1 Direct and Reciprocal lattices

A lattice is an infinite collection of points repeated at intervals of length a_1, a_2, a_3 along three non-coplanar directions. The three constants are called lattice parameters, and the corresponding a_1, a_2, and a_3 vectors are the basis vectors that define the unit cell[1]. Lattice parameters and angles between the basis vectors are called cell parameters.

A general vector **g** is a lattice vector if the following relationship is verified:

$$\mathbf{g} = n_1\mathbf{a_1} + n_2\mathbf{a_2} + n_3\mathbf{a_3} \tag{1.1}$$

where $\{n_i\}$ are integers.

A general position **r** in the unit cell is usually expressed in terms of fractional coordinates x_i, so that

$$\mathbf{r} = x_1\mathbf{a_1} + x_2\mathbf{a_2} + x_3\mathbf{a_3} \tag{1.2}$$

Consider two points, **r** and **r'**. For translation invariance, if **r'** is such that **r'**=**r**+**g**, they are said to be equivalent[1].

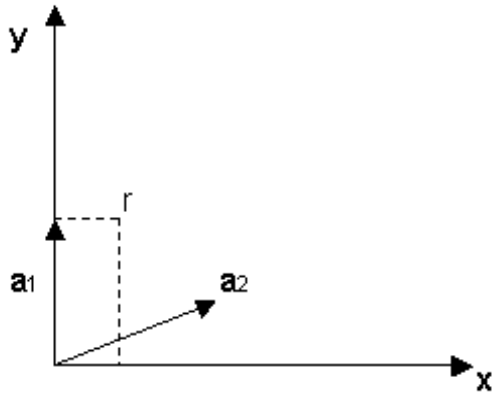

Figure 1.1. Example of the relationship among fractional and Cartesian coordinates. X and Y are the Cartesian axis, a_1=(0, 1) and a_2=($\frac{\sqrt{3}}{2}$,$\frac{1}{2}$) the basis vectors, **r** a point of coordinates, with respect to X and Y, ($\frac{1}{2}$, 1); the angle between a_2 and X is 30°. By substitution of **r**, a_1 and a_2 in eq.1.2 the values of fractional coordinates are obtained, x_1=$\frac{2\sqrt{3}-1}{2}$ and x_2=$\frac{1}{\sqrt{3}}$.

A lattice with the previous properties is called **direct lattice**, defined in the direct space. For any direct lattice a **reciprocal lattice** exists, defined through the following relationship:

$$\mathbf{a}_i \ \mathbf{b}_j = 2\pi\delta_{ij} \tag{1.3}$$

where $\{b_j\}$ are the basis vectors of the reciprocal lattice. As for the direct lattice, any reciprocal lattice vector can be written as a linear combination of the basis vectors with integer coefficients:

$$\mathbf{K} = K_1\mathbf{b_1} + K_2\mathbf{b_2} + K_3\mathbf{b_3} \tag{1.4}$$

and a general position in the reciprocal lattice as a linear combination of the basis vectors with real coefficients:

$$\mathbf{k} = k_1\mathbf{b_1} + k_2\mathbf{b_2} + k_3\mathbf{b_3} \tag{1.5}$$

When

$$|k_i| < 1$$

k belongs to the origin cell. By eq.1.3 **g** and **K** are such that:

$$\mathbf{g} \cdot \mathbf{K} = \ 2\pi n \tag{1.6}$$

Note also:

$$exp(i\mathbf{g}\cdot\mathbf{K}) = 1 \tag{1.7}$$

A particular reciprocal space unit cell is obtained by connecting one reciprocal lattice point to all its nearest neighbors and letting orthogonal planes pass through their midpoints[1]. The volume[2] within these planes is known as the **first Brillouin Zone**(BZ) and its value is given by

$$V_{BZ} = (\mathbf{b}_1 \times \mathbf{b}_2)\cdot\mathbf{b}_3 = (\mathbf{b}_3 \times \mathbf{b}_1)\cdot\mathbf{b}_2 = (\mathbf{b}_3\times\mathbf{b}_2)\cdot\mathbf{b}_1$$

or, if V_{Dir} is the volume of the direct space cell, by referring to eq.1.3,

$$V_{BZ} = \frac{(2\ \pi)^3}{V_{Dir}}$$

1.1.2 Local Functions

A basis set[3, 4, 5] of local Gaussian-type functions (GTFs), called Atomic Orbitals (AOs), is defined in the space of Cartesian coordinates:

$$\gamma_\mu(\mathbf{r}-\mathbf{g}) = N\ Y_{\ell m}(\mathbf{r}-\mathbf{R}_\mu-\mathbf{g})\ exp[-\alpha(\mathbf{r}-\mathbf{R}_\mu-\mathbf{g})^2] \tag{1.8}$$

where μ labels the AOs, R_μ is the centroid of the Gaussian, whose exponent is α, in the reference zero cell, N is a normalization factor and $Y_{\ell m}$ is a real solid spherical harmonic. The variational basis set usually consists of contractions of GTFs:

$$\chi_\mu(\mathbf{r}-\mathbf{g}) = \sum_j^{n_G} d_{\mu,j}\gamma_{\mu j}(\alpha_j;\mathbf{r}-\mathbf{R}_\mu-\mathbf{g}) \tag{1.9}$$

with n_G being the number of GTFs in the contraction and $d_{\mu,j}$ the coefficients of the GTF with the α_j exponent.

1.1.3 From GTFs to Bloch Functions

CRYSTAL evaluates all integrals in the AO basis[1]. For a lattice, made of an infinite number of atoms, this means that the Hamiltonian matrix has infinite size. A new basis set must be defined in order to solve the Shrödinger equation: the Bloch Function (BF) basis set. We will see later on the reason why two basis sets are considered; now, as a first step, we look at the BF properties.

- BFs are expressed as a function of both the position **r** in the direct lattice and the wave vector **k**, and are obtained by the Fourier Transforming (FT) of AOs[3]:

$$\phi_\mu(\mathbf{r};\mathbf{k}) \;=\; \frac{1}{\sqrt{N}} \sum_{\mathbf{g}} \; exp(i\mathbf{kg}) \; \chi_\mu^{\mathbf{g}}(\mathbf{r}-\mathbf{R}_\mu) \tag{1.10}$$

  where **g** is a direct lattice translation vector, as shown in eq.1.1, and $\chi_\mu^{\mathbf{g}}$ is the μ AO in the **g** cell.

- BFs verify Bloch's theorem, that states[1]

$$\phi_n(\mathbf{r}+\mathbf{g};\mathbf{k}) \;=\; exp(i\mathbf{kg}) \; \phi_n(\mathbf{r};\mathbf{k}) \tag{1.11}$$

  and provides a relationship among the values of an eigenfunction at equivalent points in the lattice, that is the starting function multiplied by a phase factor. Note that, for **k**=**K** (see eq.1.7), the phase factor value is 1. Consider, for example, a new point **k'**=**k**+**K**, with **K** being a reciprocal lattice vector, as shown in eq.1.4. From Bloch theorem we know that

$$\begin{aligned} \phi_n(\mathbf{r}+\mathbf{g};\mathbf{k}') = exp(i\mathbf{k}'\mathbf{g}) \; \phi_n(\mathbf{r};\mathbf{k}') = exp(i(\mathbf{k}+\mathbf{K})\mathbf{g}) \; \phi_n(\mathbf{r};\mathbf{k}') = \\ = exp(i\mathbf{kg}) \; exp(i\mathbf{Kg}) \; \phi_n(\mathbf{r};\mathbf{k}') = exp(i\mathbf{kg}) \; \phi_n(\mathbf{r};\mathbf{k}') \end{aligned} \tag{1.12}$$

  and we obtain

$$\phi_n(\mathbf{r};\mathbf{k}) = \phi_n(\mathbf{r};\mathbf{k}')$$

  that is a consequence of translation invariance: BFs are periodic functions with respect to **k**, and have the same value when equivalent **k** points are considered. This property is useful to solve the problem of an infinite number of cells in periodic systems, because it allows to restrict the analysis of **k** points to the origin cell, that corresponds to the first BZ. **k** can be considered a continuous variable; the problem of the infinite numberof **k** points in the first BZ is solved by sampling the Hamiltonian matrix at a finite number of **k** points (by choosing a k-grid in the first BZ) solving the Schrödinger equation at these points and interpolating eigenvalues and eigenvectors.

- BFs are eigenfunctions of translation operators and of all operators commuting with them, for example the Hamiltonian operator[1]. This is

because of the translation invariance of periodic systems: the potential energy of a crystal does not change for a translation by a direct lattice vector $\mathbf{g}$.

$$V(\mathbf{r}-\mathbf{g}) = V(\mathbf{r}) \tag{1.13}$$

Then, BFs are basis for the irreducible representations (IR) of the group of the lattice translations and are orthonormal when belonging to different $\mathbf{k}$ points:

$$< \ \phi_\mu(\mathbf{k}) \mid H \mid \phi_\nu(\mathbf{k}') \ > = \ h_{\mu\nu} \ \delta_{\mathbf{k}\mathbf{k}'} \tag{1.14}$$

BFs for the same $\mathbf{k}$ points are chosen so as to be orthonormal:

$$< \ \phi_\mu(\mathbf{k}) \mid \phi_\nu(\mathbf{k}) \ > = \ \delta_{\mu\nu} \tag{1.15}$$

1.1.4 The Crystalline Orbitals

Like Molecular Orbitals (MO) are linear combinations of AOs, Crystalline Orbitals (CO)[5, 1] are linear combinations of BFs:

$$\psi_n(\mathbf{r};\mathbf{k}) \ = \ \sum_j c_{jn}(\mathbf{k}) \ \phi_j(\mathbf{r};\mathbf{k}) \tag{1.16}$$

and are the solution of the Schrödinger equation

$$\widehat{H} \ \psi_n(\mathbf{r};\mathbf{k}) \ = \ E_n(\mathbf{k}) \ \psi_n(\mathbf{r};\mathbf{k}) \tag{1.17}$$

In the BF basis, eq.1.17 takes the form of a matrix equation and has to be solved for each $\mathbf{k}$ point of the reciprocal space:

$$\begin{aligned} \mathbf{H}(\mathbf{k}) \ \mathbf{C}(\mathbf{k}) \ &= \ \mathbf{S}(\mathbf{k}) \ \mathbf{C}(\mathbf{k}) \ \mathbf{E}(\mathbf{k}) \\ \mathbf{C}(\mathbf{k}) \ \mathbf{S}(\mathbf{k}) \ \mathbf{C}^\dagger(\mathbf{k}) \ &= \ \mathbf{I} \end{aligned} \tag{1.18}$$

where all matrices have equal size n_f, $\mathbf{S}(\mathbf{k})$ is the overlap matrix, and $\mathbf{C}(\mathbf{k})$ the eigenvector matrix.

The infinite Hamiltonian matrix in the AOs basis, becomes an infinite block-diagonal matrix in the BF basis set, with each block referring to one particular $\mathbf{k}$ point; the infinite number of $\mathrm{n}_f \times \mathrm{n}_f$ blocks, where n_f is the number of BFs (as many BFs as AOs), are independent from each other and can be treated separately[1].

Why, at this point, using the AOs basis set? In the BF basis representation, solving the Schrödinger equation implies the solution of a matrix equation for

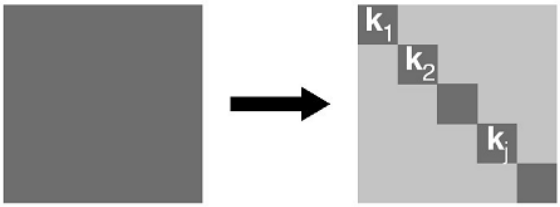

Figure 1.2. Transformation of the H matrix from AOs basis set (left) to BF basis set (right). The problem of an infinite matrix becomes an infinite number of problems of finite size.

every **k**; therefore, all matrices are defined in the reciprocal space. However, because BFs are expressed in terms of a local basis set, it is convenient to calculate all matrices first in the AO basis and then transform to BF basis. That is because we want to exploit the "chemical" properties of localized functions, by which neighborhood relationships are established. In addition, we would like to exploit the reciprocal space orthogonality by using the BFs, and that is why, after the evaluation of the various integrals in direct space, all the quantities are transformed to reciprocal space, where the Schrödinger equation is solved for a certain number of **k** points.

1.1.5 The Self Consistent Field

We summarize, in this section, all steps of a periodic calculation as performed by CRYSTAL. We will see into details later on how each matrix element is calculated. Inter-electron repulsion terms in the hamiltonian can only be calculated if CO (the solution of the Schrödinger equation) are known, so an iterative method, called Self Consistent Field (SCF)[1], is used to find the linear combination coefficients. The scheme shown in fig.(1.3) is almost the same as for the molecular case, but in a periodic system the Roothan-Hall equation has the form of eq.1.18 and it has to be solved at any **k**-point of a chosen **k**-grid.

At first, a **k**-grid is defined. As an iterative method is used, an initial guess has to be defined; the SCF cycle can start either with a density matrix guess or with a Fock matrix guess. Generally, SCF method starts with a density matrix guess. The first P matrix can be either a block-diagonal matrix, obtained as

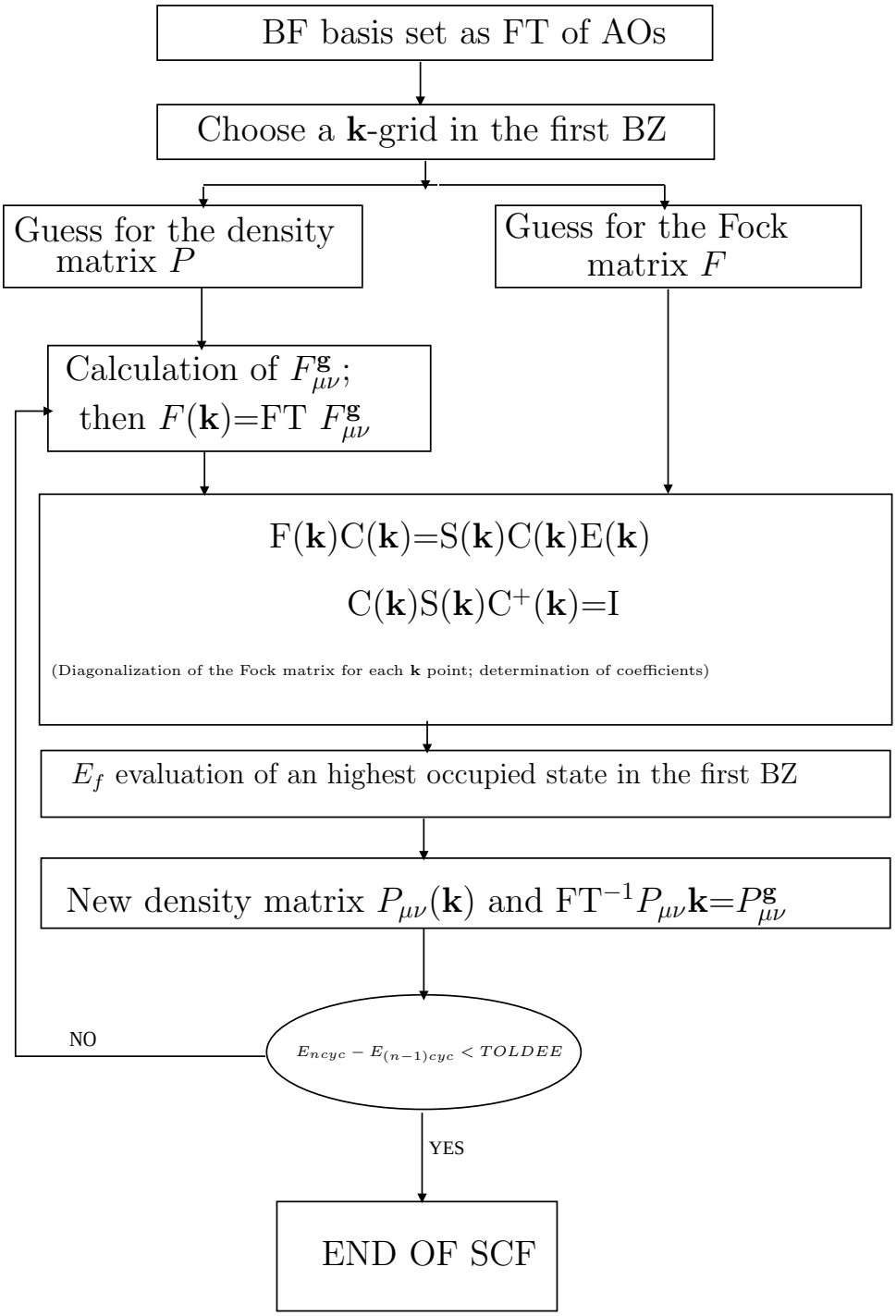

Figure 1.3. The SCF-cycle Self scheme for an infinite periodic system. FT is the Fourier Trensform, $TOLDEE$ the threshold on energy.

a sum of atomic densities with a block for each atom, or a P matrix obtained in a previous similar system calculation. If available, it is also possible a guess for the Fock matrix from a previous calculation for a similar system, instead of the P guess: in this case the first Fock matrix calculation is skipped, as shown in fig.(1.3). After that, the SCF-cycle starts, and the Fock matrix in direct space is constructed and Fourier transformed to the reciprocal space, as shown by equations from (1.24) to (1.31), in the next section. Remember that a Fock matrix for every **k**-point exists. The next step is solving the Roothan-Hall equation for each **k**-point: the Fock matrix is diagonalized and the coefficients of the linear combinations matrix $\mathbf{C}(\mathbf{k})$ are determined. The Fermi level E_F is then calculated as the energy of an highest occupied state, and so there are all elements to obtain the new density matrix, as in eq.1.21 and eq.1.22. The convergence criteria for the SCF-cycle can work on the P matrix, on the Fock matrix, or on energy value. In CRYSTAL the SCF-cycle end when the new energy is the same as the one obtained in the previous cycle (the difference $\mid E_n - E_{n-1} \mid$ must be less than a given threshold $TOLDEE$; 10^{-5} Ha is the default value for a single point of energy). The total energy per cell can be calculated after the SCF convergence as shown in eq.1.34

1.1.6 Building matrices

Let's have a look, now, at how the matrices in eq.1.18 are obtained[3, 6, 1, 4, 5]. In CRYSTAL the various matrix elements for a periodic system are calculated in direct space (AO basis) and then Fourier transformed to reciprocal space. This step is repeated M_k times, where M_k is the number of **k** points selected in the first BZ. So, a preliminary explanation about how matrices in direct and reciprocal space are related is necessary. We will consider, as an example, the overlap matrix $S_{\mu\nu}(\mathbf{k})$.

$$S_{\mu\nu}(\mathbf{k}) =< \ \phi_\mu(\mathbf{k}) \mid \phi_\nu(\mathbf{k}) \ >= \int_D \phi^*_\mu(\mathbf{r};\mathbf{k}) \ \phi_\nu(\mathbf{r};\mathbf{k}) d\mathbf{r} = \tag{1.19}$$

$$= \frac{1}{N} \sum_{\mathbf{g}} \sum_{\mathbf{g}'} exp(-i\mathbf{kg}) \ exp(i\mathbf{kg}') \int_D \chi^*_\mu(\mathbf{r}-\mathbf{g}) \ \chi_\nu(\mathbf{r}-\mathbf{g}') d\mathbf{r}$$

Consider now **r-g=r'** and **g-g'= -l**

$$S_{\mu\nu}(\mathbf{k}) = \frac{1}{N} \sum_{\mathbf{g}} \sum_{\mathbf{g'}} exp(-i\mathbf{kg})\ exp(i\mathbf{kg'}) \int_D \chi^*_\mu(\mathbf{r'})\ \chi_\nu(\mathbf{r'} + \mathbf{g} - \mathbf{g'}) d\mathbf{r'} =$$
$$= \frac{1}{N} \sum_{\mathbf{g}} \sum_{\mathbf{l}} exp(-i\mathbf{kg})\ exp[i\mathbf{k}(\mathbf{g} + \mathbf{l})] \int_D \chi^*_\mu(\mathbf{r'})\ \chi_\nu(\mathbf{r'} - \mathbf{l}) d\mathbf{r'} =$$
$$= \frac{1}{N} \sum_{\mathbf{g}} exp[-i(\mathbf{k} - \mathbf{k})\mathbf{g}] \sum_{\mathbf{l}} exp(i\mathbf{kl}) \int_D \chi^*_\mu(\mathbf{r'})\ \chi_\nu(\mathbf{r'} - \mathbf{l}) d\mathbf{r'} =$$
$$= \sum_{\mathbf{l}} exp(i\mathbf{kl}) < \chi_\mu \mid \chi^{\mathbf{l}}_\nu > = \sum_{\mathbf{l}} exp(i\mathbf{kl}) S^l_{\mu\nu}$$

where

k is one of the points of the first BZ at which the Fock matrix F is diagonalized at each SCF cycle (see below) defined in eq.1.5

g, g', l are direct lattice translation vectors, as shown in eq.1.1

μ is an AO centered in the origin cell

ν is an AO in the **g** cell

At the end of this section, a few words have to be spent for another important property, related to translational invariance: any integral, and so any matrix element, is invariant to translation For example

$$< \chi^{\mathbf{l}}_\mu \mid \chi^{\mathbf{g}}_\nu > = < \chi^0_\mu \mid \chi^{\mathbf{g-l}}_\nu > = < \chi^0_\mu \mid \chi^{\mathbf{g'}}_\nu > \tag{1.20}$$

where the linear combination between two lattice vectors, **g-l=g'**, is again a lattice vector.

In the following, we are going to describe all matrix elements involved in a periodic calculation to build the Hamiltonian matrix and solve the Schrödinger equation.

1.1.7 The Density matrix

The density matrix $\mathbf{P}(\mathbf{k})$ is a $n_f \times n_f$ matrix, defined as

$$P_{\mu\nu}(\mathbf{k}) = \sum_{n}^{Nocc.} c^*_{\mu n}(\mathbf{k})\ c_{\nu n}(\mathbf{k}) \tag{1.21}$$

where n runs over the states' index and $c_{jn}(\mathbf{k})$ are the linear combination coefficients in eq.1.16; the sum over n is extended to all occupied states N_{occ}, the number of which depends on $\mathbf{k}$. The way to go from direct to reciprocal space and from reciprocal to direct space, as shown for $S_{\mu\nu}(\mathbf{k})$, is the Fourier Transform. So, one can get the direct space density matrix by eq.1.22[1]:

$$P^{\mathbf{g}}_{\mu\nu} = \sum_{\mathbf{k}} \sum_{n}^{Nocc} exp(i\mathbf{kg})\ c^{*}_{\mu n}(\mathbf{k})\ c_{\nu n}(\mathbf{k})\ = \tag{1.22}$$

$$= \sum_{\mathbf{k}} \sum_{n} exp(i\mathbf{kg})\ c^{*}_{\mu n}(\mathbf{k})\ c_{\nu n}(\mathbf{k})\ \theta(E_F - E_n(\mathbf{k})) =$$

$$= \sum_{n} \frac{1}{V_{BZ}} \int_{BZ} exp(i\mathbf{kg})\ c^{*}_{\mu n}(\mathbf{k})\ c_{\nu n}(\mathbf{k})\ \theta(E_F - E_n(\mathbf{k}))\ d\mathbf{k}$$

where the sum over n can be considered for all states, because θ, the Heaviside step function, permits to exclude virtual states; E_F is the Fermi energy (the energy value of an highest occupied state) and V_{BZ} is the first Brillouin Zone volume. The $\mathbf{k}$-sum has been considered as an integral in the first BZ because $\mathbf{k}$ is a continue variable.

1.1.8 The Fock matrix

Let us consider, at this point, the Fock matrix in direct space. The Fock operator is defined as a sum of four contributions

$$\widehat{F} = \widehat{T} + \widehat{Z} + \widehat{C} + \widehat{X} \tag{1.23}$$

where $\widehat{T}$ and $\widehat{Z}$ are the kinetic and nuclear operators, $\widehat{C}$ is the bielectronic Coulomb operator and $\widehat{X}$ is the HF exchange term. For a general wave function ψ, corresponding to a CO, the eigenvalue of Fock operator gives the energy of that MO or CO. The general single matrix element is defined as in eq.1.24 and transformed in reciprocal space by eq.1.25 for every $\mathbf{k}$ point.

$$F^{\mathbf{g}}_{\mu\nu} = < \chi_\mu \mid F \mid \chi^{\mathbf{g}}_{\nu} > \tag{1.24}$$

$$F_{\mu\nu}(\mathbf{k}) = \sum_{\mathbf{g}}^{\infty} e^{i\mathbf{kg}} F^{\mathbf{g}}_{\mu\nu} \tag{1.25}$$

In $F_{\mu\nu}(\mathbf{k})$ μ and ν run over the AOs basis, from 1 to N (number of AOs of the unit cell), then $F_{\mu\nu}(\mathbf{k})$ is a square finite matrix.

From eq.1.23:

$$F_{\mu\nu}^{\mathbf{g}} = T_{\mu\nu}^{\mathbf{g}} + Z_{\mu\nu}^{\mathbf{g}} + C_{\mu\nu}^{\mathbf{g}} + X_{\mu\nu}^{\mathbf{g}} \tag{1.26}$$

Terms of eq.1.26 are defined as follows:

${T_{\mu\nu}}^{\mathbf{g}}$ is the kinetic integral[4]:

$$T_{\mu\nu}^{\mathbf{g}} = -\frac{1}{2} < \chi_\mu^0 \mid \nabla^2 \mid \chi_\nu^{\mathbf{g}} >= -\frac{1}{2}\int_D \chi_\mu(\mathbf{r})\nabla^2\chi_\nu(\mathbf{r}-\mathbf{g})dr \tag{1.27}$$

${Z_{\mu\nu}}^{\mathbf{g}}$ is the monoelectronic nuclear interaction integral[4]:

$$Z_{\mu\nu}^{\mathbf{g}} = \sum_{\mathbf{h}}\sum_{A}^{N_A} < \chi_\mu^0 \mid \frac{1}{|\mathbf{r}-\mathbf{R_A}-\mathbf{h}|} \mid \chi_\nu^{\mathbf{g}} >= \tag{1.28}$$

$$= \sum_{\mathbf{h}}\sum_{A}^{N_A} \int_D \chi_\mu(\mathbf{r})\ \frac{1}{|\mathbf{r}-\mathbf{R_A}-\mathbf{h}|}\ \chi_\nu(\mathbf{r}-\mathbf{g})d\mathbf{r}$$

where A is the atom in position R_A in the unit cell and $\sum_{\mathbf{h}}$ runs over all translation lattice vector.

${C_{\mu\nu}}^{\mathbf{g}}$ is the Coulomb bielectronic term[4]:

$$C_{\mu\nu}^{\mathbf{g}} = \sum_{\sigma,\omega}\sum_{\mathbf{g'}} P_{\sigma\omega}^{\mathbf{g'}} \sum_{\mathbf{h}} < \chi_\mu^0\ \chi_\nu^{\mathbf{g}} \mid \frac{1}{|\mathbf{r_1}-\mathbf{r_2}|} \mid \chi_\sigma^{\mathbf{h}}\ \chi_\omega^{\mathbf{h+g'}} > \tag{1.29}$$

$$< \chi_\mu^0\ \chi_\nu^{\mathbf{g}} \mid \frac{1}{|\mathbf{r_1}-\mathbf{r_2}|} \mid \chi_\sigma^{\mathbf{h}}\ \chi_\omega^{\mathbf{h+g'}} >= \tag{1.30}$$

$$= \int\int_D \chi_\mu(\mathbf{r_1})\chi_\nu(\mathbf{r_1}-\mathbf{g})\frac{1}{|\mathbf{r_1}-\mathbf{r_2}|}\chi_\sigma(\mathbf{r_2}-\mathbf{h})\chi_\omega(\mathbf{r_2}-\mathbf{h}-\mathbf{g'})d\mathbf{r_1}d\mathbf{r_2}$$

where ${P_{\sigma\omega}}^{\mathbf{g'}}$ is the density matrix (see eq.1.21 and eq.1.22), r_i refers to the i^{th} electron, **g'** and **h** are two translation lattice vectors. The three sums mean that all χ functions in all the infinite cells have to be considered for the second electron.

For convergence reasons, the different Coulomb terms need to be deeply reorganized; this point is, however, irrelevant to the present discussion.

${X_{\mu\nu}}^{\mathbf{g}}$ is the HF exchange term[4]:

$$X_{\mu\nu}^{\mathbf{g}} = -\frac{1}{2}\sum_{\sigma,\omega}\sum_{\mathbf{g'}} P_{\sigma\omega}^{\mathbf{g'}} \sum_{\mathbf{h}} < \chi_\mu^0\ \chi_\sigma^{\mathbf{h}} \mid \frac{1}{|\mathbf{r_1}-\mathbf{r_2}|} \mid \chi_\nu^{\mathbf{g}}\ \chi_\omega^{\mathbf{h+g'}} >= \tag{1.31}$$

$$= -\frac{1}{2}\sum_{\sigma,\omega}\sum_{\mathbf{g'}} P_{\sigma\omega}^{\mathbf{g'}} \sum_{\mathbf{h}} \int\int_D \chi_\mu(\mathbf{r_1})\ \chi_\sigma(\mathbf{r_1}-\mathbf{h})\ \frac{1}{|\mathbf{r_1}-\mathbf{r_2}|}\cdot$$

$$\cdot\chi_\nu(\mathbf{r_2}-\mathbf{g})\ \chi_\omega(\mathbf{r_2}-\mathbf{h}-\mathbf{g'})d\mathbf{r_1}d\mathbf{r_2}$$

The HF electrostatic Hamiltonian operator is defined as

$$\widehat{H} = \frac{1}{2}(\widehat{F} + \widehat{Z} + \widehat{T}) \tag{1.32}$$

and the matrix elements can be written as

$$H^{\mathbf{g}}_{\mu\nu} = \frac{1}{2}(F^{\mathbf{g}}_{\mu\nu} + Z^{\mathbf{g}}_{\mu\nu} + T^{\mathbf{g}}_{\mu\nu}) \tag{1.33}$$

Finally, the total energy per cell is given by

$$\begin{aligned} E_{tot} = \; & N + \sum_{\mu,\nu,\mathbf{g}} P^{\mathbf{g}}_{\mu\nu} \; H^{\mathbf{g}}_{\mu\nu} = \\ = \; & N + \frac{1}{2} \sum_{\mu,\nu,\mathbf{g}} P^{\mathbf{g}}_{\mu\nu} \; (F^{\mathbf{g}}_{\mu\nu} + Z^{\mathbf{g}}_{\mu\nu} + T^{\mathbf{g}}_{\mu\nu}) \end{aligned} \tag{1.34}$$

where N is the nuclear potential term. For the other terms see eq.1.26, 1.27, 1.28, and 1.22.

At this point, a question springs out: how is the $F^{\mathbf{g}}$ matrix stored? (see appendix 2, *GMFCAL* and *MONMAD* subroutines)

In a naïve scheme, one could limit the $\mathbf{g}$ sum to a finite set ($\mathbf{g} < G_m$), and ${F_{\mu\nu}}^{\mathbf{g}}$ could be organized as a set of G_m $n_f \times n_f$ matrices; this organization is actually very inefficient, because many of the matrix elements would be null, or very small, and one would like to disregard them. In the next sections and in appendix 2 we will try to answer the following questions:

A How are the infinite sums treated?

B How are the various matrices stored?

C How is the point symmetry exploited?

1.1.9 The Density Functional Theory

One of the most central problems of theoretical solid state chemistry consists in developing suitable methods for solving the many-electron Shrödinger equation. The HF method provides an approximate solution to the N-particle problem, bit it is not the best suited option for extended systems. Today, the most commonly employed scheme is the Density Functional method.

The Density Functional Theory (DFT)[7], formalised by Hohenberg and Kohn (HK) [8] and developed by Kohn and Sham (KS) [9], is based on the

functional hamiltonian dependence of a system, in its fundamental state, on the electronic density $\rho(\mathbf{r})$.

A wave function for an N-electron system contains $3N$ coordinates, three for each electron. The electron density $\rho(\mathbf{r})$ is the square of the wave function, integrated over *N-1* electron coordinates: $\rho(\mathbf{r})$ only depends on three coordinates, *independently* of the number of electrons. This means that, while the complexity of a wave function increases with the number of electrons, the electron density has the same number of variables, independently of the system size.

DFT is based on two theorems[8]. The first HK theorem states all the electronic properties of a system in its ground state are determined by the fundamental state electronic density.

The second HK theorem relates, by the variational theorem, the fundamental state electronic density $\rho_0(\mathbf{r})$ of a system to its total energy:

$$E[\rho(\mathbf{r})] \geq E[\rho_0(\mathbf{r})] \equiv E \tag{1.35}$$

The total energy is:

$$E[\rho(\mathbf{r})] = T[\rho(\mathbf{r})] + V_{ee}[\rho(\mathbf{r})] + \int_D \rho(\mathbf{r}) v(\mathbf{r}) d\mathbf{r} \tag{1.36}$$

where T is the electronic kinetic energy (see eq.(1.27) for HF method), V_{ee} is the electron-electron contribution and $v(\mathbf{r})$ is a potential interacting with electrons, including the nuclear-electronic interaction and any external potential acting on the system. As T and V_{ee} expressions are exactly known, the fundamental state can be determined by the direct minimization of the functional in eq.(1.36)[10]. Although non-omogeneous electronic distribution for molecules and solids cannot be exactly explicited, approximations have to be introduced on this term.

The KS method consists in taking as a reference a non-interacting electron system, for which the kinetic energy can be calculated. Electron-electron interactions, then, can be calculated by the classic medium-field approximation, whereas all non-classic contributions (correlations) in V_{ee} and the contribution of non-interacting electrons to kinetic energy are incorporated in a further therm, namely correlation and exchange energy.

$$E_{xc}[\rho(\mathbf{r})] = V_{ee} - \frac{1}{2} \int_D \frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r} d\mathbf{r}' \tag{1.37}$$

Note that the second integral in eq.(1.37) is the Coulomb integral C, and it has the same forulation as for the HF method (see eq.(1.29)).

$\rho(\mathbf{r})$, then, can be expressed in terms of monoelectronic functions ψ_i:

$$\rho(\mathbf{r}) = \sum_{i=1}^{N} |\psi_i|^2 \tag{1.38}$$

where the N ψ_i are the solutionstions of the N monoelectronic KS equations.

$$\begin{aligned} \widehat{H}\psi_i = \epsilon_i\psi_i \qquad i = 1,...,N \\ \widehat{H} = \widehat{T} + \widehat{Z} + \widehat{C} + \widehat{E}_{xc} \end{aligned} \tag{1.39}$$

Compare this eq.(1.39) to the Fock operator in eq.(1.23): the only difference with respect to the HF method is that in DFT the correlation energy is taken into account. The E_{xc} functional is defined as follows:

$$E_{xc} = \frac{\partial E_{xc}[\rho(\mathbf{r})]}{\partial \rho(\mathbf{r})} \tag{1.40}$$

If the exchange and correlation term is known, KS equations provide exact solutions. In order to apply the DFT method, however, approximated formulations of E_{xc} have to be defined: the difference among DFT methods is the choice of the functional form of this term. The simplest solution is the "local density approximation" (LDA)[10]:

$$E_{xc}^{LDA}[\rho(\mathbf{r})] = \int_D \epsilon_{xc}(\rho(\mathbf{r}))\rho(\mathbf{r})d\mathbf{r} \tag{1.41}$$

where $\epsilon_{xc}(\rho(\mathbf{r}))$ is the exchange-correlation energy for a particle of an uniform electron-interacting gas with density $\rho(\mathbf{r})$.

A more accurate approximation level is the "generalized gradient approximation" (GGA)[10]:

$$E_{xc}^{GGA}[\rho(\mathbf{r})] = \int_D f(\rho(\mathbf{r}),|\nabla\rho(\mathbf{r})d\mathbf{r} \tag{1.42}$$

where $f(\rho(\mathbf{r}),|\nabla\rho(\mathbf{r}))$ is a function of both the electronic density and its gradient. GGA permits a better description of non-uniform systems. From GGA approximation the "gradient-corrected" methods are derived.

The most used functionals, as they usually provide larger accuracy in geometry optimization, energy and frequancy calculations, are combinations of

functionals, and include a percentage of exact HF exchange[11]. They are known as Hybrid methods.

As an example, we consider the B3LYP method, that has been used to perform al the calculations in described in PART II of this thesis.

$$E_{xc}^{B3LYP} = [A \cdot E_x^{Slater} + (1-A) \cdot E_x^{HF} + B \cdot E_x^{Beke}] + [C \cdot E_c^{LYP} + (1-C) \cdot E_c^{VWN}] \tag{1.43}$$

where the exchange term is defined by three different functionals mixed by A and B parameters, and the correlation term is defined by two E_c functionals depending on the C parameter. A, B and C parameters are determined empirically, by best fit on a large set of molecular properties.

1.1.10 The overlap criteria. The pseudo overlap

In this section the problem of the truncation of the infinite sums met in the previous equations will be discussed into detail[5, 4]. AOs belonging to a given atom are grouped into shells. A shell λ is a group of AOs characterized by the same quantum numbers n and l (for example the 3d-shell) or by the same main quantum numbers n (for example an sp-shell); in CRYSTAL s, p, sp, d and f shells can be used. A single, normalized s-type GTF called **adjoined Gaussian** (γ_λ, **EXAD** vector in the *BASATO* module) is associated with each shell (λ) and the α_λ exponent of γ_λ is the smallest among all exponents of the Gaussian functions in the contraction.

The various integrals are evaluated in bunches (see appendix 2, *SELEZ* subroutine) and treated on the basis of overlap criteria: the adjoined Gaussians are used to estimate the overlap between AOs. For example the 25 kinetic or overlap integrals of a couple of d-d shells are produced by a single call to the subroutine computing them. The single s-s overlap is evaluated for selection, and the full set of 25 d-d integrals is then calculated or skipped, according to the value of the overlap.

Consider a simple integral such as $T_{\mu\nu}^{\mathbf{g}}$ or $S_{\mu\nu}^{\mathbf{g}}$. It is easily shown that it goes exponentially to zero when the distance D=$|R_\mu$-$(R_\nu+\mathbf{g})|$, where R_μ is the position of the atom on which γ_μ is centered, increases. Then, thresholds T_i are defined, such to limit to a finite set the infinite set of $|R_\mu$-$(R_\nu+\mathbf{g})|$: when the overlap between the two involved functions is smaller than 10^{-T_i} the integral is disregarded. For reason that we will not discuss here, five different T_i values are used in CRYSTAL, four of which influence the size of the matrices: T_1, T_3,

T_4 and T_5, while T_2 influence only the cost of the calculation.

Let us have a look, now, to the meaning of the five threshold values. In order to simplify the analysis, the equations described in the previous section are summarized in tab.(1.1).

The total energy per cell is:

$$E_{tot} = N + \frac{1}{2}\sum_{\mu,\nu}\sum_{\mathbf{g}} P^{\mathbf{g}}_{\mu\nu}\,(F^{\mathbf{g}}_{\mu\nu} + Z^{\mathbf{g}}_{\mu\nu} + T^{\mathbf{g}}_{\mu\nu}) =$$

$$= N + \frac{1}{2}\sum_{\mu,\nu}\sum_{\mathbf{g}} P^{\mathbf{g}}_{\mu\nu}\,(Z^{\mathbf{g}}_{\mu\nu} + T^{\mathbf{g}}_{\mu\nu} + C^{\mathbf{g}}_{\mu\nu} + X^{\mathbf{g}}_{\mu\nu} + Z^{\mathbf{g}}_{\mu\nu} + T^{\mathbf{g}}_{\mu\nu}) =$$

$$= N + \frac{1}{2}\sum_{\mu,\nu}\sum_{\mathbf{g}} P^{\mathbf{g}}_{\mu\nu}\,(2Z^{\mathbf{g}}_{\mu\nu} + 2T^{\mathbf{g}}_{\mu\nu} + C^{\mathbf{g}}_{\mu\nu} + X^{\mathbf{g}}_{\mu\nu})$$

We can rewrite the total energy as a sum of five contributions:

$$E_{tot} = E_{Nuclear} + \underbrace{E_{Kinetic} + E_{Nucl-Electron} + E_{Coulomb} + E_{Exchange}}_{E_{Electronic}}$$

Every contribution to electronic energy, according to both the previous equations, can be written as follows:

$$E_{Kinetic} = -\frac{1}{2}\sum_{\mu\nu}\sum_{\mathbf{g}} P^{\mathbf{g}}_{\mu\nu}\overbrace{<\ \chi^0_\mu \mid \nabla^2 \mid \chi^{\mathbf{g}}_\nu>}^{T_1}$$

$$E_{Nucl-Electron} = \sum_{\mu\nu}\sum_{\mathbf{g}} P^{\mathbf{g}}_{\mu\nu}\sum_{\mathbf{h}}\sum_{A}^{N_A}\overbrace{<\ \chi^0_\mu \mid \frac{1}{|\mathbf{r}-\mathbf{R_A}-\mathbf{h}|} \mid \chi^{\mathbf{g}}_\nu>}^{T_1}$$

$$E_{Coulomb} = \frac{1}{2}\sum_{\mu\nu}\sum_{\mathbf{g}} P^{\mathbf{g}}_{\mu\nu}\sum_{\sigma,\omega}\sum_{\mathbf{g}'} P^{\mathbf{g}'}_{\sigma\omega}\sum_{\mathbf{h}}\overbrace{<\ \overbrace{\chi^0_\mu\ \chi^{\mathbf{g}}_\nu}^{T_1} \mid \frac{1}{|\mathbf{r_1}-\mathbf{r_2}|} \mid \overbrace{\chi^{\mathbf{h}}_\sigma\ \chi^{\mathbf{h}+\mathbf{g}'}_\omega}^{T_1}\ >}^{T_2}$$

$$E_{Exchange} = -\frac{1}{4}\sum_{\mu\nu}\sum_{\mathbf{g}}\overbrace{P^{\mathbf{g}}_{\mu\nu}}^{T_4}\sum_{\sigma,\omega}\sum_{\mathbf{g}'}\overbrace{P^{\mathbf{g}'}_{\sigma\omega}}^{T_5}\sum_{\mathbf{h}}\overbrace{<\ \overbrace{\chi^0_\mu\ \chi^{\mathbf{h}}_\sigma}^{T_3} \mid \frac{1}{|\mathbf{r_1}-\mathbf{r_2}|} \mid \overbrace{\chi^{\mathbf{g}}_\nu\ \chi^{\mathbf{h}+\mathbf{g}'}_\omega}^{T_3}\ >}^{T_3}$$

Table 1.1. Summary of the HF energy contributions. Further details are given in section (1.1.8), from eq.1.23 to eq.1.34.T_1-T_5 represent the five thresholds.

T_1 - It is used for ${T_{\mu\nu}}^{\mathbf{g}}$, ${Z_{\mu\nu}}^{\mathbf{g}}$ and ${C_{\mu\nu}}^{\mathbf{g}}$ integrals (see tab.(1.1)). They are disregarded if the overlap ${S_{\mu\nu}}^{\mathbf{g}}$ is smaller than 10^{-T_1}; the bielectronic Coulomb integral

$$< \chi_\mu^0 \; \chi_\nu^{\mathbf{g}} \mid \frac{1}{|\mathbf{r_1} - \mathbf{r_2}|} \mid \chi_\sigma^{\mathbf{h}} \; \chi_\omega^{\mathbf{h+g'}} \; >$$

is disregarded also if $S_{\sigma\omega}^{\mathbf{g'}}$ is smaller than 10^{-T_1}. Although **g** and **g'** infinite sums are truncated by T_1, the problem of the infinite **h** sum in the Coulomb term remains: this problem is solved by T_2.

T_2 - Coulomb bielectronic integrals are evaluated either exactly, or, when the two charge distributions do not overlap, through a bipolar expansion. It is known that the electrostatic potential generated by a charge distribution at a point external to it can be approximated by a multipolar expansion of the charge distribution[5].
A total pseudo-charge ρ_A is associated with each atom A (or shell A), and it can be obtained from Mulliken atomic (shell) charge expression:

$$\rho_A(\mathbf{r} - \mathbf{r_A}) = \sum_{\sigma \in A} \sum_{\omega} \sum_{\mathbf{g'}} P_{\sigma\omega}^{\mathbf{g'}} \chi_\sigma(\mathbf{r}) \chi_\omega(\mathbf{r} - \mathbf{g'}) - Z_A \delta(\mathbf{r} - \mathbf{R_A}) \quad (1.44)$$

where the last term is the nuclear charge contribution. The cell charge distribution, in terms of Mulliken atomic charge shown in eq.1.44, is

$$\rho(\mathbf{r}) = \sum_A \sum_{\mathbf{h}} \rho_A(\mathbf{r} - \mathbf{R_A} - \mathbf{h}) \quad (1.45)$$

$$\rho_A(\mathbf{r} - \mathbf{R_A} - \mathbf{h}) = \sum_{\sigma \in A} \sum_{\omega} \sum_{\mathbf{g'}} P_{\sigma\omega}^{\mathbf{g'}} \chi_\sigma(\mathbf{r} - \mathbf{R_A} - \mathbf{h}) \chi_\omega(\mathbf{r} - \mathbf{R_B} - \mathbf{h} - \mathbf{g'})$$

where B is the atom on which the ω orbital is centered. Using these definitions, the Coulomb term in eq.1.29 becomes:

$$C_{\mu\nu}^{\mathbf{g}} = \sum_A \sum_{\mathbf{h}} \int \int_D \rho_{\mu\nu}(\mathbf{r_1} - \mathbf{g}) \frac{1}{\mid \mathbf{r_1} - \mathbf{r_2} \mid} \rho_A(\mathbf{r_2} - \mathbf{R_A} - \mathbf{h}) d\mathbf{r_1} d\mathbf{r_2} \quad (1.46)$$

In these expressions, **h** summation is related to the distance between two interacting pseudo-charges and **h** vectors are subdivided in two sets. The first, {H}, is a finite set where bielectronic integrals are evaluated exactly and correspond to overlapping distributions; the second,$\{H_A\}$, is an infinite region where integrals are approximated by expanding ρ_A into multipoles and evaluating the series analytically to infinity[5]. The

assignment of **h** vectors to H_A depends on the overlap $S_{\mu\nu,A}$ between ${\rho_A}^{\mathbf{h}}$ and $\rho_{\mu\nu}$=$\{\ \chi^0_\mu\chi^g_\nu\}$, that must be smaller than 10^{-T_2}. When the charge distributions ρ_A and $\rho_{\mu\nu}$ do not overlap for any **h**, the relative Coulomb matrix elements are expressed in terms of the multipole moments of ρ_A eq.1.47 and of the field integrals eq.1.48:

$$\gamma_{\ell m}(A) = \int_D \rho_A(\mathbf{r_2} - \mathbf{R_A}) N_{\ell m} Y_{\ell m}(\mathbf{r_2} - \mathbf{R_A}) d\mathbf{r_2} \qquad (1.47)$$

$$\varphi_{\ell m\mu\nu}(\mathbf{g},\mathbf{R_A} + \mathbf{h}) = (1.48)$$
$$= \int_D \chi_\mu(\mathbf{r_1})\chi_\nu(\mathbf{r_1} - \mathbf{g}) Y_{\ell m}(\mathbf{r_1} - \mathbf{R_A} - \mathbf{h}) \Big(\frac{1}{\mid \mathbf{r_1} - \mathbf{R_A} - \mathbf{h} \mid}\Big)^{2\ell+1} d\mathbf{r_1}$$

where $N_{\ell m}$ is a normalization factor and $Y_{\ell m}$ is a real solid spherical harmonic. The infinite **h** summation can be performed by using the Ewald technique, which reduces it to a finite number of terms[12].

T_3 - It is the same as T_1 for the exchange series: the bielectronic integral

$$< \chi^0_\mu\ \chi^{\mathbf{h}}_\sigma \mid \frac{1}{|\mathbf{r_1} - \mathbf{r_2}|} \mid \chi^{\mathbf{g}}_\nu\ \chi^{\mathbf{h+g'}}_\omega >$$

is disregarded if either $S^{\mathbf{h}}_{\mu\sigma}$ or $S^{\mathbf{h+g'-g}}_{\nu\omega}$ are smaller than 10^{-T_3}. Note that the exponential decay of the product $\chi^0_\mu\chi^{\mathbf{h}}_\sigma$ is enough to truncate the **h** summation. For a similar reason, in the product $\chi^{\mathbf{g}}_\nu\chi^{\mathbf{h+g'}}_\omega$ the distance |**g**-(**h**+**g'**)| cannot be too large. However, for every **h**, there are infinite couples of **g** and **g'** vectors such that |**g**-(**h**+**g'**)| $\simeq 0$. This last problem is solved by T_4 and T_5.

T_4,T_5 - For the exchange series **g'** is truncated if $S^{\mathbf{g'}}_{\sigma\omega} < 10^{-T_5}$, and the **g** series is truncated if $S^{\mathbf{g}}_{\mu\nu} < 10^{-T_4}$. In a localized basis set, the elements of the density matrix of an insulator decay exponentially with the distance between the two centers. The same is true for a conductor, even if the decay is much slower[5]: the larger the gap, the faster the decay. The exponential decay of the density matrix permits to disregard integrals involving **g** and **g'** vectors with large moduli. So, the extension of the sums over **g** and **g'** depends on the long range behavior of the density matrix. For T_4 and T_5 not a true overlap is involved: that is why, in this case, we consider T_4 and T_5 *pseudo-overlap* tolerances.

Suppose we are studying a general system, with a number N_A of atoms in the unit cell and a number M_λ of shells on each atom (note that M_λ can have different values if different atoms are present). Two complementary questions arise:

1. As seen above, only a selected number of **g** vectors will be considered, out of the infinite set; however, in order to decide which ones are interesting, should we explore ALL of them? It is clear that we need a criterion for exploring the infinite set, in order to avoid using a huge amount of time for this inspection.

2. With the selected basis, the unit cell contains a number $N_{\lambda TOT}$ of shells (labeled with λ)
$$N_{\lambda TOT} = \sum_A \sum_\lambda \lambda_A \tag{1.49}$$
where the A summation runs over atoms, and a much bigger number of AOs N_{AOs}. Overall, in $F^{\mathbf{g}}_{\mu\nu}$ we will have $N_{\lambda TOT} \times N_{\lambda TOT}$ couples and $N_{AOs} \times N_{AOs}$ matrix elements for each **g** vector. For example, a graphite layer (two atoms per cell) with two shells (s and sp, 5 AOs) on each atom will have 4×4=16 shell couples and 10×10=100 matrix elements for each **g** vector. However, some of the overlaps S_{λ_i,λ_j} might be zero for the FULL set of **g** vectors, including the reference cell, so that none of the S_{λ_i,λ_j} integrals need to be stored. How can we decide which overlap matrix elements have to be disregarded and which one have to be calculate?

Exploiting the sparse nature of the matrices requires a packed labeling of the shell couples, **g** vectors and matrix elements. This is done in the first part of the CRYSTAL code and is discussed in appendix 2.

1.2 Symmetry Properties

Before entering into the CRYSTAL code details, for which the first part has been deeply analyzed in appendix 2, a few words about point symmetry[3] must be said. It is well known that both point and translational symmetry operators are present in crystals: as a consequence, **Space Groups** have to be considered.

The general symmetry operators of a space group O_{SG} can be expressed in the form:

$$O_{SG} =< R|\mathbf{f}(R) + \mathbf{g} > \tag{1.50}$$

where R denotes a proper or improper rotation, $\mathbf{f}$(R) a fractionary translator, and $\mathbf{g}$ a direct lattice vector. The set of rotations R forms a group of order h, the point group of the crystal (only if all $\mathbf{f}$(R)=0). We have already dealt with the translational properties, so in this section we shall discuss how local symmetry can be exploited and how this two kind of symmetry are used together to reduce the number of integrals to be calculated. Most examples will concern the two dimensional group of the graphite monolayer (P_{6mm}), which is just complex enough to illustrate the most important points, while permitting simple graphical representations. The labels for atoms, cells, symmetry operators, are given in fig.(1.4) and in tab.(1.2).

Let's have a look, now, to figs. (1.4) and (1.5); we can split the point symmetry in two complementary effects: the rotation of lattice points (the shells' and AOs' centers) and the one of basis set functions. For example, in fig.(1.4), atom b in cell 1 goes on atom a' in cell 6 by the effect of the C_6^{-1} operator, while, in fig.(1.5), the p_x^b function is rotated on atom a' and becomes a linear combination of p_x and p_y functions.

	$\widehat{R}$	a^R	O_a^R	b^R	O_b^R	$O_i^R - O_i^R$	$O_a^R - O_b^R$	$O_b^R - O_a^R$
1	E	a	1	b	1	1	1	1
2	C_2	b'	2	a"	2	1	1	1
3	C_3	a"	2	b"	6	1	3	4
4	C_3^2	a'	3	b'	2	1	5	6
5	C_6	b"	6	a	1	1	6	5
6	C_6^5	b	1	a'	3	1	4	3
7	σ_d^1	a	1	b"	6	1	5	6
8	σ_d^2	a'	3	b	1	1	3	4
9	σ_d^3	a"	2	b'	2	1	1	1
10	σ_v^1	b	1	a	1	1	1	1
11	σ_v^2	b"	6	a"	2	1	4	3
12	σ_v^3	b'	2	a'	3	1	6	5

Table 1.2. Effect of the point symmetry operators[3] on the unit cell atoms of the graphite monolayer. a^R is the atom to which a is transported by R^{-1}; O_a^R is the label of the direct lattice vector which brings back a^R into the fundamental cell; $O_a^R - O_b^R$ is the label of the difference vector; O_i^R stands for O_a^R or O_b^R; labels according to fig.(1.4). Further details in appendix 2, *GROTA1* subroutine.

In the following the transformation properties of AOs are first obtained, and then used to analyze the symmetry relationships of the various quantities that enter the theory, i.e. one and two electron integrals and sums seen above.

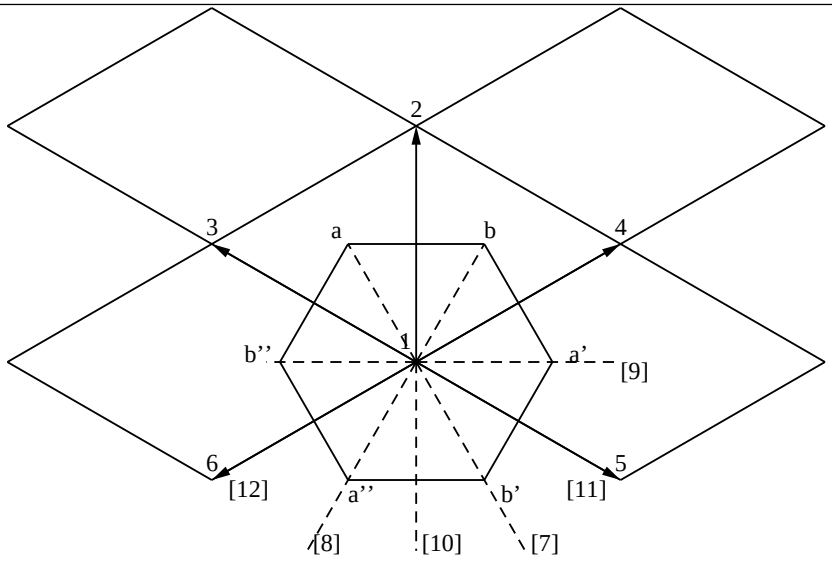

Figure 1.4. Symbols adopted for illustrating symmetry properties of the graphite monolayer[3], see tab.(1.2). Cells are labeled by numbers at their low corner; a,b (a', b' ;a",b") identify the two translationally inequivalent carbon atoms; the traces of symmetry planes are outlined. Numbers in square parenthesis are the symmetry operator labels, as shown in tab.(1.2).

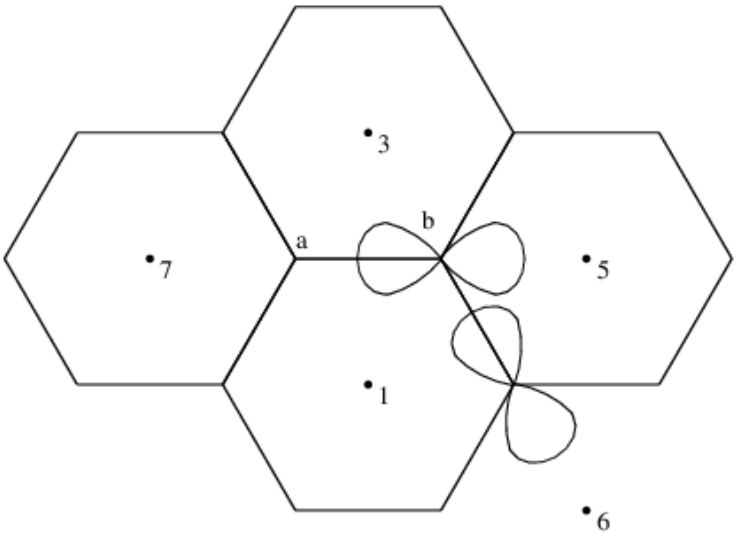

Figure 1.5. Effect of the symmetry operator C_6^{-1} on a p_x AO of atom b[3].

1.2.1 Transformation properties of the Atomic Orbitals

For the purposes of the present section it is convenient to use a new notation for the general atomic orbital[3]:

$$\chi_\mu(\mathbf{r}-\mathbf{g}-\mathbf{R}_\mu) = [s,\ell,m;\mathbf{g}] \tag{1.51}$$

where the shell index s identifies both the fractional coordinate $\mathbf{R}_\mu$ of the atom to which the AO belongs, and its radial shape (depending on n, the main quantum number), ℓ and m are the secondary and the magnetic quantum numbers. In this section, in order to avoid confusions, the general notation for atomic position and atomic orbitals centers will be expressed by $\mathbf{S}$ instead of $\mathbf{R}$. As regards shells, a general shell will be indicated by λ, as in the previous sections, while a specific shell λ_s will be identified by its index s.

When the symmetry operator $\widehat{R}$ is applied to [s,ℓ,m;$\mathbf{g}$] , its effect is to bring the AO in an equivalent shell (s^R) within the crystal cell $\mathbf{g}^R$, and to rotate it.

$$\begin{aligned} \widehat{R}(\mathbf{S}_\mu + \mathbf{g} - \mathbf{f}) = \widehat{R}(\mathbf{S}_\mu - \mathbf{f}) + \widehat{R}\mathbf{g} = \mathbf{S}_\mu^R - \mathbf{O}_\mu^R + \widehat{R}\mathbf{g} - f \\ \widehat{R}(\mathbf{S}_\mu - \mathbf{f}) = \mathbf{S}_\mu^R - \mathbf{O}_\mu^R - f \\ \mathbf{g}^R = \widehat{R}\mathbf{g} - \mathbf{O}_\mu^R \end{aligned} \tag{1.52}$$

where $\mathbf{O}_\mu^R$ is the vector that brings the new center back to the origin cell. The old center is related to the new one by the relation:

$$\mathbf{g} + \mathbf{S}_\mu = \widehat{R}^{-1}\ (\mathbf{g}^R + \mathbf{S}_\mu^R) + \mathbf{f} = \widehat{R}^{-1}\mathbf{g}^R + \widehat{R}^{-1}\mathbf{S}_\mu^R + \mathbf{f} \tag{1.53}$$

For a better explanation, two examples are given. Labels are referred to fig.(1.4).

1. Let's apply the C_6^{-1} operator to the atom b of the origin cell 1 of graphite shown in fig.(1.4).
   $\mathbf{S}_\mu$ = b
   $\widehat{R} = C_6^{-1}$
   $\mathbf{g} = \mathbf{g}_1$
   $\mathbf{f} = 0$
   C_6^{-1} (b +$\mathbf{g}_1$) = a +$\mathbf{g}_6$ +$\mathbf{g}_1$; b goes on a' in cell number 6.
   $\mathbf{O}_b^R$ = -$\mathbf{g}_6$ = $\mathbf{g}_7$
   $\mathbf{g}^R$ = $\mathbf{g}_1$ -$\mathbf{g}_7$ = $\mathbf{g}_6$
   The old center is obtained by the new one:
   b +$\mathbf{g}_1$ = $C_6\mathbf{g}_6$ +C_6 a = $\mathbf{g}_5$ +b -$\mathbf{g}_5$

2. Let's apply, now, the C_6^{-1} operator to the atom a in cell 5 of graphite.
   $\mathbf{S}_\mu$ = a
   $\widehat{R} = C_6^{-1}$
   $\mathbf{g}$ =$\mathbf{g}_5$

$\mathbf{f} = 0$
C_6^{-1} (a $+\mathbf{g}_5$) = b $-\mathbf{g}_1$ $+\mathbf{g}_6$, a in cell 5 goes on b in cell number 6.
$\mathbf{O}_a^R = \mathbf{g}_1$
$\mathbf{g}^R = -\mathbf{g}_1 + \mathbf{g}_6 = \mathbf{g}_6$
The old center is obtained by the new one:
a $+\mathbf{g}_5 = C_6\mathbf{g}_6 + C_6$ b $= \mathbf{g}_5 +$a $-\mathbf{g}_1$

To express the change in orientation of the AO, we shall introduce orthogonal matrices[3] T(R,ℓ) of order $2\ell+1$ which represent the rotation R in the basis set of the real solid harmonics with quantum numbers m and ℓ. We then have:

$$\widehat{R}[s,\ell,m;\mathbf{g}] \equiv [s,\ell,m;\mathbf{g}]_R = \sum_{m'} T(R,\ell;m,m')[s^R,\ell,m';\mathbf{g}^R] \tag{1.54}$$

where m is the T row matrix index, m' the column one and the matrix elements (m,m') are the linear combination coefficients of the AOs belonging to shell s with the same quantum number ℓ. The function obtained by the application of the $\widehat{R}$ symmetry operator on the considered AO has its center in the $\mathbf{g}^R$ cell and coordinates $\mathbf{S}_\mu^R$.

At this point, it is convenient to group all shells λ=[s;$\mathbf{g}$] on the different atoms of the crystal into stars of symmetry related members. Each star is identified by a "generating member" [SG] from which the i^{th} member of the star $[SG]_i$ is obtained by application of a well defined symmetry operator V_i. There is therefore an unique correspondence:

$$[SG]_i \equiv [S^{V_i}G^{V_i}] \longleftrightarrow [s;g] \tag{1.55}$$

(see *GORDSH*, *GV* and *GILDA1* subroutine descriptions in appendix 2).

1.2.2 Transformation properties of direct space integrals and matrices

Consider now a quantity Q_Q which is obtained by integrating over all spatial coordinates the matrix element of a totally symmetric operator $\widehat{Q}$ with respect to two terms that transform, according to eq.(1.54)[3]. Because of the translational symmetry, Q_Q remains unchanged if the centers of the two terms are displaced by the same lattice vector $\mathbf{g}$. We can therefore always assume that

the center of the first term is in the zero reference cell, and define, accordingly:

$$Q_Q^{\mathbf{g}}(\mu_1\mu_2) = Q_Q(s_1,\ell_1,m_1; s_2,\ell_2,m_2,\mathbf{g}) = \int_D [s_1,\ell_1,m_1;\mathbf{0}]\widehat{R}[s_2,\ell_2,m_2;\mathbf{g}]d\tau \tag{1.56}$$

Q_Q is also unchanged if the same $\widehat{R}$ operator is applied to both factors of the integrand. By using eq.(1.54) we obtain:

$$Q_Q(s_1,\ell_1,m_1; s_2,\ell_2,m_2,\mathbf{g}) = \tag{1.57}$$
$$= \sum_{m_1'm_2'} T(R,\ell_1; m_1,m_1')T(R,\ell_2; m_2,m_2')Q_Q(s_1^R,\ell_1,m_1'; s_2^R,\ell_2,m_2',\mathbf{g'}^R)$$

where s^R corresponds to the $\mathbf{S}_\mu^R$ center, defined from eq.(1.52), and the vector $\mathbf{g'}^R$,

$$\mathbf{g'}^R \equiv \mathbf{g}_2^R - \mathbf{g}_1^R = \mathbf{g} - \mathbf{O}_{r_2}^R + \mathbf{O}_{r_1}^R \tag{1.58}$$

is the difference between the two lattice vectors $\mathbf{g}_2^R = \mathbf{g}\text{-}\mathbf{O}_{s_2}^R$ and $\mathbf{g}_1^R = -\ \mathbf{O}_{s_1}^R$ which identify the cell into which the two centers have been carried by the $\widehat{R}$ operator. We can rewrite eq.(1.57) in a matrix form:

$$Q(s_1^R; s_2^R,\mathbf{g'}^R) = X(R,\ell_1,\ell_2)Q(s_1; s_2,\mathbf{g}) \tag{1.59}$$

The twelve $\mathbf{g'}^R$ vectors generated when $\mathbf{g}$=0 for the different atomic couples in graphite are reported in the last three columns of tab.(1.2).

1.2.3 The irreducible set

Since all quantities Q_Q^g, which are the representation in direct space of a totally symmetric operator, transform according to eq.(1.59), we can calculate explicitly the required matrix element for an irreducible set of terms and obtain all the others by means of simple rotations[3]. We discuss here how the irreducible set can be generated for a full exploitation of local symmetry.

The triplets indices [s_1; s_2, $\mathbf{g}$] and [s_1^R; s_2^R, $\mathbf{g'}^R$] (see appendix 2, *GV* subroutine) will be said to be symmetry related to each other; all symmetry related triplets are said to form a family. In order to find all the members of the family to which [s_1^R; s_2^R, $\mathbf{g'}^R$] belongs, we shall apply to it all symmetry operators. Note that all vectors $\mathbf{g'}^R+\mathbf{S}_1^R\text{-}\mathbf{S}_2^R$ of a family form a star of vectors of equal length (see appendix 2, *GCALCO* subroutine).

How can we proceed to organize all triplets in families? How can we decide the representative member of each family? We first select a fundamental set

RS	G	n(RSG)	Family members and generating operators					
aa	1	2	[aa1]1	[bb1]2				
	2	12	[aa2]1	[aa3]10	[aa4]4	[aa5]3	[aa6]8	[aa7]9	[bb2]10
				[bb3]5	[bb4]12	[bb5]11	[bb6]6	[bb7]2	
	8	6	[aa8]1	[aa10]8	[aa12]10	[bb9]5	[bb11]2	[bb13]11	
	9	6	[aa9]1	[aa11]10	[aa13]9	[bb8]6	[bb10]5	[bb12]2	
ab	1	6	[ab1]1	[ab3]3	[ab5]4	[ab1]2	[ab4]6	[ba6]5	
	2	6	[ba2]1	[ba7]9	[ba11]10	[ba2]5	[ba7]2	[ba8]6	
	4	12	[ab4]1	[ab6]9	[ab10]3	[ab12]10	[ab14]8	[ba15]4	[ba3]2
				[ba5]10	[ba9]12	[ba13]6	[ba16]4	[ba17]5	

Table 1.3. Generating couples (RS) and vectors (G) of the symmetry families in the graphite monolayer[3]. Atoms and vectors are labeled as in fig.(1.6), point group operators as in tab.(1.2). For example, [bb1]2 (see first row) means that member [bb1] is obtained from the generating member [aa1] by applying the symmetry operator number 2. Further information about this table will be given in appendix 2.

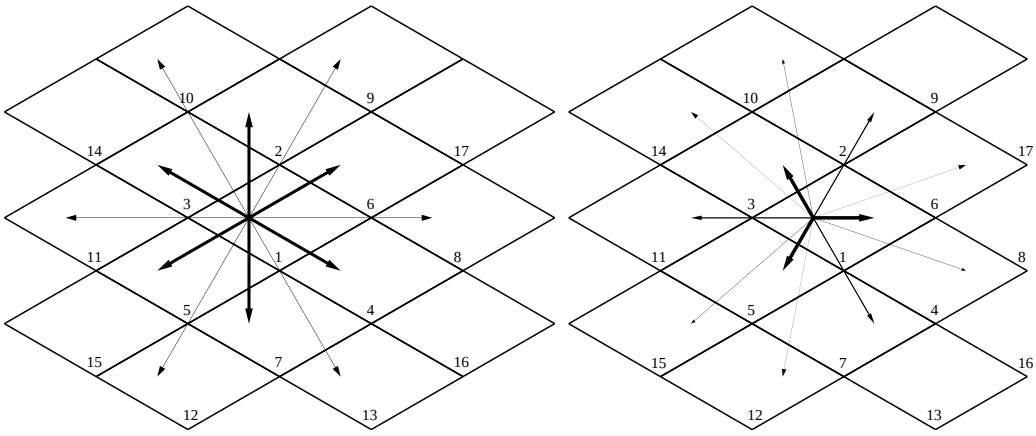

Figure 1.6. Stars of **g** vectors in graphite classified according to the distance **r**=|**S**$_1$-**S**$_2$-**g**|[3]. Thick-solid, solid and thin-solid lines indicate vectors belonging to the first, the second and the third stars, respectively. In the left figure **S**$_1$=a and **S**$_2$=a, two stars; in the right one **S**$_1$=a and **S**$_2$=b, three stars (see appendix 2,$GCALCO$ and GV subroutines).

of shell couples (λ_{s_1},λ_{s_2}) which will be denoted by RS; this set is such that all couples can be generated from them. Next, for each RS couple, we consider lattice vectors of progressively increasing length, |**G**|, starting from **G**=0, and generate the family of triplets that are symmetry related to [RSG]. Each family is identified by its generating member, [RSG], and will contain a certain number n[RSG] of members. The generic ith member of the family, [RSG]$_i$ is obtained by applying to [RSG] a well defined operator $\widehat{R}_i$. The procedure

is carried on until all triplets have been classified with $|\mathbf{G}|$ within a certain maximum radius (defined in *SELEZ* subroutine). This procedure permits us to establish a unique correspondence between the triplets and the set $[\mathrm{RSG}]_i$ (see appendix 2, *GILDA1* subroutine)

$$[RSG]_i \equiv [R^{V_i}; S^{V_i}, G^{V_i}] \longleftrightarrow [s_1; s_2, g] \tag{1.60}$$

Eq.(1.59) can be reformulated as follows

$$Q_Q(RSG)_i = X(RS)_i Q_Q(RSG) \tag{1.61}$$

and shows in particular that the calculation of the matrix $\mathrm{F}^{\mathbf{g}}$ can be restricted to its irreducible part F(RSG).

If, on the contrary, we want to specify a given couple of AOs belonging to the two shells, we must consider not only the geometrical properties, but also the function shapes and their orientation (for example the couples $(2\mathrm{p}_x,2\mathrm{p}_z)$ and $(2\mathrm{p}_y,2\mathrm{p}_z)$ are different): a further index t has to be defined; it can assume values from 1 to $\mathrm{t(RS)}=(2\ell_R+1)(2\ell_S+1)$, and it is such that the following correspondence can be established : $(\lambda_i,\lambda_j,\mathbf{g}) \longleftrightarrow [(\mathrm{RSG})_i,\mathrm{t}]$.

The above discussion can be clarified by the example of graphite. The fundamental couples are then aa and ab, since bb and ba are generated from these by symmetry operations. A few generating vectors **g** of each couple, and the corresponding families are shown in fig.(1.6) and given in tab.(1.3). It results that we must consider 4 and 3 **g** vectors for aa and ab couples, out of a set of 24 and 26 respectively.

1.2.4 Symmetry summary

We can resume in this paragraph, all has been said about symmetry, and answer the questions put forward in the last sections. Remember that the symmetry selection main objectivity is to reduce the number of integral calculations and, as consequence, to avoid a large amount of time and memory.

- By analyzing the transformation properties of AOs and shells, an irreducible set of shells $\lambda=(\mathrm{s};\mathbf{g})$, labeled by [SG], can be defined, with $[\mathrm{SG}]_i$ being symmetry related to [SG] by the $\widehat{R}_i$ operator.

- Consider now a general couple of shell: $[(\mathrm{s}_1;\mathbf{g'}),(\mathrm{s}_2;\mathbf{h})]$. It will always be possible to assume the first term in the zero cell. As consequence, only $[\mathrm{s}_1;\mathrm{s}_2,\mathbf{g}]$ related integrals will be calculated, with $\mathbf{g}=\mathbf{h}\text{-}\mathbf{g'}$.

- As shells are classified by symmetry, an irreducible set of couples of shells [RSG] can be established, with $[\mathrm{RSG}]_i$ obtained by applying the symmetry operator $\widehat{R}_i$ to $[\mathrm{RSG}]_1$. A further reduction of the number of integrals to be calculated is performed: for example, all $[\mathrm{RSG}]_i$ overlap integrals will have the same value.

- A question about the $\mathbf{g}$ vectors remain: how to explore them? We've just seen that lattice vectors can be divided in stars of equivalent $\mathbf{g}$ vectors, having the same length and obtaining from the irreducible $\mathbf{g}$ rotation. So, only the irreducible set of $\mathbf{g}$ will be explored, and, in particular, the analysis is performed by considering irreducible $\mathbf{g}$ vectors of increasing length.

Chapter 2

Introduction to the CRYSTAL code

The CRYSTAL program has the following structure:

```
Program Crystal
  call initpp
  call f90main
  call stoppp
END Program Crystal
```

INITPP and *STOPPP* are two subroutines containing instructions for starting and stopping the program respectively, as indicated by their names. We do not go into details of them, and focus our attemption to the main subroutine, *F90MAIN*. It is structured into three main three main blocks:

- The first block is controlled by the *INT_SCREEN* subroutine. First of all, the *INPUT* routine is called, in order to read the three 'input blocks': geometry (*CRYSTA* subroutine), basis set (*INPBAS* subroutine) and Hamiltonian-SCF (*READM2* subroutine). Then, preliminary calculations and selections according to the T_i tolerances (see later on) are performed, symmetry relations are established, packed labeling is prepared.

- The second block is controlled by the *INT_CALC* subroutine, where the required integrals are evaluated and stored on external units. Note that, under the SCFDIR directive, the integrals are evaluated at each SCF cycle and used directly.

- The third block is controlled by the *SCF* subroutine performing SCF.

In the following some of the subroutines of the *INT_SCREEN* branch will be discussed. Two examples will be used to clarify the meaning of the various quantities:

1. A graphite monolayer with two shells for each carbon atom
2. α-quartz with two shells on oxygen and three on silicon.

In all examples a STO-3G basis set with default exponents is used.

Chapter 3

Symmetry and selection

In this paragraph some of the features of the subroutine *INT_SCREEN* will be discussed; in particular our attention will be focused on the part of the code related to selection and symmetry and shown in table 3.1. The *INT_SCREEN* subroutine calls a sequence of subroutines, each of which builds information stored in a set of compact vectors, that will be used in the following two main blocks of the CRYSTAL code. The meaning of each vector is described in the next paragraphs.

Subroutines	Print parameters (n,m)	Vectors
GROTA1	(6,1)	***MGNAV***, ***LAV***, ***L34A***, LA34
SELEZ		INA12, ILA12T, ICCT, NLA21T, ILANA, NLANA, KVRSP, ICCAT, ICCV
GORDSH	(8,1)	**NCF**, **JNCDU**, **LA3**, **LA4**, NMVLA, ICCS1
GV	(7,LA34F)	**LA34V**, **LA34X**, **NQGSHG**, **NSHG**, **NGSHG**, **ICCS2**
GILDA1	(5,ICOF)	**NGIF**, **NNGI**, **NSHGI**, **GSYMS**, **INO**, **INOIV**, **MGV**, **LSH**, INZVLA
GMFCAL	(10,1)	**NSTATG**, **IDIME**, **IDMF**, **IDMCOU**, **NNNC**, **NNNC2**,**INZVLB**, JPOINT, IROF

Table 3.1. *GROTA1* section in the structure of the *INT_SCREEN* subroutine: in the first column *GROTA1*, that calls the other subroutines reported in the table, is directly called by *INT_SCREEN*. Results can be printed in the CRYSTAL output by setting LPRINT(n)=m (see second column; the meaning of LA34F and ICOF values will be explained in the *GROTA1* and *GORDSH* sections). In the last column all relevant vectors calculated by the corresponding subroutine (stored in the module *MEMORY_SCREEN*) and used in other parts of the code are listed: the printable ones are bold sized, matrices in italic.

3.1 SELEZ

The *SELEZ* subroutine (see table 3.1) estimates the overlap between two shells, disregards shell couples with overlap smaller than e^{-T_x} (with T_x being the maximum among the T_1, T_3, T_4, T_5 thresholds) and lists selected atom and shell pairs in the unit cell (see the end of section 1.1.10). For the overlap evaluation, an adjoined gaussian centered on the corresponding atom is used for each shell. As a general shell is labeled λ_n (with n being an integer), a general couple will be labeled as (λ_i,λ_j), where i = 1,LAF and j = 1,LAF; $LAF = inf(20)$ is the total number of shells.

When the size of the cell or the basis set per atom increases, the number of shell couples becomes very large. However, many of the couples give a negligible contribution to the summations (see table 1.1 in section 1.1.10), because of the low or null overlap, so that it is convenient to disregard them, in order to avoid waste of memory.

As a couple is characterized by two indices (λ_i,λ_j), information on shell pairs might be conveniently stored in matrix form. For example, in the graphite case (example 1 in section 2), which couples have a significant overlap? When T_x=4 is chosen, the following matrix can be built, where 1 is for overlaps $S_{\lambda_i,\lambda_j} > 10^{-T_x}$ and 0 corresponds to $S_{\lambda_i,\lambda_j} < 10^{-T_x}$.

	λ_1	λ_2	λ_3	λ_4
λ_1	1	1	0	1
λ_2	1	1	1	1
λ_3	0	1	1	1
λ_4	1	1	1	1

Table 3.2. The shell overlaps matrix.

For such a small system only 2 out of 16 elements should be discarded; however, larger cells behave differently. Can we reduce the number of matrix elements, by avoiding the storage of null terms? Can we rearrange information?

The use of arrays instead of matrices permits both a lower memory occupation and a faster memory access. As more than one index is involved, pointers, that permit to store data in a compact form, need to be defined.

Suppose we store sequentially the 14 elements of table 3.2, indicating in vector $v2$ the column label, and in vector $v1$ the starting point in $v2$ of the

compacted rows:
$v1(\lambda_i)$=0 3 7 10 14
$v2(\lambda_j)$=1 2 4 1 2 3 4 2 3 4 1 2 3 4
This list is more compact and convenient than the matrix in table 3.2, because neglected pairs are automatically excluded. The number of accepted neighbors of shell λ_i is given by

$$v1(\lambda_i + 1) - v1(\lambda_i) \tag{3.1}$$

and they are stored in $v2$ starting from $v1(\lambda_i)$+1. So, for example, for shell λ_3, $v1(4)$-$v1(3)$=10-7=3: there are three neighbors; they can be found in $v2(8)$, $v2(9)$, $v2(10)$ and are shells 2, 3, 4, as shown in the third row of table 3.2. Note that $v1$ is $LAF + 1$ long, as it must indicate the last element of $v2$ (see equation 3.1, where $\lambda_i + 1$ appears).

One could also decide a more compact storage: as the matrix in table 3.2 is symmetric, only half a matrix could be stored.

$$\begin{aligned} v1(\lambda_i) &= 0\ 1 \quad 3 \quad 5 \quad 10 \\ v2(\lambda_j) &= 1\ 12 \ \ 23 \quad 1234 \end{aligned} \tag{3.2}$$

All quantities evaluated by the CRYSTAL code are stored in a compact form, by using schemes similar to the one sketched above. As regards pointers, it has to be said that in the CRYSTAL code the value of the pointer first element, in some cases is 0, in some others is 1. By referring to example 3.2, the pointer could also be written as:
$v1(\lambda_i)$=1 2 4 6 11
In this case eq.(3.1) remains valid, but the first neighbor of λ_i, in $v2$, is position $v1(\lambda_i)$ instead of $v1(\lambda_i)$+1 (see above).

Let us come back to our *SELEZ* subroutine; all the logic in the CRYSTAL integral package is based on the shell idea; however, in order to speed up the selection, it is more convenient to perform a first selection on an atomic basis, considering only the most diffuse shell of the atom. A general couple of atoms will be indicated as (at_i,at_j) It is clear that if the (at_i,at_j) overlap is disregarded (less than 10^{-T_x}), ALL the related (λ_{at_i},λ_{at_j}) overlaps will be disregarded; on the contrary, if (at_i,at_j) is accepted, all the (λ_{at_i},λ_{at_j}) couples have to be checked.

The *SELEZ* subroutine can be divided into three parts: the first one concerns the atomic selection. At first, the largest adjoined gaussian of each

atom is used to select atom pairs, and two arrays are evaluated: **ICCAT** and **INA12**, with a similar structure to $v1$ and $v2$ respectively (but for atoms instead of shells). **INA12** lists the at_j atoms of all the selected (at_i,at_j) pairs, with $i \geq j$ in order to avoid double counting; **ICCAT(i)** (with i = 1,NAF+1, where $NAF = inf(24)$ is the number of atoms in the unit cell), points the starting position for the (at_i,at_j) couple, with at_i = 1,NAF. The total number of selected (at_i,at_j) couples is ICCAT(NAF+1)-1, and corresponds to the size of INA12; this value is stored as **MAX_ATOM_COUPLES**.

As an example, α-quartz (example 2), with three Si (label=1 2 3) and six O (label=4 5 6 7 8 9) in the unit cell, has been used to build tab.(3.3), that shows the dependence of the pair selection on T_x. When $T_x > 8$ (not shown) no pair is discarded and we obtain the full list of 45 pairs [$(9+1) \times frac92$]. For T_x=8 only three pairs are dropped; if T_x is lowered to 4, the number of retained pairs is reduced, and for T_x=2 it decreases to 24.

TOLINTEG = 1 1 1 1 2 $T_x = 2$
ICCAT
1 2 4 7 10 13 16 19 22 25
INA12
1 1 2 1 2 3 1 3 4 1 2 5 2 3 6 1 2 7 1 3 8 2 3 9
TOLINTEG= 2 2 2 2 4 $T_x = 4$
ICCAT
1 2 4 7 11 15 19 23 27 31
INA12
1 1 2 1 2 3 1 2 3 4 1 2 3 5 1 2 3 6 1 2 3 7 1 2 3 8 1 2 3 9
TOLINTEG= 4 4 4 4 8 $T_x = 8$
ICCAT
1 2 4 7 11 16 22 28 35 43
INA12
1 1 2 1 2 3 1 2 3 4 1 2 3 4 5 1 2 3 4 5 6 1 2 3 4 6 7 1 2 3 5 6 7 8 1 2 3 4 5 7 8 9

Table 3.3. ICCAT and INA12 vectors for α-quartz (3 shells on Si and two on O, STO-3G basis set with default exponents) with different tolerance values (TOLINTEG= T_1 T_2 T_3 T_4 T_5). Neighbors of atom 1 (overlap>10^{-T_x}) start at the first place in INA12, neighbors of atom 2 start in the second position in INA12 and are atoms 1 and 2, neighbors of atom 3 starting in the fourth position are atoms 1, 2 and 3, and so on.

Once the selection has been performed on atoms, it is repeated for shell couples belonging to the selected atoms. Note that the overlap depends not only on the TOLINTEG values, but also on the basis set, and, in particular, on the exponents: overlap between two diffuse Gaussian functions is larger than the one between two Gaussian functions with large exponents.

At the beginning of the second part of the subroutine three vectors are calculated: **ILA12T**, **ICCT** and **NLA21T**. **ICCT** and **ILA12T** are built exactly as ICCAT and INA12 respectively, but refer to shells; note, however, that ICCT(1)=0 (ICCAT(1)=1) and that ILA12T contains both couples (λ_i,λ_j) and (λ_j,λ_i), whereas INA12 only (at_i,at_j) with $i \geq j$. **ILA12T** size corresponds to ICCT(LAF+1)=**MAX_G_COUPLES** value (the total number of couples of shells). **NLA21T** contains the position of the transposed couple (λ_j,λ_i) in the ILA12T vector: it gives the couple with transposed-index label. **ICCV** and **KVRSP** vectors, sized respectively $(LAF + 1)$ and $(LAF \times LAF)$, provide the reverse information with respect to ILA12T. In particular, given a pair of shells (λ_i,λ_j), KVRSP$((\lambda_i - 1) \times LAF + \lambda_j)$ contains the sequence number of that pair in the compact ILA12T list, *i.e.* the position in ILA12T, whereas ICCV stores all $(\lambda_i - 1) \times LAF$ products. Table 3.4 shows these quantities calculated for graphite.

ICCT	0				3				7			10				14	
ILA12T	1	2		4	1	2	**3**	4		2	3	4	1	2	3	4	
NLA21T	1	4		11	2	5	**8**	12		6	9	13	3	7	10	14	
KVRSP	1	2	0	3	4	5	6	7	0	8	9	10	11	12	13	14	
ICCV	0				4				8				12				16

Table 3.4. ICCT, ILA12T, NLA21T, KVRSP and ICCV vectors for graphite, TOLINTEG=(2 2 2 2 4), $LAF = 4$. Consider, first, ILA12T and ICCT: for example, shell 3 is a neighbor of shell 2 (because it is in position 6, and neighbors of shell 2 go from 3 to 7). Then, we want to answer the question: where is couple (λ_j,λ_i) in ILA21T? This information is contained in NLA21T: for example, where is couple (4 1)? In ILA12T $\lambda_i = 1$ and $\lambda_j = 4$, so (1 4) is the third in ILA12T (couple (1 3) have been disregarded). In NLA21T(3) we read 11: the (4 1) couple is in ILA12T(11). The relation between the two quantities can be expressed as ILA12T(λ_i,λ_j)=NLA21T(λ_j,λ_i) and $\widetilde{ILA12T}(n)$=NLA21T(n) (with n = 1,**MAX_G_COUPLES**). As a further example, let's consider the couple (1 2): it is in the second position of ILA12T. NLA21T(2)=4; ILA12T(4)=(2 1). Consider, now, KVRSP and ICCV. Which is, for example, the sequence number of the couple (3 4)? KVRSP((3-1)×4+4)=KVRSP(12)=10. And which is the couple number 13? ILA12T(13)=λ_j=3. KVRSP(ICCV(λ_i)+3)=13, ICCV(λ_i)+3=15, ICCV(λ_i)=12, $\lambda_i = 4$. The couple number 13 is then (4 3).

At this point, a question arise: given a couple of atoms, can we go back to the corresponding shell couple? Part of this information is lost: the shell couple list is incomplete because the null terms have been discarded. It is clear that a new quantity has to be defined. To this purpose, in the last part of

the subroutine, **ILANA** and **NLANA** vectors are built, and will be used any time the neighbor analysis has to be performed (*NEIGHBOR* subroutine).

ILANA contains the position of the couples of shells in NLA21T, and for each atom couple (at_i,at_j) lists the relative couple of shells (only the previously selected ones). NLANA is a pointer on each atom couple in ILANA: it indicates the starting point for each couple of atoms. In the example of graphite, then, NLANA is a 3+1=4 size vector: there are 3 couples of atoms (11, 12, 22). In order to avoid useless information storage, a further selection is performed, considering that the couples (λ_j,λ_i) and (λ_i,λ_j) are the same: only the couple with i<j is store. When both λ_i and λ_j belong to the same atom, however, both (λ_i,λ_j) and (λ_j,λ_i) couples are stored; in the graphite example ILANA is then a 11 size vector, as shown in tab.(3.5).

ILANA	1	2	4	5	8	11	12	9	10	13	14	
Couples	11	21	12	22	23	14	24	33	43	34	44	
NLANA	1				5			8				12

Table 3.5. ILANA and NLANA for the graphite case. For positions 1 to 4 in ILANA $\lambda_i \in at_i$, at_i=1, $\lambda_j \in at_j$, at_j=1; for positions 5 to 7 at_i=1 and at_j=2; for positions 8 to 11 at_i=2 and at_j=2.

3.2 GROTA1

After the overlap selection (see *SELEZ*), symmetry effects are analyzed and the matrices and vectors listed in table 3.1 for *GROTA1* are calculated.

In order to exploit both the translational and the point symmetry and reduce the number of integrals to be calculated, in the *GROTA1* and related subroutines the effect of point symmetry operators on atoms and shells of the unit cell is analyzed.

MGNAV is a $(NAF) \times (MVF)$ matrix ($MVF = inf(2)$ is the number of point symmetry operators) that shows which lattice vector brings back the atom s in the zero-cell when the $\widehat{R}^{-1}$ operator is applied to it (see equation 1.53).

LAV is a $(LAF) \times (MVF)$ matrix that shows on which shell goes a shell λ_s by the effect of a symmetry operator $\widehat{R}^{-1}$, see equation 1.53 (shells equivalent by translation do have the same label; then in graphite shells labels are numbers from 1 to 4.

NAF	MVF											
	1	2	3	4	5	6	7	8	9	10	11	12
1	1	2	6	2	6	1	1	6	2	1	2	6
2	1	2	2	4	1	4	4	1	2	1	4	2

Table 3.6. MGNAV matrix for graphite, with 2 atoms (NAF) in the unit cell and 12 (MVF) operators.

LAF	MVF											
	1	2	3	4	5	6	7	8	9	10	11	12
1	1	3	1	1	3	3	1	1	1	3	3	3
2	2	4	2	2	4	4	2	2	2	4	4	4
3	3	1	3	3	1	1	3	3	3	1	1	1
4	4	2	4	4	2	2	4	4	4	2	2	2

Table 3.7. LAV matrix for the graphite example with 4 shell (LAF) and 12 (MVF) operators.

For a better understanding, see figures 1.4, 1.5, and table 1.2 in section 1.2, and compare them to tables 3.6 and 3.7 in this section.

Then, shell couples in the unit cell are classified as a function of the vector joining them. If $\mathbf{R}_i$ and $\mathbf{R}_j$ are the positions of shells λ_i and λ_j on atoms at_i and at_j (at_i=at_j or $at_i \neq at_j$), all couples (λ_i,λ_j) with different $\mathbf{R}_{ij}$=($\mathbf{R}_i$-$\mathbf{R}_j$) are listed; note that ($\mathbf{R}_i$-$\mathbf{R}_j$)$\neq$($\mathbf{R}_j$-$\mathbf{R}_i$). If there is more than one couple with the same ($\mathbf{R}_i$-$\mathbf{R}_j$), for example (λ_i,λ_j) and (λ_k,λ_l) with $\mathbf{R}_{kl}$=$\mathbf{R}_{ij}$, and such that

$$\widehat{R}^{-1}\mathbf{R}_{ij} = \widehat{R}^{-1}\mathbf{R}_{kl} \tag{3.3}$$

for all $\widehat{R}$ operators, only the first couple for each type is considered. The case where the two shells coincide is treated as a special case. If the atom contains more than one shell, two types are selected (in spite of the fact that always ($\mathbf{R}_{ij}$=$\mathbf{0}$), the first with $j > i$, the second with $j < i$). The reason for this exception is related to the hermitian character of the matrix, and will be analyzed later on.

The result of this analysis is stored in the **L34A** matrix, sized $2 \times LA34F$, where $LA34F = inf(73)$ is the number of type of independent couples of shells (CT). L34A contains the CT.

Then, each couple is related to one of the LA34F CT: the **LA34** array (**MAX_G_COUPLES** size) shows to which CT the couples belong. Given a

Type	1	2	3	4
λ_i	2	1	3	2
λ_j	1	2	2	3

Table 3.8. L34A matrix for the graphite example, with TOLINTEG=(2 2 2 2 4). Note that both (λ_i,λ_j) and (λ_j,λ_i) couples are considered (see *SELEZ*).

generic couple (λ_i,λ_j), its couple type CT is obtained by CT=LA34(KVRSP(($\lambda_i-1)\times LAF+\lambda_j$)). For example, suppose we are interested to know the CT of couple (3 3). First, we must find the position of couple (3 3) in the ILA12T vector; using the KVRSP (see *SELEZ*):
$(\lambda_i - 1) \times LAF = (3-1) \times 4 = 8$
$KVRSP(8 + \lambda_j) = KVRSP(11) = 9$, the ILA12T position of couple (3 3)
We look, then, in LA34 to obtain the CT of (3 3): LA34(9)=1=CT(3 3).

λ_i,λ_j	11	12	14	21	22	23	24	32	33	34	41	42	43	44
LA34	1	2	4	1	1	4	4	3	1	2	3	3	1	1

Table 3.9. Couples ordered as in ILA12T and LA34 vector for the graphite example, TOLINTEG=(2 2 2 2 4).

3.3 GORDSH

In the *SELEZ* subroutine, we have a first classification that reduces the number of couples according to the overlap criteria, whereas the following part of *GROTA1* evaluates the effect of the point symmetry operators on atoms and shells and classifies the shell couples by distance vectors in different couple types *CT*. The set of couples, resulting from *SELEZ*, is reorganized in the *GORDSH* subroutine in symmetry related subsets and reduced according to hermiticity. Both classifications are present and used at any stage of the code. It is clear that we need vectors that permit to go from one classification to the other one. In this section, we will discuss how the symmetry sets are generated and stored, and how the correspondence among the 2 classifications is maintained.

Consider again the graphite example: the total number of possible shell couples is 16. We apply now the symmetry operators to each couple. The

carbon labels are a and b. 6 out of 12 operators transform a to b (and vice versa, see section **??**), then shell 1 to 3 and shell 2 to 4, and, obviously, each couple is transformed by these operators in another couple (1 1 to 3 3, for example). Then the 16 couples (14 if T_x=4 is used) will be organized in 8 (7) couple sets; within a SET, the second couple can be generated by symmetry from the first one. We consider the case with T_x=4: the 14 couples are organized as shown below (note that the label of the first shell is always equal or larger than the label of the second shell in the first couple). Note also that sets "2 3 4 1" and "1 2 3 4" can be generated from "3 2 1 4" and "2 1 4 3" by exploiting the hermitian character of the matrices.

1 1	**3 3**
2 1	**4 3**
2 2	**4 4**
(2 3)	(4 1)
3 2	**1 4**
(1 2)	(3 4)
4 2	**2 4**

Then only 5 sets are stored (6 for T_x>4); memory will be kept of the fact that 2 sets will be used to generated 2 other sets; they will be attributed a WEIGHT equal to 2 in our case (see column 5 in table 3.11).

The label of the first and second shells of a couple (ordered from 1 to 10) is stored in the **LA3** and **LA4** vectors, respectively. The starting point of each set in LA3 and LA4 is given by the **NCF** array.

LA3	1	3	2	4	2	4	**3**	1	4	2
LA4	1	3	1	3	2	4	**2**	4	2	4
NCF	0		2		4		6		8	

Table 3.10. NCF, LA3 and LA4 vectors for graphite. For example, couple (3 2), which is couple number 7, appears in position 7 in both LA3 and LA4.

The total number of couple sets corresponds to **MVLAF**= $inf(56)$. The couple sets are then classified in type of couple sets (CST): couples with the same CST behave in the same way under symmetry operators effect; for each CST a "mother" couple set is defined, and corresponds to the first couple set of that CST in the symmetry order. Note that couple sets are classified under geometrical properties only (so that, for example, (1 1) and (3 3) behave as (2 2) and (4 4) if we just consider the shells as a spherical object; when taking

into account the shape of the functions, obviously, the two sets are different). The number of *CST* is indicated as **ICOF**.

The link between the MVLAF couple sets and the ICOF mother couple sets is the pointer array **NMVLA**: it has ICOF size and shows the position of the mother couple sets in the MVLAF sets. We will see later in this section an example of NMVLA. Preliminary, the meaning of the quantities printed with LPRINT(8)=1 is explained. In *GORDSH*, the instructions corresponding to the printed results, organized in two tables, are:

```
 WRITE(IOUT,305)ICOF
      DO 302 IC=1,ICOF
      IFN=NMVLA(IC)
      INZ=NCF(IFN)
      IFN=NCF(IFN+1)
      ID1=IFN-INZ
302   WRITE(IOUT,303)IC,ID1,INZVLA(IC),(LA3(I),LA4(I),I=INZ+1,IFN)
!!end of the first table
 WRITE(IOUT,245)MVLAF
      DO 202 IC=1,MVLAF
      INZ=NCF(IC)
      IFN=NCF(IC+1)
      ID1=IFN-INZ
      L3=IROF(IC)+1
202   WRITE(IOUT,250)IC,L3,ID1,INZ,JNCDU(IC),(LA3(I),LA4(I),I=INZ+1,IFN)
!!end of the second table
```

The first printed table contains information about the mother couple sets; result for graphite (TOLINTEG=(2 2 2 2 4)) are shown in table 3.11.

- NUMBER OF TYPES OF COUPLE SETS = **ICOF**.

- TYPE: couples of the same type behave in the same way when symmetry operators of the group are applied. This means, in our graphite example, that couples can be divided in two groups by referring to their behavior when the 12 symmetry operators are applied. The numbers in this column correspond to an index, IC, running from 1 to **ICOF**.

- SIZE is the number of couples for each type, and it is obtained from the **NCF** vector, that gives the position of each set. SIZE=NCF(IC+1)-NCF(IC), with IC=1,ICOF.

- BASE column contains the starting point on LA3(I) and LA4(I) vectors for the couples of the set. In the part of code shown before, the corresponding vector is INZVLA; however, the INZVLA array will be

recalculated and stored in the *GILDA1* subroutine, and it will have another meaning.

- COUPLES OF THE SET are the "mother" couples for each type of couple set, the corresponding vectors are **LA3(I)** and **LA4(I)**, where I is referred to the first couple for each type.

```
GORDSH --- NUMBER OF TYPES OF COUPLE SETS     2
  TYPE  SIZE  BASE                 COUPLES OF THE SET
    1     2     0                    1   1    3   3
    2     2     2                    3   2    1   4
```

Table 3.11. First table printed by setting LPRINT(8)=1 for graphite.

In table 3.12 information about all the MVLAF couple sets is stored:

- NUMBER OF COUPLE SETS = **MVLAF**.

- SET: this column contains only the couple label, corresponding to an index, IC, running from 1 to **MVLAF**.

- TYPE has the same meaning as in table 3.11, but here all the non-symmetry related couples of the sets, and not only the mother-couples of different type, are considered. Type column corresponds to **L3** vector, which is obtained from IROF(i) vector, with i=1,MVLAF, by the relation L3(i)=IROF(i)+1, as shown in the code. For each set, L3 shows to which type of set it belongs. The meaning of IROF will change in a subsequent part of the code (subroutine *GMFCAL*).

- SIZE is the number of couples for each type, and it is obtained from the **NCF(IC)** array in the same way as already explained, but in this case IC=1,MVLAF.

- BASE is a pointer vector corresponding to **NCF(IC)** vector and containing, for the couple (λ_i,λ_j), the starting i and j on LA3 and LA4 vectors for each set.

- WEIGHT is the number of sets that can be generated by hermiticity from the corresponding set; this column corresponds to **JNCDU(IC)** vector, with IC=1,MVLAF.

- COUPLES OF THE SET columns contain all the couples for each set. The first shell of each couple belongs to **LA3** and the second one to **LA4**.

```
GORDSH --- NUMBER OF COUPLE SETS     5
  SET  TYPE  SIZE  BASE WEIGHT   COUPLES OF THE SET
   1     1     2     0      1      1   1    3   3
   2     1     2     2      2      2   1    4   3
   3     1     2     4      1      2   2    4   4
   4     2     2     6      2      3   2    1   4
   5     2     2     8      1      4   2    2   4


  *LA3   (1 3 2 4 2 4 3 1 4 2)
  *LA4   (1 3 1 3 2 4 2 4 2 4)
  *IROF  (0 0 0 1 1)
  *NMVLA (1 4)
```

Table 3.12. Second table printed by setting LPRINT(8)=1. * Not printed vectors.

Let us come back, now, to **NMVLA**, shown in table 3.12 for the present example.
NMVLA(1)=1, couple set number 1 [(1 1),(3 3)] in table 3.12 is the first mother set (CST number 1) in table 3.11
NMVLA(2)=4, couple set number 4 [(3 2),(1 4)] in table 3.12 is the second mother set (CST number 2) in table 3.11

Now, consider again the graphite example when TOLINTEG=(6 6 6 6 12). The couples of interest are stored in the ILA12T vector and LA3, LA4 vectors (see table 3.13). The vector that permits to go from ILA12T to LA3, LA4 and vice versa is **ICCS1(k)**, with NCF(MVLAF+1) size, containing the position in ILA12T of the couple LA3(k),LA4(k) (see table 3.13).

```
...
DO MVLA=1,MVLAF
 DO K=NCF(MVLA)+1,NCF(MVLA+1)
  L1=LA3(K)
  L2=LA4(K)
  IS=ICCS1(K)
 ENDDO
ENDDO
....
```

For reasons of completeness we will explain the **JPOINT** array here in the *GORDSH* section, even though it is calculated in the *GMFCAL* subroutine. **JPOINT** has the same size as ILA12T (ICCT(LAF+1)=MAX_G_COUPLES),

k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
LA3(k)	1	3	2	4	2	4	3	1	3	1	4	2				
LA4(k)	1	3	1	3	2	4	1	3	2	4	2	4				
ICCS1	1	11	5	15	6	16	9	3	10	4	14	8				
JPOINT	1	2	4	5	2	3	5	6	4	5	1	2	5	6	2	3
ILA12T	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
couples	11	(12)	13	14	21	22	(23)	24	31	32	33	(34)	(41)	42	43	44

Table 3.13. La3, LA4, ICCS1, JPOINT, ILA12T for graphite with TOLINTEG=(6 6 6 6 12). Couples in the last lines are in the ILA12T order, couples into parenthesis are obtained by hermiticity from the corresponding transposed ones. k is an index running from 1 to ICCT(LAF+1). Where is, for example, the couple (1 4) in the symmetry order? We have ILA12T(KVRSP(ICCT(1)+4))=ILA12T(4)=(1 4), so ICCS1(xx)=4, and xx=10: the couple is the 10^{th} in the symmetry order. On the contrary, if we are interested in going from the symmetry to the sequential order, we can pose the question: where is the couple (3 3) in the sequential order? For that couple, k=2, so ICCS1(2)=11: it is the 11-th in the ILA12T sequence. As concerns JPOINT, which set, for example, does couple (2 3) belong to? ILA12T(KVRSP(ICCT(2)+3)=ILA12T(7)=(2 3), JPOINT(7)=5: couple (2 3) belongs to the 5-th couple set.

and relates each couple to its couple set: it answers the question: for a given couple, in the ILA12T sequence, which set does it belong to? For a better understanding, see table 3.13.

3.4 GCALCO

Once the shell couples (λ_i,λ_j) of interest have been selected (disregarding those such that none of the $S^{g}_{\lambda_i\lambda_j} > T_x$ for all the infinite set of **g** vectors, see *SELEZ*) and grouped in family sets, according to symmetry relations (see *GORDSH*), we must take into account the translation vectors (individuated as **g**). As usual, we aim to use labels rather than vectors or mathematical operators (es. distances). The global list of **g** vectors is performed in the *GCALCO* subroutine, called at the beginning of the calculation (before the *GROTA1* call), into the *CRYSTA* subroutine. A large cluster, containing N cells (N$\sim$3000), is built and the **g** vectors of the cells are then ordered in stars (vectors with the same module |**g**|); within a star, the lattice vectors are ordered according to their components (eg. 100, 010, 001...). The **g** vectors are then stored, and can be printed by setting LPRINT(1)=n, with n indicating the number of the **g** vectors we want to be printed. Note that this LPRINT has to be typed in the input geometry section. In the following the first 11 **g**

for the graphite case are shown.

```
GCALCO - MAX INDICES DIRECT LATTICE VECTOR  54  54   0
NO.OF VECTORS CREATED 2999 STARS  232 RMAX   131.19405

STAR N.   1 N. OF VECTORS   1 R=  0.0000000E+00
  LG I(LG) LX   LY   LZ         X           Y           Z
   1    1    0    0    0        0.000000    0.000000    0.000000

STAR N.   2 N. OF VECTORS   6 R=  4.5731372E+00
  LG I(LG) LX   LY   LZ         X           Y           Z
   2    3   -1   -1    0       -3.960453   -2.286569    0.000000
   3    2    1    1    0        3.960453    2.286569    0.000000
   4    5   -1    0    0       -3.960453    2.286569    0.000000
   5    4    1    0    0        3.960453   -2.286569    0.000000
   6    7    0   -1    0        0.000000   -4.573137    0.000000
   7    6    0    1    0        0.000000    4.573137    0.000000

STAR N.   3 N. OF VECTORS   6 R=  7.9209061E+00
  LG I(LG) LX   LY   LZ         X           Y           Z
   8    9   -2   -1    0       -7.920906    0.000000    0.000000
   9    8    2    1    0        7.920906    0.000000    0.000000
  10   11   -1   -2    0       -3.960453   -6.859706    0.000000
  11   10    1    2    0        3.960453    6.859706    0.000000
  12   13   -1    1   ....      ........    ........    ........
  [............]
```

Let's analyse, now, the printed quantities. The first string contains the maximum index of direct lattice vectors (I1,I2,I3 in the code; in this case I3=0 because we are considering a bidimensional system): the graphite cluster is built by adding 54 unit cells in all direction. The second string contains the number of created **g** (NGPRI, it is 2999 by default), the corresponding total number of stars (NSTAR) in which the lattice vectors are subdivided, and the maximum module of the **g** vectors (R in the code, it corresponds to the module of the 2999-th vector).

All the 2999 **g** vectors are then listed in NSTAR sequences. For each star the label of the star (LSH=1,NSTAR), the number of lattice vector in that star (M=J-N, where J is the label of the first **g** in the LSH+1 star and N the one of the first **g** in the LSH star) and the module of its vectors (R) are given. Consider, for example, the 2^{nd} star, LSH=2: J=8 (the first **g** of the LSH+1=3-rd star), N=2 (the first **g** of the LSH=2-nd star), M=8-2=6 (number of the lattice vector in the star).

The LG and I(LG) column, are respectively the sequence of the **g** vectors and their reverse vectors. For example, in the second star, for **g**=8, -**g**=9.

The corresponding variables in the code are: NN1(M) for the reverse vectors sequence (I(LG)) and M, running from N to J-1, for the **g** sequence (LG).

Finally, for each **g** vector, two triplets of numbers exist: the first one corresponds to the Bravais lattice vector indices (in the code: LG(I,M), with I running from 1 to 3 -the space dimension), the second one to their Cartesian coordinates (XG(I,M)).

Labels attributed to the **g** vectors *GCALCO* permit to access to their Cartesian coordinates, and then to perform mathematical operations, such as distances ($\mathbf{D}=|\mathbf{R}_\mu\text{-}\mathbf{R}_\nu\text{-}\mathbf{g}|$, for example) in any part of the code.

3.5 GV

The matrix elements we are interested to classify do not depend directly on $|\mathbf{g}|$, but on $|\mathbf{R}_\mu\text{-}\mathbf{R}_\nu\text{-}\mathbf{g}|$ (see section (1.1)), so couples have to be classified with a (λ_i,λ_j,**g**) label (see section (1.2)). For this reason we must define a set of **g** vectors for each CT (see *GROTA1*; they are 4 for graphite).

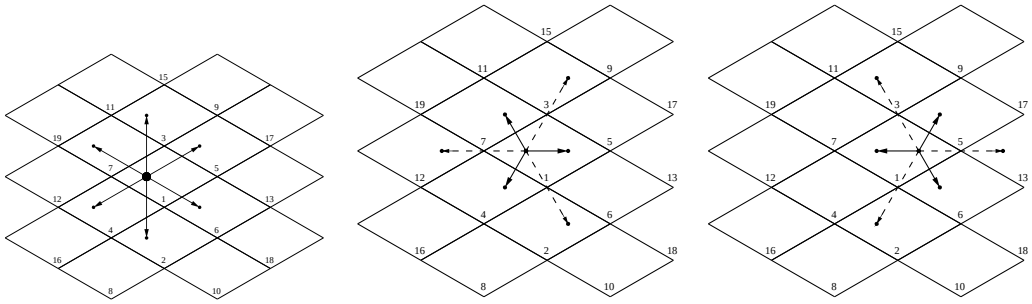

Figure 3.1. In sequence: first star of g for couple (1 2), first (solid) and second (dashed) stars of g for couples (1 3) and (3 1).

Obviously, for each couple, there will be a " maximum interacting distance", beyond which the considered shells do not interact anymore (overlap less than T_x): on the basis of the chosen thresholds, only a certain number of **g** for each couple will be considered. For each couple type we need to know the involved **g** vectors, their organization in stars (note that a star can contain **g** vectors such that $|\mathbf{R}_\mu\text{-}\mathbf{R}_\nu\text{-}\mathbf{g}_i|= |\mathbf{R}_\mu\text{-}\mathbf{R}_\nu\text{-}\mathbf{g}_j|$, but such that they cannot be obtained from each other by symmetry) and their label in the *GCALCO* sequence. This information is stored in two vectors: **NQGSHG**, that contains the ordered

list of all the **g** vectors for each CT and each star, and **NGSHG**, that permits to find the starting point of a star of a given CT.

TYPE,L1,L2=1 2 1 NUMBER OF G-STARS=6

STAR	SIZE	G-VECTORS					
1	1	1					
2	6	2	3	4	5	6	7
3	6	8	9	10	11	12	13
4	6	14	15	16	17	18	19
5	12	20	21	22	23	24	25
		26	27	28	29	30	31
6	6	32	33	34	35	36	37

TYPE,L1,L2=2 1 2 NUMBER OF G-STARS=6

STAR	SIZE	G-VECTORS					
1	1	1					
2	6	3	2	5	4	7	6
3	6	9	8	11	10	13	12
4	6	15	14	17	16	19	18
5	12	21	20	23	22	25	24
		27	26	29	28	31	30
6	6	33	32	35	34	37	36

TYPE,L1,L2=3 3 1 NUMBER OF G-STARS=9

STAR	SIZE	G-VECTORS					
1	3	1	4	7			
2	3	2	3	12			
3	6	5	6	8	11	16	19
4	6	9	10	14	15	26	30
5	3	13	22	29			
6	6	17	18	20	25	34	37
7	3	21	24	42			
8	6	23	28	32	33	50	54
9	6	27	31	38	41	46	53

TYPE,L1,L2=4 1 3 NUMBER OF G-STARS=9

STAR	SIZE	G-VECTORS					
1	3	1	5	6			
2	3	3	2	13			
3	6	4	7	9	10	17	18
4	6	8	11	15	14	27	31
5	3	12	23	28			
6	6	16	19	21	24	35	36
7	3	20	25	43			
8	6	22	29	33	32	51	55
9	6	26	30	39	40	47	52

Table 3.14. The LA34F tables printed by setting LPRINT(7)=LA34F for graphite. The first number at the top of the table is a label (TYPE in the tables, L34 in the code), running from 1 to LA34F (see *GROTA1*), while the following couple of numbers is the considered CT stored in L34A matrix by *GROTA1*; the number of g-stars is NSHG(CT) all the other numbers are obtained from NSHG, NGSHG and NQGSHG arrays. In this case TOLINTEG is (5 5 5 5 12) instead of (2 2 2 2 4) used in tab.(3.8), so that also the 13 and 31 couples give a non-zero overlap for a set of **g** vectors.

In particular, **NQGSHG** is a (LA34F×MGFDMF) size array, where MGFDMF=inf(145) corresponds to the max number of **g** vectors allocated on each CT; it contains the sequence of all **g** vectors of all CT, according to the *GCALCO* global list. In the case of graphite NQGSHG has 4x42=168 dimension, and all of them, apart from the null ones, are listed in tab.(3.14). The whole NQGSHG vector appears as follows:

```
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21
22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37  0  0  0  0  0
 1  3  2  5  4  7  6  9  8 11 10 13 12 15 14 17 16 19 18 21 20
23 22 25 24 27 26 29 28 31 30 33 32 35 34 37 36  0  0  0  0  0
 1  4  7  2  3 12  5  6  8 11 16 19  9 10 14 15 26 30 13 22 29
17 18 20 25 34 37 21 24 42 23 28 32 33 50 54 27 31 38 41 46 53
 1  5  6  3  2 13  4  7  9 10 17 18  8 11 15 14 27 31 12 23 28
16 19 21 24 35 36 20 25 43 22 29 33 32 51 55 26 30 39 40 47 52
```

Note that all labels in NQGSHG refer to the labels attributed to each **g** vector in the *GCALCO* routine.

But, how to find, for example, the **g** vectors of star 4 of the third CT? The **NGSHG** array has (LA34F×LSTDMF) size, where LSTDMF=inf(37) is the maximum number of **g** vectors per couple set, and contains the information for the starting point of every star in NQGSHG. In this case it is

```
  0   1   7  13  19  31  37   0   0   0 0 [...]
 42  43  49  55  61  73  79   0   0   0 0 [...]
 84  87  90  96 102 105 111 114 120 126 0 [...]
126 129 132 138 144 147 153 156 162 168 0 [...] 0
```

(When a quantity is allocated, its size must be previously defined, and must be valid for any calculation. However, in this case, the the number of star for each couple is not known before the *GV* analysis, but it is always known that the number of stars is less or equal to the maximum number of **g** vectors per couple. This is the reason why this last vector has such a big size, even if almost all its components are null terms.)

Let's answer, now, to the previous question. We are interested to know where is the starting point of the 4-th star **g** information for CT 3. The position is given by:
NGSHG(STAR+(CT-1)*LSTDMF)=NGSHG(4+(3-1)*43)=NGSHG(90)=96
NQGSHG(96+1)=9
The 4-th star of the CT number 3 starts in the 96+1=97 position in NQGSHG vector (96+1 because NGSHG(1)=0). Compare with the 3-rd table in tab.(3.14): for STAR=4 the first **g** is the number 9. Suppose we also want to know where the 4-th star of the 3-rd CT ends in the NQGSHG array: we first must establish where the 5-th star starts.
NGSHG(5+(3-1)*43)=102
NQGSHG(102)=30
The star goes from **g** labeled from 9 to 30 in the NQGSHG order, from the position 97 to the position 102 in the NQGSHG array.

As usual the number of **g** in the i-th star is given by the difference
NGSHG(i+1)-NGSHG(i)=Number of **g**
The **NSHG(i)** array , with i=1,LA34F, stores the number of **g** stars for each CT and, in the specific case, it is (6 6 9 9). So, given a CT, the label in the STAR column runs from 1 to **NSHG(CT)**. The total number of stars for each CT is obtained on the basis of the selected tolerance T_x (see sections (1.1) and (3.1)). For example, if TOLINTEG values are 4 4 4 4 8, or 2 2 2 2 4, NSHG is (5 5 6 6), or (3 3 3 3).

In calculating the integrals, we will deal with the couples; as the classification of the (λ_i,λ_j,**g**) is performed per CT, not for each single couple (we

remind for example that in graphite couples 11 12 and 22 belongs to the same type of couple), we must be able to go directly from the couple (λ_i,λ_j,**g**) to its **g** organization, skipping all passages leading a general couple to the reference couple. That is because if there are too many passages, a lot of vectors and indices are involved from different subroutines, with the consequence of an increased calculation cost. On this purpose, **LA34X** and **LA34V** vectors have been calculated. They have **MAX_G_COUPLES** size (see *GROTA1*), that is the total number of couples of shells, and are referred to couples ordered in the ILA12T array (see *SELEZ*). For every couple of shells **LA34X** shows the number of g-stars, while **LA34V** contains the starting point in NQGSHG (see above in this section) for the **g** vectors.

KVRSP	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
LA34X	6	6	9	9	6	6	9	9	9	9	6	6	9	9	6	6
LA34V	0	43	129	129	0	0	129	129	86	86	0	43	86	86	0	0
ICCV	0				4				8				12			
ILA12T	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Couples	11	12	13	14	21	22	23	24	31	32	33	34	41	42	43	44

Table 3.15. LA34X, LA34V, ICCV, ILA12T and couples ordered as in ILA12T, KVRSP for graphite. How many stars, for examples, for couple (4 3)? KVRSP(ICCV(4)+3)=15 (see *GROTA1* and *SELEZ*), LA34X(15)=6: 6 stars are allocated on couple (4 3). And where do they start? LA34V(15)=0: the **g** vectors for couple (4 3) start in the first position in NQGSHG, as for couple (1 2), so CT(4 3)=CT(1 2)=2. Let's consider, as a further example, the couple (2 3): KVRSP(ICCV(2)+3)=7, LA34X(7)=9=number of stars allocated on (2 3); LA34V(7)=129: the **g** vectors for couple (2 3) start in the 129-th position in NQGSHG, as for couple (1 3), so CT(2 3)=CT(1 3)=4.

ICCS2 has the same size as ICCS1 vector calculated in *GORDSH* subroutine, and contains the starting point of couples in ICCS1 sequence in NGSHG. We remind that couples are ordered not only in the "natural" sequence (ILA12T), but also in the "symmetry" order (ICCS1). We are interested to know the starting point in NGSHG of couple in the symmetry order as well.

3.6 GILDA1

The subroutine *GILDA1* starts from the classification of the couple of shells performed by *GORDSH* and classifies the **g** vectors by symmetry, in order to find the irreducible ones for each couple set. As anticipated in the *GV* section,

ICCS1	1	11	5	15	6	16	9	3	10	4	14	8
Couples	11	33	21	43	22	44	31	13	32	14	42	24
ICCS2	0	0	0	0	0	0	86	129	86	129	86	129

Table 3.16. ICCS1, couples in the symmetry order and ICCS2 for graphite. Couple 33, for example, is in the 11th position in the sequential order of ILA12T, is the second one in the symmetry order, and the corresponding **g** vectors start at point 0 in NGSHG array; couple (1 4) is in the 4-th position in the sequential order of ILA12T, in the 10-th one in the symmetry order, and the corresponding **g** vectors start at point 129 in NGSHG array.

all **g** belonging to the same star have the same module; so, for each star, there will be one, two or more irreducible **g** that can represent the whole star of g. Results (see tab.(3.1)) are organized and printed in a set of ICOF tables. In our graphite example ICOF=2, so two different sets of irreducible **g** exist. To clarify the meaning of all calculated quantities the first CST printed table for graphite will be taken as an example.

```
               ICO=1,ICOF                                N1    Number of corresponding
                                                                non-null g in NQGSHG
*TYPE OF COUPLE   1     G-VECTORS CLASSIFIED TO STAR   6    (  37)
*TYPE OF COUPLE   2     G-VECTORS CLASSIFIED TO STAR   9    (  42)

*Printed strings
```

At the beginning, before the printed tables, all couples are listed; N1 in the code corresponds to the number of g stars of the mother-couple in the LA34X sequence. In the code:

```
...
 ICO=1,ICOF
 MVLA=NMVLA(ICO)
 INZ=NCF(MVLA)+1
 N1=LA34X(ICCS1(INZ1))
...
```

The analysis results of the first CST (first table) starts as follows.

```
                    ICO                        NCFF=NCF(MVLA+1)-NCF(MVLA)
 *SET TYPE OF COUPLES  1     NUMBER OF COUPLES    2
 *THEY ARE    1  1      3  3
              from LA3(I),LA4(I)
```

In the code:

```
...
DO ICO=1,LPR5
      MVLA=NMVLA(ICO)
      INZ=NCF(MVLA)
      IFN=NCF(MVLA+1)
      NCFF=IFN-INZ
      WRITE(IOUT,763)ICO,NCFF,(LA3(I),LA4(I),I=INZ+1,IFN)
      NO=NGIF(ICO)
...
```

We need, now, to identify the **g** vectors, for each CST, able to generate all the star of **g** vectors of all the other symmetry related couple sets. Then, for each CST, we need to know:

- Where does the **g** vectors sequence for a CST start in the NQGSHG array (**INZVLA** vector)?
- How many G irreducible for each CST (**NGIF** vector for the g-stars classified in GV)? And for each star (**NNGI** array)?
- How are they distributed in each star (**NSHGI** array)?
- How sons are generated by mother (**MGV**, **INO** and **INOIV** vectors)?
- We need obviously to know their GCALCO labels, in order to have their Cartesian coordinates.

INZVLA array has ICOF size and contains the starting position of the **g** vectors for each CST in NQGSHG. In this case it is (0 84), that means: the first CST [(1 1),(3 3)] **g** vectors start in the INZVLA(1)+1=1-st position in NQGSHG, the second CST [(3 2),(2 3)] **g** vectors start in the INZVLA(2)+1=85-th position in NQGSHG.

NGIF array has ICOF size and contains the number of irreducible **g** for each CST. It is related to **NNGI** vector, that lists all irreducible **g** for each CST: **NGIF** is a pointer on each CST in **NNGI**.

NSHGI is a ICOF×NGDMF sized vector, where NGDMF=inf(39), the maximum number of irreducible **g** four a couple set type; it contains information about the distribution of irreducible **g** in the stars (see the four tables in the *GV* section).

GSYMS is a vector with the same NSHGI size, that stores the symmetry weight of each couple set type. What means "symmetry weight"? Let's explain with an example. Consider the vector number 2 in NNGI. When all the

NNGI	1	2	8	9	14	20	21	32	0	1	2	5	9	13	17	21	23	27
NGIF								8										9

Table 3.17. NNGI and NGIF vectors for graphite. The maximum number of irreducible **g** is established, and it is inf(39)=9 in this case. Where do the CST irreducible **g** start? The starting point of a given CST is obtained by (CST-1)*inf(39)+1. In this case, for example, CST=1 starts in the (CST-1)*inf(39)+1=1-st position in NNGI and CST=2 starts in the (CST-1)*inf(39)+1=10-th position in NNGI. Where does the list for each CST end? It ends in the (CST-1)*inf(39)+NSHG(CST) position. For example, the second CST list of irreducible **g** vectors, starting in the 10-th position, ends in the (2-1)*9+NSHG(2)=9+9=18-th position, and contains NSHG(2)=9 vectors.

NSHGI	1	1	2	1	2	1	0	0	0	1	1	1	1	1	1	1	1	1

Table 3.18. NSHGI vector for graphite. Compare to tab.(3.19). The first CST **g** irreducible set is composed by 8 vectors. For a given star of a given CST, which is the corresponding irreducible **g**? For example, which is the irreducible **g** of the 4-th star of CST 1? For CST=1, NGDMF(CST-1)=0, NSHGI(n+NGDMF(CS-1))=NSHGI(n). NNGI(NSHGI(1+0)+NSHGI(2+0)+NSHGI(3+0)+NSHGI(4+0))=NNGI(1+1+2+1)=NNGI(5)=14. **g**=14 is the irreducible vector in the 4-th star. When, given a star, more than one irreducible vector exist, how can we find all their labels? For example, which are the irreducible **g** of the third star of CST 1? NNGI(NSHGI(1)+NSHGI(2)+NSHGI(3))=NNGI(1+1+2)=NNGI(4)=9; by this equation we obtain the last irreducible **g** of the star: in order to find the first one, we calculate the last irreducible **g** of the previous star and add "+1" to its position. NNGI(NSHGI(1)+NSHGI(2)+1)=NNGI(3)=8. Let's take the 5-th star of the same CST: NNGI(NSHGI(1)+NSHGI(2)+NSHGI(3)+NSHGI(4)+1)=NNGI(6)=20: it is the first vector. NNGI(NSHGI(1)+NSHGI(2)+NSHGI(3)+NSHGI(4)+NSHGI(5))=NNGI(7)=21, the second orreducible **g**.

symmetry operators are applied on vector 2, 6 vectors are generated. Remember, now, that there are two couples related by symmetry, 11 and 33. So the symmetry weight grows to 12. We need this information because when a group of integrals are symmetry related, only one of them need to be calculated: in this case only the matrix elements labeled with (1,1,g=2) are calculated.

GSYMS	2	12	6	6	12	12	12	12	0	6	6	12	12	6	12	6	12	12

Table 3.19. GSYMS vector.

```
NUMBER OF IRREDUCIBLE G-VECTORS   8  THEY ARE  (8 from NGIF)
1    2    8    9   14   20   21   32          (the first 8 vectors in NNGI)

THEIR SYMMETRY WEIGHTS ARE
 2  12  6  6 12  12  12  12                    (the first 8 terms in GSYMS)

DISTRIBUTION IN THE STARS
   1    1    2    1    2    1                 (NSGI for the first couple set g)
```

As for each symmetry operator R_i there is a correspondence $[ab\mathbf{g}] \rightarrow [a^R b^R \mathbf{g}^R]$ (see section (1.2) we are also interested to show, within a star, and given a **g** vector, which is the operator that brings this **g** vector to the corresponding irreducible **g**, and vice versa. **MGV** is a vector with (ICOF×NGDMF×MVF) size (NGDMF×MVF=MGFBI=inf(33), the total number of **g** for each CST originated by symmetry from the irreducible set); it tells which symmetry operator brings a general **g** of the star on a $\mathbf{g}_{irr}$. Results for the first CST are shown in tab.(3.20). We can answer, for example, the question: which operator brings vector 18 to its irreducible **g** (that is to the $\mathbf{g}_{irr}$ that can generate it)? The third and the twelfth operators transform it into the irreducible **g** number 13. We also know, as consequence, that the reverse of the third and the twelfth operators bring, applied to the irreducible **g** number 13, generate the 18-th **g**. In this way, we can go directly from the irreducible set to the full set of **g**.

```
FOR EACH SYMMETRY OPERATOR IV THE G (INDEXED FROM ZERO)GENERATING G IRREDUCIBLE IS REPORTED
G IRR. ;          IV=1,MVF
   1   1       0   0   0   0   0   0   0   0   0   0   0   0  (vectors of
   2   2       1   2   6   4   3   5   5   3   2   1   4   6    NNGI are labeled with 0 as
   3   8       7   8  10  12  11   9  12   7  10   9   8  11    starting point)
   4   9       8   7   9  11  12  10  11   8   9  10   7  12
   5  14      13  14  18  16  15  17  17  15  14  13  16  18
   6  20      19  20  28  26  25  27  30  21  24  23  22  29
   7  21      20  19  27  25  26  28  29  22  23  24  21  30
   8  32      31  32  36  34  33  35  35  33  32  31  34  36
```

Table 3.20. Symmetry operators effect on the irreducible **g** of the first CST (11 and 33 couples). The first column numbers are indices running from 1 to NGIF(1), the second column lists the irreducible vectors as in NNGI and the third one labels them by starting from 0 instead of 1. Numbers in the MVF columns are referred to the third column labels.

Symmetry operators in graphite can be divided in two groups: the first group contains the six operators that do not move atom a on atom b (see section (1.2)), the second one contains the six operators that move atom a on atom b. In the last part of this section, the effect of symmetry operators

on **g** vectors are deeply analyzed, in order to find which operators act on the irreducible **g** vectors of each couple of the CST to generate the other vectors; in our graphite example: which operators act on the irreducible **g** vectors of the couple 11 to generate the other vectors? And which ones can generate the **g** vectors of the 33^{th}couple?

```
SET TYPE  1 POSITION OF THE COUPLE IN THE SET  1COUPLE  1  1
 THE G IRR. (1ST ROW) AND THE SYMM. OPERATOR (2ND ROW) GENERATING EACH G-VECTOR IS REPORTED
 STAR  ; G GENERATED

     1  (star of g 1)
     0  (irreducible g labeled with 0 as starting point)
 *   1  (number of the symmetry operator)
     2  (second star of g)
     1    1    1    1    1    1
 *   1    9    8    3    7    4
     3
     2    3    3    2    3    2
 *   1    1    4    4    3    3
     4
     4    4    4    4    4    4
 *   1    9    8    3    7    4
     5
     5    6    5    6    6    5    6    5    6    5    6    5
 *   1    1    8    8    9    9    3    3    4    4    7    7
     6
     7    7    7    7    7    7
 *   1    9    8    3    7    4
```

Note that in stars 3 and 5 two vectors are necessary: this is why two vectors are chosen in these stars instead of one (see NSHGI vector); and note that only six operators are used. For couple 33 the other six ones will be used. The meaning of the table is the following. Let's take, for example, the third star.

```
 3
     2    3    3    2    3    2
 *   1    1    4    4    3    3
```

For couple 11 the fifth **g** of the star is obtained by applying the third operator on the third irreducible **g** vector. Labels of the lattice vectors, only in this case, do not refer to the *GCALCO* list: the irreducible vectors for each couple of the CST are enumerated from 0 to NGIF(CST)-1.

```
 SET TYPE  1 POSITION OF THE COUPLE IN THE SET  2COUPLE  3  3
 THE G IRR. (1ST ROW) AND THE SYMM. OPERATOR (2ND ROW) GENERATING EACH G-VECTOR IS REPORTED
 STAR  ; G GENERATED

     1
```

```
      0
 *    2
      2
      1    1    1    1    1    1
 *   10    2    6   11    5   12
      3
      3    2    2    3    2    3
 *    2    2    5    5    6    6
      4
      4    4    4    4    4    4
 *   10    2    6   11    5   12
      5
      6    5    6    5    5    6    5    6    5    6    5    6
 *    2    2   11   11   10   10    6    6    5    5   12   12
      6
      7    7    7    7    7    7
 *   10    2    6   11    5   12
```

Two more tables are prepared: **INO** and **INOIV**. The first one contains the sequence of all irreducible **g** in the order shown in the tables (0 1 1 1 1 1 1 2 3 3 2 3 2....). Its size is NHSQ×(number of mother-couples), where NHSQ=INF(145) is the maximum number of g for each couple. In particular, all sequence of all tables are listed (remember that in this case LA34F=4). **INOIV** is the same as **INO** but lists the symmetry operators generating the **g** vectors from the irreducible **g** (es. 1 1 9 8 3 7 4 ...).

As an example of how to use the whole information stored until this point, suppose we want to evaluate the minimal set of integrals $T^{g}_{\mu\nu}$. We have to introduce the *MONMAD* subroutine, that construct the Fock matrix from the irreducible information. It is directly called by the *INT_CALC* block and, in the following, part of the corresponding code is shown and discussed. In particular, the attemption must be paid to the loop on couple sets to choose the irreducible (λi,λj), the loop on stars of **g** and, within a star, on the irreducible **g**, to consider only the irreducible set (λi,λj,$\mathbf{g}^{irr}$); at this point all $T^{\mathbf{g}_{irr}}_{\mu\nu}$ with $\mu \in \lambda_i$ and $\nu \in \lambda_j$ are evaluated.

```
      SUBROUTINE MONMAD                                          1%
      ...
      MVLA=0                                                     2%
9997  CONTINUE                                                   3%
      MVLA=MVLA+1                                                4%
      IF(MVLA.EQ.MVLAF)GOTO 997                                  5%
      NGSH=IDMCOU(MVLA)                                          6%
      IF(NGSH.EQ.0)GOTO 1                                        7%
      L2=NCF(MVLA)                                               8%
      L1=LA3(L2+1)                                               9%
```

```
L2=LA4(L2+1)                                                    10%
X1OLD=XLOLD(1,L1)                                               11%
Y1OLD=XLOLD(2,L1)                                               12%
Z1OLD=XLOLD(3,L1)                                               13%
...
MOCF=LATAO(L1)*LATAO(L2)                                        14%
IROSE=IROF(MVLA)                                                15%
ICOSTA=NSTATG(MVLA)                                             16%
ULT=JNCDU(MVLA)                                                 17%
NUGIR=IROSE                                                     18%
DO MGISH=1,NGSH                                                 19%
DO NGI=1,NSHGI(IROSE+MGISH)                                     20%
NUGIR=NUGIR+1                                                   21%
UOT=GSYMS(NUGIR)*ULT                                            22%
MG=NNGI(NUGIR)                                                  23%
BA(1)=XG(1,MG)+X2(1)                                            24%
BA(2)=XG(2,MG)+X2(2)                                            25%
BA(3)=XG(3,MG)+X2(3)                                            26%
...
WRITE(IOO1)UOT,ICOSTA,MOCF,MOCFL                                27%
CALL MONWRI(IOO1,ATZ,OTZ,SHIFT,CJ,MOFLST,MOCF,MOCFL,LAF)        28%
ICOSTA=ICOSTA+MOCF                                              29%
ENDDO                                                           30%
ENDDO                                                           31%
GOTO 9997                                                       32%
```

line 1 Begin of the *MONMAD* subroutine: overview of the construction of the Fock matrix from the irreducible information

line 2 Initialization of the variable "MVLA", which describes for each cycle a couple set.

line 4 Plus one to the variable "MVLA", equivalent to a DO loop. The first value is the couple set 1.

line 5 Exit when the "MVLA" variable is equal to "MVLAF", which is the number of the last couple set.

line 6 How many g-stars are interesting? overlap and other rules determine for each couple set the relevant number of stars to study.

line 8,9,10 "L2" is a pointer in the "LA3" and "LA4" vectors. NCF contains the positions for each couple set of the first couple. For example: the NCF(2)=2; the first shell of the first couple of the couple set 2 is "2" ("LA3(2+1)=2"); the second shell of this couple is "1" ("LA4(2+1)=1").

line 11,12,13 access to shell coordinates.

line 14 Dimension of the bloc that we are going to build in the Fock matrix , number of OA in the first shell L1 times number of OA in the second shell L2 of the couple.

line 15 Give the type of couple set. We assume shells as points. The couple set 5 is equivalent to the couple set 6, which are equivalent to the couple set 4. For example, the couple set 4 is the second type of couple set. The symmetry information is stored only for the type of couple set.

line 16 Positions in the irreducible matrix of the starting point of each couple set.

line 17 Storing of the weight of each couple set related to the hermiticity reduction.

line 19 For the selected couple set, a do loop on g stars.

line 20 For the selected g star, a do loop is done on irreducible g-vectors.

line 21 The loop read one after one the values of NNGI and GSYMS vectors.

line 23 Access to the irreducible g-vector.

line 24,25,26 Access to the coordinates of the g-vector

Once the irreducible terms have been evaluated, by applying the point symmetry operators the lower half of the Fock matrix is obtained, and, then, the whole matrix is built by hermiticity.

3.7 GMFCAL

In the *GMFCAL* subroutine information calculated in the previous subroutines is used to organize the Fock and density matrices. In particular, the size and the the couple positions in reducible and irreducible matrices are estimated. Before entering into further details, we must remember that, in CRYSTAL, a matrix is stored in a vector form, and elements are ordered first with respect to the row index and second with respect to the column index (eg. 11 12 13 .. 21 22 23 ..).

The four matrices have the following structure.

$P^{\mathbf{g}_{irr}}$ is the irreducible density matrix, and contains only the blocks $P^{\mathbf{g}_{irr}}_{\lambda_i\lambda_j}$ such that the couple (λ_i,λ_j,$\mathbf{g}^{irr}$) belongs to the irreducible set (all other sets can be generated from this set by symmetry; see section 1.2.3) and that $S^{\mathbf{g}_{irr}}_{\lambda_i\lambda_j} > T_{max1}$. $\mathbf{g}_{irr}$ are the irreducible $\mathbf{g}$ vectors of the couple (λ_i,λ_j), and T_{max1} is the maximum among T_1, T_4 and T_5 thresholds. $inf(19) = ICONTA$ is the size of P^{irr}

$F^{\mathbf{g}_{irr}}$ is the irreducible Fock matrix, and contains only the blocks $F^{\mathbf{g}_{irr}}_{\lambda_i\lambda_j}$ such that the couple (λ_i,λ_j,$\mathbf{g}^{irr}$) belongs to the irreducible set and that $S^{\mathbf{g}_{irr}}_{\lambda_i\lambda_j} > T_{max2}$, with T_{max2} being the maximum among T_1 and T_4. As T_5 is always higher than T_4 (so that all the exchange integrals whose second pseudocharge is not negligible are taken into account), P is always larger than the F. In order to simplify the code, a fix allocation for both the irreducible matrices is chosen, and it corresponds to the size of the irreducible density matrix (the highest criterium is used): so, the irreducible Fock matrix will contain a lot of null terms.

$P^{\mathbf{g}_{red}}$ is the whole density matrix, and contains all the $P^{\mathbf{g}}_{\lambda_i\lambda_j}$ elements generated by symmetry from $P^{\mathbf{g}_{irr}}$. $inf(12) = NPGT$ is the $P^{\mathbf{g}_{red}}$ size (also LIM022 in the code).

F^{red} is the lower half Fock matrix and contains all elements generated from $F^{\mathbf{g}_{irr}}_{\lambda_i\lambda_j}$ with i>j by applying the point symmetry operators; $inf(11) = NFGT$ is the lower half $F^{\mathbf{g}}$ size (also LIM021 in the code).

We show the first information printed by setting LPRINT(10)=1 for graphite, with two different TOLINTEG settings (the maximum index for the **g** vectors is $MGF = inf(28)$; note that only the Fock matrix size changes by increasing the first four threshold exponents).

```
DIMENSIONS  P(G)=3176  F(G)=822   P(G),F(G) (IRR)= 314
MAX G-VECTOR INDEX FOR 1- AND 2-ELECTRON INTEGRALS  55  (TOLINTEG 5 5 5 5 12)

DIMENSIONS  P(G)=3176  F(G)=873   P(G),F(G) (IRR)= 314
MAX G-VECTOR INDEX FOR 1- AND 2-ELECTRON INTEGRALS  55  (TOLINTEG 6 6 6 6 12)
```

3.7.1 The irreducible matrices

NSTATG, with size MVLAF+1, contains the starting point of each couple set in the irreducible Fock and density matrices. The number of elements for the set i is given by NSTATG(i+1)-NSTATG(i). In the considered case of graphite it is:

```
  NSTATG    0    1   21  149  150  170  314
```

IDIME, **IDIMF**, **IDMCOU** are three MVLAF sized arrays, and contain, respectively, the number of g-star to be considered for each couple set in the P, F and S matrices. As an example, these quantities are shown for graphite, with two threshold sets.

```
STARS OF G-VECTORS OF THE (L1,L2) COUPLE FOR P(G)
   1    4    6    1    5    9

STARS OF G-VECTORS OF THE (L1,L2) COUPLE FOR F(G)
   1    2    4    0    3    4 (TOLINTEG 5 5 5 5 12)
   1    2    4    1    3    5 (TOLINTEG 6 6 6 6 12)

STARS OF G-VECTORS OF THE (L1,L2) COUPLE FOR S(G)
   1    2    4    0    3    4 (TOLINTEG 5 5 5 5 12)
   1    2    4    1    3    5 (TOLINTEG 6 6 6 6 12)
```

As already said, the highest threshold exponent is used to establish the irreducible matrix sizes, and that it corresponds to the size of the irreducible P matrix. Then, the array containing the number of g-stars to explore for each CS (IDIME) have to be considered. The number of corresponding irreducible **g** vectors for each star is given by NSHGI array in *GILDA1*:

```
NSHGI(1+(CST-1)NGDMF)=NSHGI(1+(1-1)*9)=         1
NSHGI(1+0)+NSHGI(2+0)+NSHGI(3+0)+NSHGI(4+0)=    5
NSHGI(1+0)+NSHGI(2+0)+..... +NSHGI(6+0)=  8
NSHGI(1+(2-1)*9)=                               1
NSHGI(1+10)+NSHGI(2+10)+.....+NSHGI(5+10)= 5
NSHGI(1+10)+NSHGI(2+10)+.....+NSHGI(9+10)= 9
(for the first three couple set CST=1, for the other CST=2)
```

Let's build NSTATG:

Shell	1	2	3	4
AO	1s	4×2sp	1s	4×2sp

1) NSTATG(1)=0

2) Couple Set number 1 (CS1): 11 33 (1s,1s; 1 $\mathbf{g}_{irr}$)

number of irreducible elements (nie)=1

NSTATG(2)=NSTATG(1)+nie=0+1

3) CS2: 21 43 (1s,4×2sp; 5 $\mathbf{g}_{irr}$)

nie=1×4×5=20

NSTATG(3)=NSTATG(2)+nie=21

4) CS3: 22 44 (4×2sp,4×2sp; 8 $\mathbf{g}_{irr}$)

nie=4×4×8=128

NSTATG(4)=NSTATG(3)+nie=149

5)...

If we want to calculate the number of elements of the irreducible Fock matrix, we have to follow the same scheme, but we must refer to IDIMF instead of IDIME. The number of irreducible **g** for the couple sets can be obtained from NSHGI: 1 3 5 0 3 4 (TOLINTEG 5 5 5 5 12).

1) NSTATGdiF(1)=0

2) Couple Set number 1 (CS1): 11 33 (1s,1s; 1 $\mathbf{g}_{irr}$)

number of irreducible elements (nie)=1

NSTATGdiF(2)= NSTATGdiF(1)+nie=0+1

3) CS2: 21 43 (1s,4×2sp; 2 $\mathbf{g}_{irr}$)

nie=1×4×2=8

NSTATGdiF(3)= NSTATGdiF(2)+nie=9

4) CS3: 22 44 (4×2sp,4×2sp; 4 $\mathbf{g}_{irr}$)

nie=4×4×5=80

NSTATGdiF(4)= NSTATGdiF(3)+nie=89

5)...

NSTATGdiF 0 1 9 89 89 113 177

How are these elements stored in the 314 size Fock vector? The number of elements for each couple set is defined by NSTATG. For couple set [(2 1),(4 3)], for example, 20 positions are available, and only the first 8 are occupied by non-null terms; for CS [(2 2),(4 4)], 128 positions are present, but only the first 80 contain non-zero terms, and so on.

3.7.2 The density matrix

The whole density matrix is obtained by applying the point symmetry operators and then the hermiticity to the irreducible matrix. All the couple (λ_i,λ_j,$\mathbf{g}$) generated from the irreducible set, with $S^{\mathbf{g}}_{\lambda_i\lambda_j} > T_{max1}$, are considered, and the **NNNC** array is built: it points the starting position in the P matrix for each couple, and its size corresponds to the total number of couple in the couple sets. In the graphite example it is (second row lists the couples in the symmetry order):

```
   FIRST POSITION OF THE (L1,L2) COUPLE IN P(G)
NNNC    0   1   2   154   306   898   1490   1493  1496  1664  1832  2504
(L1 L2) 11  33  21  43    22    44    31     13    32    14    42    24
```

Let's see, now, how to calculate the number of density matrix elements.

Shell	1	2	3	4
AO	1s	4×2sp	1s	4×2sp

IDIME(1 4 6 1 5 9)

The corresponding total number of **g** vectors for each couple set is obtained by NQGSHG array, in the *GV* subroutine (see table 3.14); the results are the following:

IDIME(1)=1, CST=1, 1 **g** vector
IDIME(2)=4, CST=1, 19 **g** vectors
IDIME(3)=6, CST=1, 37 **g** vectors
IDIME(4)=1, CST=2, 3 **g** vectors
IDIME(5)=5, CST=2, 21 **g** vectors
IDIME(6)=9, CST=2, 42 **g** vectors

NNNC(1)=0

1)CS 11 33 (1s,1s; 1 **g**)

number of elements (ne)=1

NNNC(2)=NNNC(1)+ne=1

NNNC(3)=NNNC(2)+ne=2

2)CS 21 43 (1s,4×2p; 19 **g**)

ne=1×4×19×2=152

(2 is because of the weight of the CS, see *GORDSH*: the CS [(2 1),(4 3)] generates the couples [(1 2),(3 4)] by hermiticity)

NNNC(4)=NNNC(3)+ne=154

NNNC(5)=NNNC(4)+ne=306

3)CS 22 44 (4×2p,4×2p; 37 **g**)

ne=4×4×37=592

NNNC(6)=NNNC(5)+ne=898

NNNC(7)=NNNC(6)+ne=1490

4)....

3.7.3 The lower half Foch matrix

The lower half Fock matrix needs to be stored: it is obtained from the irreducible one by symmetry. The reason why only half matrix is considered is because we want to avoid as more as possible a large amount of memory: we know that the other half elements can be generated by hermiticity any time we need them. So, why the whole density matrix is stored? The density matrix is used to evaluate all the bielectronic integrals, so the generation of the transposed elements would be repeated a large number of times, with a large waste of time.

The first position of a couple (λ_i,λ_j), with $i > j$, in the Fock matrix, is given by the **NNNC2** vector. The vectors obtained for graphite with two TOLINTEG settings is here shown (the last row lists the lower half matrix couples in the symmetry order).

```
 FIRST POSITION OF THE (L1,L2) COUPLE IN F(G)
NNNC2    0  1   2  30  58  248 438      438  486  534    (TOLINTEG 5 5 5 5 12)
         0  1   2  30  58  248 438      441  489  537    (TOLINTEG 6 6 6 6 12)
(L1 L2) 11 33  21  43  22   44  31(13)   32   41   42(24)
```

Let's see, now, how to calculate the number of the Fock matrix elements and to obtain NNNC2 (with TOLINTEG 5 5 5 5 12).

IDIMF(1 2 4 0 3 4), see section 3.7.1

The corresponding total number of **g** vectors for each couple set is obtained, as for NNNC, by NQGSHG array, in the *GV* subroutine (see table 3.14):

IDIMF(1)=1, CST=1, 1 **g** vector

IDIMF(2)=2, CST=1, 7 **g** vectors

IDIMF(3)=4, CST=1, 19 **g** vectors

IDIMF(4)=0, CST=2, 0 **g** vectors

IDIMF(5)=3, CST=2, 12 **g** vectors

IDIMF(6)=5, CST=2, 18 **g** vectors

NNNC2(1)=0

1)CS 11 33 (1s,1s; 1 **g**)

number of elements (ne)=1

NNNC2(2)=NNNC2(1)+ne=1

NNNC2(3)=NNNC2(2)+ne=2

2)CS 21 43 (1s,4×2p; 7 **g**)

ne=1×4×7=28

NNNC2(4)=NNNC2(3)+ne=30

NNNC2(5)=NNNC2(4)+ne=58

3)CS 22 44 (4×2p,4×2p; 19 **g**). In this case we are considering a diagonal block, so only half the block needs to be stored. The number of the lower half matrix elements, for an n×n matrix, is (n×n+n)/2.

ne=[(4×4×+4)/2]×19=190

NNNC2(6)=NNNC2(5)+ne=248

NNNC2(7)=NNNC2(6)+ne=438

4)CS 31 13 (1s,1s; 0 **g**). Only half the set is considered, because the second block is in the upper part of the matrix. However, in the Fock matrix, this set is disregarded because no **g** vector is taken into account.

ne=0

NNNC2(8)=NNNC2(7)+ne=438

5)CS 32 41 (1s,4×2p; 12 **g**)

ne=1×4×12=48

NNNC2(9)=NNNC2(8)+ne=486

NNNC2(10)=NNNC2(9)+ne=534

6)CS 42 24 (4×2p,4×2p; 18 **g**) Even in this case, as for the 31 13 set, only half the set is considered, because the second block is in the upper part of the matrix. ne=4×4×12=288

NNNC2(10)+ne=822=NFGT

3.7.4

Once the number of elements in all matrices have been established, we need to know how is a single element of the matrices calculated. As regards the irreducible Fock matrix, each monoelectronic integral block, say $M^{\mathbf{g}_{irr}}_{\lambda_i\lambda_j}$, is put

on the $F^{\mathbf{g}_{irr}}$ vector starting from a position (**NSTATG** vector) defined in the *GMFCAL* subroutine; each bielectronic integral (say $B^{\mathbf{g}_{irr}\mathbf{g'}}_{\lambda_i\lambda_j\lambda_k\lambda_m}$) is first saturated with $P^{\mathbf{g'}}_{\lambda_k\lambda_m}$ (the starting point for (λ_k,λ_m,**g'**) must be calculated) and then inserted in $F^{\mathbf{g}_{irr}}_{\lambda_i\lambda_j}$ (see equations 1.29 1.31). The irreducible density matrix elements are built as shown in equations 1.21 and 1.22.

We also need to explore two additional structures:

- when the integrals are calculated, how are they selected and stored? (see *MONMAD* subroutine, explained in the *GILDA1* section)

- when going from the $F^{\mathbf{g}}_{\mu\nu}$ vector to the $F_{\mu\nu}(\mathbf{k})$ square matrix, how labels and positions are recovered? (*TRASFO* subroutine)

The *TRASFO* subroutine perform the Fourier Transform on the Fock matrix elements to obtain the F(**k**) matrices, for a given number of **k** points, and built the square Fock matrices in the reciprocal space (the lower half) that must then be diagonalized in the SCF cycle.

Two other arrays are evaluated and stored in the *GMFCAL* subroutine: IROF and JPOINT. We have already dealt with the **JPOINT** vector in the *GORDSH* section; it is calculated by the *GMFCAL* subroutine, and indicates, for a given couple, the corresponding couple set. The **IROF** array is here recalculated and stored, and has a different meaning with respect to the IROF vector calculated in *GORDSH* (and not stored). It has the size of the number of couple set, and indicates the starting irreducible **g** vector for a couple set in the NNGI sequence (see *GILDA1*, table 3.19). *INZVLB(MVLA)=INZVLA(IROF(MVLA)+1)— aggiungere

NNGI	1	2	8	9	14	20	21	32	0	1	2	5	9	13	17	21	23	27
NGIF									8									9
IROF	0	0	0	9	9	9												
CS(CST)	1(1)	2(1)	3(1)	4(2)	5(2)	6(2)												

Table 3.21. IROF vector for graphite. CS is the label of the couple set (from 1 to 6) and CST is the label of the couple set type to which the CS belongs. Where do the irreducible lattice vectors start, for example, for the couple set number 3? CS=3, CST=1, IROF(CS)=IROF(3)=0: they start in the 0+1=1-st position in NNGI array and end in the NGIF(CS)=NGIF(1)=8-th position. So they are 1, 2, 8, 9, 14, 20, 21, 32.

Chapter 4

Bielectronic integrals in frequency calculation

One of the crucial points of a SCF calculation is the evaluation of bielectronic integrals, having the form

$$I_{\mu\nu,\sigma\rho} =< \chi_\mu\chi_\nu|I|\chi_\sigma\chi_\rho > \tag{4.1}$$

where χ_μ,χ_ν,χ_σ,χ_ρ are the atomic orbitals, that will be also indicate with μ, ν, σ and ω, and I a general bielectronic operator.

The calculation cost of such terms is often heavy, and for large systems they are so many that they need to be calculated at each SCF step and cannot be stored (direct calculation). When frequencies are computed, for example, a SCF calculation is performed for each atomic displacement (3N, where N is the number of irreducible atoms), in which the symmetry is reduced. Consider, for example the boehmite ($Cmc2_1$ space group), with 4 irreducible atoms per cell:
1 central point + 4 atoms $\times$3 displacements= 13 SCF calculations are required.

ATOM	STEP	ENERGY (AU)	N.CYC	DE	SYM
CENTRAL POINT		-7.873866328166E+02	20	0.0000E+00	4
1 AL DX	1 * 1.8897E-03	-7.873866326348E+02	5	1.8186E-07	1
1 AL DY	1 * 1.8897E-03	-7.873866329734E+02	10	-1.5680E-07	2
1 AL DZ	1 * 1.8897E-03	-7.873866324539E+02	7	3.6277E-07	2
3 O DX	1 * 1.8897E-03	-7.873866326816E+02	6	1.3504E-07	1
3 O DY	1 * 1.8897E-03	-7.873866325019E+02	10	3.1478E-07	2
3 O DZ	1 * 1.8897E-03	-7.873866322880E+02	6	5.2868E-07	2
5 O DX	1 * 1.8897E-03	-7.873866326780E+02	5	1.3864E-07	1
5 O DY	1 * 1.8897E-03	-7.873866318934E+02	6	9.2326E-07	2
5 O DZ	1 * 1.8897E-03	-7.873866326555E+02	8	1.6113E-07	2
7 H DX	1 * 1.8897E-03	-7.873866327973E+02	5	1.9365E-08	1

```
7 H  DY  1 *  1.8897E-03   -7.873866327950E+02     8     2.1614E-08   2
7 H  DZ  1 *  1.8897E-03   -7.873866322569E+02     8     5.5976E-07   2
```

Bielectronic integrals are then calculated 13×(number of cycles) times and the symmetry is reduced from 4 to 2 or 1. Each single SCF calculation takes about 40 minutes (parallel calculation with four processors), and so 13 SCF calculations will take about 9 hours.

A simple analysis shows, however, that many of the integrals (and gradients) of the symmetry reduced system (after the atom displacement) remain unchanged with respect to the high symmetry of the central point and could be obtained by rotation. Only the integrals $< \chi_\mu \chi_\nu | I | \chi_\sigma \chi_\rho >$ with μ or ν or λ or ρ belonging to the displaced atom must be calculated at each SCF.

In order to exploit this "invariance", we must know how integrals are calculated (in blocks or one by one) and the effect of the symmetry operators on them. This is what we are going to discuss in the following sections of this chapter.

4.1 Transformation properties of mono and bielectronic integrals

In this section we want to discuss the transformation properties of matrix elements such as

$$M_{\mu\nu} = \int \chi_\mu \widehat{M} \chi_\nu d\tau \tag{4.2}$$

where $\widehat{M}$ is any of the monoelectronic operators, and

$$I_{\mu\nu,\sigma\omega} = \int_D \chi_\mu(\mathbf{r_1}) \chi_\nu(\mathbf{r_1}) \widehat{I}(\mathbf{r_1},\mathbf{r_2}) \chi_\sigma(\mathbf{r_2}) \chi_\omega(\mathbf{r_2}) d\mathbf{r_1} d\mathbf{r_2} \tag{4.3}$$

where $\widehat{I}(\mathbf{r_1},\mathbf{r_2})$ in our case is $r_{1,2}^{-1} = |\mathbf{r_1} - \mathbf{r_2}|^{-1}$.

We begin remembering that the χ AOs are bases for the irreducible representations of the rotation group, with size 1 (s), 4 (sp), 3 (p), 5 (d), 7 (f) and so on.

Let us consider two functions that transform according to the {i} and {j} irreducible representations, whose corresponding matrices are R^i and R^j: the product of such functions will transform according to the $R^i \otimes R^j$ direct product. In the present case, i and j are the labels of the shell type (or irreducible representation) to which the functions belong.

Then, we first represent the symmetry operators in the basis of the shell types (in CRYSTAL the *GSYM11* subroutine generates matrices of 1×1, 4×4 and 3×3 size for s, sp and p functions and the *GSYM33* subroutine matrices of 5×5 and 7×7 size for d and f functions) and then the direct product between the two matrices is calculated (in the *GSYM22* routine).

Suppose we have a C_3 operator along the z Cartesian axis acting on a s and a sp shells. Their corresponding rotation matrices are, respectively:

$$C_3^s = 1 \tag{4.4}$$
$$C_3^{sp} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -\frac{1}{2} & \frac{\sqrt{3}}{2} & 0 \\ 0 & -\frac{\sqrt{3}}{2} & -\frac{1}{2} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Consider, now, the symmetry operator $\widehat{R} = R^\lambda$ (with λ being the shell type label) acting on the λ shell.

$$R^\lambda \lambda = \lambda' \tag{4.5}$$

Each orbital belonging to the λ' shell can then be obtained by a linear combination of the orbitals of the λ shell. This means that, for example, when $\lambda = p$, we get:

$$R^p p_x = p'_x = R^p_{11} p_x + R^p_{12} p_y + R^p_{13} p_z = \sum_{i=x,y,z} R^p_{1i} p_i \tag{4.6}$$
$$R^p p_y = p'_y = \sum_{i=x,y,z} R^p_{2i} p_i$$
$$R^p p_z = p'_z = \sum_{i=x,y,z} R^p_{3i} p_i$$

where R^p are the elements of the $\widehat{R}$ representation matrix in the {p} basis. A similar scheme can be applied for a general $\lambda_1\lambda_2$ couple of shells, centered on atoms 1 and 2 respectively (note that atoms 1 and 2 can be the same atom).

$$(R^{\lambda_1} \otimes R^{\lambda_2})(\lambda_1\lambda_2) = (\lambda_1\lambda_2)' = \sum_{\lambda_1,\lambda_2} R^{\lambda_1,\lambda_2}_{\lambda'_1\lambda_1,\lambda'_2\lambda_2}(\lambda_1\lambda_2) \tag{4.7}$$

Along the same line, the $M_{\mu\nu}$ element in equation 4.2 will be transformed into the $M_{\mu'\nu'}$ element, and the $\widehat{R}$ operator can be written as the direct product of the symmetry operator matrices corresponding to the shell types:

$$M_{\mu'\nu'} = \widehat{R} M_{\mu\nu} = \sum_{\mu\nu} (R^{\lambda_1}_{\mu'\mu} R^{\lambda_2}_{\nu'\nu}) M_{\mu\nu} \tag{4.8}$$

Let us consider, as an example, λ_1 =p and λ_2 =d, with $\{\mu_1,\mu_2,\mu_3\}\epsilon\lambda_1$ and $\{\nu_1,\nu_2,\nu_3,\nu_4,\nu_5\}\epsilon\lambda_2$. The direct product of operators in the basis of p and d orbitals generates a $3\cdot 5\times 3\cdot 5 = 15\times 15$ matrix, so that one can write

$$\underset{15}{|M'>} = \underset{15\times 15}{(R^p\otimes R^d)}\underset{15}{|M>}$$

where the $|M>$ vectors contain the 15 possible $\mu\nu$ matrix elements.

Equation 4.8 can be written in a short way as

$$M_i = \sum_j T_{ij} M_j \tag{4.9}$$

where:

a. A vector notation has been used for the couple of indices $\mu\nu$; for example suppose again μ and ν belonging to p and d shell respectively. The couple $\mu\nu$ span then 3×5 values:

	d_{z^2}	d_{xz}	d_{yz}	$d_{x^2-y^2}$	d_{xy}
p_x	1	4	7	10	13
p_y	2	5	8	11	14
p_z	3	6	9	12	15

(4.10)

that can be vectorialised in the order indicated in the above matrix.

b. The T_{ij} matrix is the direct product $R^{\lambda_1}\otimes R^{\lambda_2}$.

It happens that most of the elements of the R matrices, and as a consequence of the T matrices, are null, so that they can be skipped in the summation in equation 4.8. This means that the summation in equation 4.9 can be substituted by a shorter summation

$$M_i = \sum_{j'} \mathcal{T}[i,j']M[j'] \tag{4.11}$$

where the notation indicates that we have a list $\mathcal{T}$ of T non-null terms, and for each of them we need to know the associated i and j indices (where taking the element in $M[j']$ and where pointing it in M_i). The compact list and the related pointing vectors of the non-null elements (ZO34, NO34, NOM34, KO34, NOVF34 and LOVF34) are built in the $GSYM22$ subroutine.

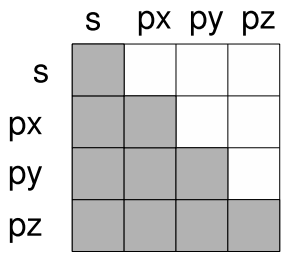

Figure 4.1. Matrix for $\lambda_1 = \lambda_2$ =sp. Each square represent a matrix element $\mu\nu$, only the filled squares are stored.

When $\lambda_1 = \lambda_2$ we may be interested in building either the full block, or only half a matrix block; for example the elements with $\mu \geq \nu$, as shown in figure 4.1.

In the full block case NOVF34 is used as a pointer; for only half a block the list is even shorter and it is stored in the same vector, ZO34, with pointer LOVF34.

Let us consider now a general bielectronic integral. We have:

$$I_{\mu'\nu',\sigma'\omega'} = \sum_{\mu\nu\sigma\omega} \underbrace{R^{\lambda_A}_{\mu'\mu} R^{\lambda_B}_{\nu'\nu}}_{\mathcal{T}_{L'L}} \underbrace{R^{\lambda_C}_{\sigma'\sigma} R^{\lambda_D}_{\omega'\omega}}_{\mathcal{T}_{N'N}} I_{\mu\nu,\sigma\omega} \tag{4.12}$$

$$I_{L',N'} = \sum_{LN} \mathcal{T}_{L'L} \mathcal{T}_{N'N} I_{LN} \tag{4.13}$$

We must underline that if the atoms on which μ and ν are centered are not at the origin, equations 4.2-4.13 remain valid, but integrals will refer to AOs centered on atoms A_μ,B_ν,C_σ,D_ω, and to the atoms to which $A_{\mu'}$,$B_{\nu'}$,$C_{\sigma'}$,$D_{\omega'}$ arrive by the effect of the $\widehat{R}$ operator: $\widehat{R}A_\mu = A_{\mu'}$.

Now, we are ready to discuss how the previous equations are implemented in CRYSTAL.

4.1.1 The *GSYM11* subroutine

The *GSYM11* subroutine is called by the *INT_SCREEN* subroutine.
It prepares intermediate quantities for the *GSYMM22* routine, that build information for rotating matrix elements $M_{\mu\nu}$. *GSYM11* defines the symmetry operator matrices R for all shell types and all operators, such that the total number of matrices is given by the total number of symmetry operators (MVF=inf(2)) times the total number of shell types (MXTP+1, where

MXTP=inf(175) is the label of the shell type with the maximum number of AOs).

These matrices are stored in compact form in the **TTO** vector (null terms are disregarded). Three pointers, **MMO**, **MMOM** and **MINZ** are defined: they point, respectively, the row and the column indices of non-null elements and the starting point of the point group operators.

Two arrays, defined in the *SHELL_INFO* module, are used: SHNAO, giving for each shell type the corresponding number of atomic orbitals, and SHTXT, where the shell "text" labels are listed. Their position indices run from 0 to 7 and correspond to the CRYSTAL input basis set labels defining the type of the shell (only shell types from 0 to 4 can be used, see table 4.1). Note that if the basis set contains, for example, s and d shells, only the s and d matrices are stored.

I	0	1	2	3	4	5	6	7
SHNAO(I)	1	4	3	5	7	9	11	13
SHTXT(I)	S	SP	P	D	F	G	H	I

Table 4.1. SHNAO and SHTXT vectors.

The first do-loop chain of this subroutine runs on the shell types and on the operators, in order to build the representation matrices (see equations 4.4). The *GSYMM33* subroutine is called to build matrices for d and f shells.

Matrices are then stored in a single vector, TTO0, by columns, as shown in equation 4.10, and the operators appear in their sequential order (from 1 to MVF). Such a vector, however, could contain a large number of useless null terms. Non-null terms in TTO0 are then copied to the compact TTO array (size: LIMROT1=inf(138)) in the second do-loop chain running again on the shell types, and for each shell type on the operators. Three pointers are then defined: MMO, MMOM and MINZ.

MMO and MMOM have the same size as TTO and give for each TTO element the corresponding row and column indices in the original representation matrices, respectively. MINZ is a MVF×(MXTP+1) array and gives the starting point for each operator and for each shell type in the TTO global array.

Example

Let us discuss the example of ammonia, with one s shell on hydrogen atoms and one s and one sp shells on nitrogen (MVF=6, MXTP=1). Table 4.2, whose elements can be printed by setting LPRINT(9) in the CRYSTAL input file, shows the representation matrices of the six operators of ammonia in the basis of the s and sp shells. The number of non-null elements in table 4.2, inf(138), is 38; the size of the TTO vector, and so of the MMO and MMOM arrays, is then 38, whereas the size of the MINZ vector is $6 \times 2 = 12$. The first sixteen elements of these four vectors are shown in tab.4.3.

MVF	SHELL TYPE 0	SHELL TYPE 1			
1	1	1	0	0	0
		0	1	0	0
		0	0	1	0
		0	0	0	1
2	1	1	0	0	0
		0	$-\frac{1}{2}$	$-\frac{\sqrt{3}}{2}$	0
		0	$\frac{\sqrt{3}}{2}$	$-\frac{1}{2}$	0
		0	0	0	1
3	1	1	0	0	0
		0	$-\frac{1}{2}$	$\frac{\sqrt{3}}{2}$	0
		0	$-\frac{\sqrt{3}}{2}$	$-\frac{1}{2}$	0
		0	0	0	1
4	1	1	0	0	0
		0	$\frac{1}{2}$	$\frac{\sqrt{3}}{2}$	0
		0	$\frac{\sqrt{3}}{2}$	$-\frac{1}{2}$	0
		0	0	0	1
5	1	1	0	0	0
		0	-1	0	0
		0	0	1	0
		0	0	0	1
6	1	1	0	0	0
		0	$\frac{1}{2}$	$-\frac{\sqrt{3}}{2}$	0
		0	$-\frac{\sqrt{3}}{2}$	$-\frac{1}{2}$	0
		0	0	0	1

Table 4.2. The C_{3v} group representation matrices in the s (SHELL TYPE 0) and sp (SHELL TYPE 1) basis.

4.1.2 GSYM22

After the *GSYM11*, the *INT_SCREEN* subroutine calls the *GSYM22* subroutine, that organises in a compact form direct products (**ZO34**) between two symmetry operator matrices corresponding to the shell types of each possible couple of shell types (see equations 4.8 and 4.9). Five pointers are defined:

MMO	1	1	1	1	1	1	1	2	3	4	1	2	3	2	3	4	..	..
MMOM	1	1	1	1	1	1	1	2	3	4	1	2	2	3	3	4	..	..
TTO	1	1	1	1	1	1	1	1	1	1	1	-0.5	0.866	-0.866	-0.5	1	..	..
MINZ	0	1	2	3	4	5	6				10						16	..

Table 4.3. The first few elements of the MMO, MMOM, TTO and MINZ vectors corresponding to the matrices given in table 4.2. Remember that MMO and MMOM contain the row and column indices respectively, and MINZ the starting point of each operator in the TTO sequence. The non-null matrix elements for SHELL TYPE 0 come first (the vertical line indicates where its data end), then SHELL TYPE 1 follows.

- **NO34**, row index of direct product elements;
- **NOM34**, column index of direct product elements;
- **KO34**, column index of elements in the transposed matrix;
- **NOVF34**, starting point of each operator in the ZO34 vector;
- **LOVF34**, starting point of each operator in the ZO34 vector when $\lambda_1 = \lambda_2$ (see figure 4.1 and the related discussion; in this case only half the matrix is stored).

The *GSYM22* subroutine is structured in three large do-loop chains. In the first one, the size of the quantities to be allocated (LIMROT2) is evaluated. The do-loops run on the first shell type label (λ_1 in the equations), on the second shell type label (λ_2, with $\lambda_2 \leq \lambda_1$) and on all the point group operators, and the number of products between elements of the TTO array are calculated (null terms have already been disregarded by *GSYM11*). Let us consider again the ammonia example.

- $\lambda_1 = 0$, $\lambda_2 = 0$: 6 elements (6 direct products between 1×1 matrices)
- $\lambda_1 = 1$, $\lambda_2 = 0$: for $\lambda = 1$ four out of 6 operators have 6 non-null elements and 2 have 4 non-null terms (see table 4.2). The total number of elements is then $6 \times 4 + 4 \times 2 = 32$.
- $\lambda_1 = 1$, $\lambda_2 = 1$: 4 direct product matrices with $6 \times 6 = 36$ and 2 with $4 \times 4 = 16$ non-null elements. The total number of elements is then LIMROTX$= 6 + 32 + 36 \times 4 + 16 \times 2 = 214$.

In these calculations we have just considered the cases with $\lambda_2 \leq \lambda_1$; we can, however, obtain the total number of elements in the following way:

LIMROT2=LIMROTX+LIMROTX-LIMROTY

Suppose we have three shell types: λ_1, λ_2 and λ_3. The number of elements of the blocks of the following couples are then calculated (LIMROTX):

(λ_1,λ_1); (λ_2,λ_1); (λ_2,λ_2); (λ_3,λ_1); (λ_3,λ_2); (λ_3,λ_3).

The number of transposed block element can be obtained by adding twice LIMROTX and subtracting the number of diagonal block elements. For the reason discussed in figure 4.1, only the number of elements in the upper-half diagonal block is subtracted (LIMROTY), so that also the half diagonal blocks can be stored in the calculated vectors.
For ammonia we get: LIMROT2=364, LIMROTY=64, LIMROTX=214.

In order to avoid useless operations, the calculations of matrices corresponding to shell types not present in the considered calculation is disregarded (IF condition in the λ_1 and λ_2 do-loops); for example we can use s, sp and d shells, so we do not need to calculate matrices involving p shells. Parameters from inf(106) to inf(113) refer to shells from s to i and are equal to 1 if the shell is present (the condition is satisfied) and null when the shell is not present (the corresponding cycle of the do-loop chain is not executed).

The quantities listed above are then allocated with LIMROT2 size. The second do-loop chain runs on λ_1, λ_2 (with $\lambda_2 \leq \lambda_1$), the operators (MVF) and, for each operator, on all the non-null elements of the operator. The direct product array is ZO34, and NO34 and NOM34 arrays contain the row and column indices respectively, by which each element of the ZO34 vector is identified. In order to be able to easily transpose the blocks contained in ZO34, the KO34 array is calculated: it indicates, for each element, its column index in the transposed matrix (and NOM34 is here used to identify the transposed row index). Table 4.4 shows the elements of these arrays for the ammonia case described previously for λ_1,λ_2=0,0 and for λ_1,λ_2=0,1 with the 4-th operator.

When $\lambda_1 \neq \lambda_2$ the matrices for λ_2,λ_1 (with $\lambda_2 > \lambda_1$) are generated by transposition of the matrices for λ_1,λ_2 and stored in the ZO34 vector.

A further pointer indicating the starting point of each operator in the ZO34 vector, NOVF34, is defined in the second do-loop chain; its size is the number of different couples of shell types (in the ammonia case it is 4: s,s;

s,sp; sp,s; sp,sp) times the number of the symmetry operator incremented by one,(MVF+1). The last element of NOVF34, obviously, is the total number of direct product elements, and it is stored also as inf(76) at the end of the second do-loop chain (in our example inf(76)=246). The NOVF34 array is shown in table 4.5 for the ammonia example; in order to understand how does it work, look at table 4.6.

λ_1,λ_2	0,0							0,1							
operator	1	2	3	4	5	6	..	4						..	..
i	1	1	1	1	1	1	..	17	18	19	20	21	22	..	..
NO34(i)	1	1	1	1	1	1	..	1	2	3	2	3	4	..	..
NOM34(i)	1	1	1	1	1	1	..	1	2	2	3	3	4	..	..
KO34(i)	1	1	1	1	1	1	..	1	2	3	2	3	4	..	..
ZO34(i)	1.0	1.0	1.0	1.0	1.0	1.0	..	1.0	0.5	0.866	0.866	-0.5	1.0	..	..

Table 4.4. Parts of NO34, NOM34, KO34 and ZO34 vectors for the ammonia example.

NOVF34(0:6,0,0)	0	1	2	3	4	5	6
NOVF34(0:6,0,1)	6	10	16	22	28	32	38
NOVF34(0:6,1,0)	38	42	48	54	60	64	70
NOVF34(0:6,1,1)	70	86	122	158	194	210	246

Table 4.5. The NOVF34 pointer for the ammonia example. The three indices of NOVF34 indicate, respectively, the operator (from 0 to MVF+1), the shell type λ_1 and the shell type λ_2.

λ_1,λ_2	0,0						0,1						...	
Sym. op.	1	2	3	4	5	6	1	2	3	4	5	6	...	...
NOVF34	0	1	2	3	4	5	6	10	16	22	28	32	38	...

Table 4.6. Elements of the first two rows of NOVF34 array in table 4.5. For example: where does the first operator of the 0,0 couple start in the ZO34 vector? It starts in the position NOVF34(0,0,0)+1=1 and contains NOVF34(1,0,0)-NOVF34(0,0,0)=1 elements. And the first operator of the 0,1 couple ? It runs from the position NOVF34(0,0,1)+1=7 to the position NOVF34(1,0,1)=10 and the number of elements is NOVF34(1,0,1)-NOVF34(0,1,0)=4 And the sixth operator of the 0,1 couple? It runs from NOVF34(5,0,1)+1=33 to NOVF34(6,0,1)=38 and contains NOVF34(6,1,0)-NOVF34(5,0,1)=6 elements.

The third do-loop chain, running on the shell types λ_1, the operators and the elements of the operators, calculate the low half matrices of direct products

between matrices with $\lambda_1 = \lambda_2$, in the way discussed in the previous, and store the results in the ZO34 array, from the position inf(76)+1 to the position LIMROT2. The pointer LOVF34, similar to the NOVF34 array, points the starting point of each operator of each couple in the ZO34 array. It has only two indices, as $\lambda_1 = \lambda_2$, and it is shown in table 4.7 for the ammonia case.

LOVF34(0:6,0)	246	247	248	249	250	251	252
LOVF34(0:6,1)	252	262	285	308	331	341	364

Table 4.7. The LOVF34 pointer for the ammonia example. The two indices indicate, respectively, the operator (from 0 to MVF) and the shell type λ_1 ($=\lambda_2$).

4.2 Preliminary subroutines

Before entering more into details of the bielectronic integral subroutines, it is useful to analyse the structure of a few other subroutines, calculating preliminar quantities that will be used in the bielectronic calculation "core". We begin with a few comments about gaussian functions, remembering that atomic orbitals are expressed as contractions of gaussian functions. In particular, we are interested here in evaluating the product of s-type gaussian functions, as each shell (of any type: s, sp, p, d and f) is represented by an s-gaussian (the adjoined gaussian) that is used for the screening and selection

4.2.1 Gaussian functions and overlap

The definite integral of a x-only gaussian function is

$$\int_{-\infty}^{\infty} e^{-\gamma x^2} dx = \left(\frac{\pi}{\gamma}\right)^{\frac{1}{2}} \tag{4.14}$$

In the case of a 3D gaussian, we get

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-\gamma(x^2+y^2+z^2)} dxdydz = \tag{4.15}$$

$$= \int_{-\infty}^{\infty} e^{-\gamma x^2} dx \int_{-\infty}^{\infty} e^{-\gamma y^2} dy \int_{-\infty}^{\infty} e^{-\gamma z^2} dz = \left(\frac{\pi}{\gamma}\right)^{\frac{3}{2}}$$

A normalized s-type gaussian function g_a, centered at s_a, must be such that

$$\int_{-\infty}^{\infty} g_a g_a d\tau = N^2 \int e^{-2\alpha(\mathbf{r}-\mathbf{s}_a)^2} d\mathbf{r} = N^2 \left(\frac{\pi}{2\alpha}\right)^{\frac{3}{2}} \tag{4.16}$$

and the normal factor N for each gaussian is $N = \left(\frac{2\alpha}{\pi}\right)^{\frac{3}{4}}$. As a consequence, g_a takes the form:

$$g_a(\alpha,\mathbf{r}-\mathbf{s_a}) = \left(\frac{2\alpha}{\pi}\right)^{\frac{3}{4}} \cdot e^{-\alpha(\mathbf{r}-\mathbf{s_a})^2} \tag{4.17}$$

The product of two gaussian functions, g_a and g_b, centered in $\mathbf{s_a}$ and $\mathbf{s_b}$, respectively, can be expressed as follows:

$$g_a(\alpha,\mathbf{r}-\mathbf{s_a}) \cdot g_b(\beta,\mathbf{r}-\mathbf{s_b}) = \left(\frac{2}{\pi}\right)^{\frac{3}{2}} (\alpha\beta)^{\frac{3}{4}} e^{-\frac{\alpha\beta}{\alpha+\beta}(\mathbf{s_a}-\mathbf{s_b})^2} e^{-(\alpha+\beta)(\mathbf{r}-\mathbf{s_{ab}})^2} \tag{4.18}$$

with $\mathbf{s_{ab}}$ being

$$\mathbf{s} = \frac{\alpha \mathbf{s_a} + \beta \mathbf{s_b}}{\alpha + \beta} \tag{4.19}$$

Let us integrate, now, eq.4.18

$$\rho_{ab} = \left(\frac{2}{\pi}\right)^{\frac{3}{2}} (\alpha\beta)^{\frac{3}{4}} e^{-\frac{\alpha\beta}{\alpha+\beta}(\mathbf{s_a}-\mathbf{s_b})^2} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-(\alpha+\beta)(\mathbf{r}-\mathbf{s})^2} d\mathbf{r} = \tag{4.20}$$

$$= \left(\frac{2}{\pi}\right)^{\frac{3}{2}} (\alpha\beta)^{\frac{3}{4}} e^{-\frac{\alpha\beta}{\alpha+\beta}(\mathbf{s_a}-\mathbf{s_b})^2} \left(\frac{\pi}{\alpha+\beta}\right)^{\frac{3}{2}} = \left(\frac{4\alpha\beta}{(\alpha+\beta)^2}\right)^{\frac{3}{4}} e^{-\frac{\alpha\beta}{\alpha+\beta}(\mathbf{s_a}-\mathbf{s_b})^2}$$

In the calculation of the bielectronic integrals, many different relative $\mathbf{s_a}$ and $\mathbf{s_b}$ positions are explored for the first couple, as $\mathbf{s_a}$ and $\mathbf{s_b}$ (corresponding to the atomic positions on which the shells λ_a and λ_b are centered) are separated by different $\mathbf{g}$ translation vectors. We can consider, then, the $\mathbf{s_a}$ and $\mathbf{s_b}$ position in the zero cell and add to $\mathbf{s_b}$ the $\mathbf{g}$ vectors, such that:

$$\rho_{ab} = \left(\frac{4\alpha\beta}{(\alpha+\beta)^2}\right)^{\frac{3}{4}} e^{-\frac{\alpha\beta}{\alpha+\beta}(\mathbf{s_a}-\mathbf{s_b}-\mathbf{g})^2} \tag{4.21}$$

$$\mathbf{s_{ab}} = \frac{\alpha \mathbf{s_a} + \beta \mathbf{s_b} + \mathbf{g}}{\alpha + \beta}$$

The aim is to compare a given tolerance (say T_x) with the overlap $\rho^{\mathbf{g}}_{\mu\nu}$ between two AO, namely μ and ν belonging to the λ_a and λ_b shells respectively, and select cases where $\rho^{\mathbf{g}}_{\mu\nu} \geq 10^{-T_x}$; in particular, the AO are organized into shells, and the selection of each $\mu\nu$ term is performed on the basis of the overlap between the two adjoined gaussians of the λ_a and λ_b shells. By taking the natural logarithm of both sides we obtain:

$$ln\rho^{\mathbf{g}}_{ab} \equiv S_{ab} \geq ln(10^{-T_x}) \equiv ln[(10^{-1})^{T_x}] \equiv T_x \cdot ln(0.1) \equiv -2.303 T_x \tag{4.22}$$

Almost all the values given in the CRYSTAL[4] input for the thresholds refer to the negative decimal logarithm of the thresholds, the exception being the ones stored as inf(41) and inf(43), that we will discuss in section 4.3.5, corresponding to the negative natural logarithm of the thresholds. The formula implemented in the code, then, is the one corresponding to the natural logarithm of the overlap:

$$S_{ab} = \frac{3}{4} ln\left(\frac{4\alpha\beta}{(\alpha+\beta)^2}\right) - \frac{\alpha\beta}{\alpha+\beta} \cdot \mathbf{D^2} \tag{4.23}$$

$$\mathbf{D^2} = |\mathbf{s}_a - \mathbf{s}_b - \mathbf{g}|^2$$

and it is calculated for each mother couple of the couple sets and for the first **g** vector of each star of the considered couple. As a consequence, given two gaussian functions with α and β exponents, the first term of eq.4.23 is a constant, whereas the second one depends on the **g** distance between the two functions.

For this reason, the first term of eq.4.23 will be calculated only once for each "$\alpha\beta$" couple and stored in the **SCOST** vector (size=number of couple sets, *SHELLD* subroutine), whereas D is calculated for each **g** star (see instruction 97 in tab.4.11, *SHELLXN* routine). The exponents α and β of the most diffuse gaussian functions of two contractions (the adjoined gaussians, see section 1.1.10) are contained in the **EXAD** vector, whose size is the total number of shells; it is built in the *GAUNOV* subroutine, called by *INPBAS*.

Let us consider, now, the second electron of a bielectronic integral, whose orbitals are $\sigma \in \lambda_c$ and $\omega \in \lambda_d$. The λ_c and λ_d shells are centered in $\mathbf{s_c}$ and $\mathbf{s_d}$ respectively, and their adjoined gaussians have γ and δ exponents. The same comments as for λ_a and λ_b hold, and ρ_{cd}, $\mathbf{s_{cd}}$ and S_{cd} can be obtained by the same operation shown in the previous.

In the following sections, we will be interested also in the logarithm of the overlap between two overlap distributions, renormalized to 1 and separated by **h**:

$$[\lambda_a,\lambda_b(\mathbf{g})],[\lambda_c(\mathbf{h}),\lambda_d(\mathbf{h}+\mathbf{g}')]$$

We can write, then, the $S_{ab,cd}$ term as

$$S_{ab,cd} = \frac{3}{4} ln \left(\frac{4(\alpha+\beta)(\gamma+\delta)}{(\alpha+\beta+\gamma+\delta)^2} \right) - \frac{(\alpha+\beta)(\gamma+\delta)}{\alpha+\beta+\gamma+\delta} \cdot \mathbf{D^2} \tag{4.24}$$

$$\mathbf{D^2} = |\mathbf{s_{ab}} - \mathbf{s_{cd}} - \mathbf{h}|^2$$

and the corresponding centroid as:

$$\mathbf{s_{ab,cd}} = \frac{\alpha\mathbf{s_a} + \beta\mathbf{s_b} + \gamma\mathbf{s_c} + \delta\mathbf{s_d}}{\alpha+\beta+\gamma+\delta} \tag{4.25}$$

4.2.2 The *SHELLD* subroutine

The *SHELLD* subroutine is called by *INT_SCREEN* after the *GSYM22* subroutine, in order to perform the overlap selection for the exchange integrals on the basis of the tolerance discussed in section 1.1.10 (T_3, T_4, T_5).

In chapter 3.1 we have shown how the selection of the $\lambda_1\lambda_2$ couples is performed on the basis of the most severe tolerance among T_1, T_3, T_4, T_5. Then the couples are ordered by symmetry into couple sets and the selection on the **g** vectors is performed on the basis of the overlap criteria for terms like $< \mu^0 \nu^{\mathbf{g}} >$.

Here we show how the sums are truncated in the case of the exchange contribution to the Fock matrix and the energy[4]:

$$F^{\mathbf{g}}_{\mu\nu} = -\frac{1}{2}\sum_{\sigma,\omega}\sum_{\mathbf{g}'}\overbrace{P^{\mathbf{g}'}_{\sigma\omega}}^{T_5}\sum_{\mathbf{h}} < \overbrace{\chi^0_\mu\ \chi^{\mathbf{h}}_\sigma}^{T_3} \mid \frac{1}{|\mathbf{r_1}-\mathbf{r_2}|} \mid \overbrace{\chi^{\mathbf{g}}_\nu\ \chi^{\mathbf{h}+\mathbf{g}'}_\omega}^{T_3} > \qquad (4.26)$$

$$E_x = \frac{1}{2}\sum_{\mu\nu}\sum_{\mathbf{g}}\overbrace{P^{\mathbf{g}}_{\mu\nu}}^{T_4} F^{\mathbf{g}}_{\mu\nu} \qquad (4.27)$$

The first lines of the subroutine define the tolerance by which the selection is performed (also T_1 and T_2, used later on for the Coulomb terms) are evaluated:

CCCFAJ=$-\frac{1}{2.303*ITOL(1)}$
TTTTT2=$-2.303ITOL(2) - 1.5 * ln(2)$
BCCFAK=$-2.303ITOL(3)$
CCCFAK=$0 - \frac{50}{BCCFAK)}$
T6=BCCFAK*1.5
T4=$-2.303ITOL(4)$
T5=$-2.303ITOL(5)$
(the first four quantities are stored in the *CLADDD_MODULE*)

A DO-loop on the couple sets (MVLA in the code) evaluates the total number of translation vectors giving a non negligible overlap between two shells (**g**, **g**′ and **h** translation vectors in eq.4.26). First, the constant term of eq.4.18 is calculated in logarithmic form and stored, for each couple set, in the **SCOST** vector.

The ICCS1 vector (section 3.3) attributes to the "mother" couple MVLA its position in the symmetry ordered list of couples; information about the starting point in the translation vector list NQGSHG (LA34V in section 3.5) and the total number of stars to be considered for each set (LA34X) is then recalled, and the distance $\mathbf{s}_b(x,y,z) - \mathbf{s}_a(x,y,z)$ in the unit cell calculated (X1,Y1,Z1

quantities in the code).

```
      DO MVLA=1,MVLAF
      NXC=NCF(MVLA)
      L1=LA3(NXC+1)
      L2=LA4(NXC+1)
      EX8=EXAD(L1)
      EX9=EXAD(L2)
      EX12=1._FLOAT/(EX8+EX9)
      EX34=EX8*EX9*EX12
      EX12=LOG(EX34*EX12*4._FLOAT)*0.75_FLOAT
      SCOST(MVLA)=EX12
      I1=0
      I2=0
      I3=0
      NGSH=ICCS1(NXC+1)
      IPPOP=LA34V(NGSH)
      NGSH=LA34X(NGSH)
      X1=XL(1,L2)-XL(1,L1)
      Y1=XL(2,L2)-XL(2,L1)
      Z1=XL(3,L2)-XL(3,L1)
```

Three IF-conditions, shown in the following, permit to calculate the number of stars to be considered for $\mathbf{g}$, $\mathbf{g}'$ and $\mathbf{h}$, (see eq.4.26), **IDMEXG(MVLA)**, **IDMEXL(MVLA)** and **IDMEX(MVLA)** respectively (I2, I3 and I1 in the code). In particular, the condition $S12 \geq -T_x$, with x being 4, 5 and 3 for $\mathbf{g}$, $\mathbf{g}'$ and $\mathbf{h}$ respectively, must be satisfied for a given star LSTAR (see eq.4.23)

```
703   IF(LSTAR.EQ.NGSH)GOTO 7023
      LSTAR=LSTAR+1
      NW=.FALSE.
      LUF=NQGSHG(NGSHG(IPPOP+LSTAR)+1)
      S12=EX12-EX34*
     *((XG(1,LUF)+X1)**2+(XG(2,LUF)+Y1)**2+(XG(3,LUF)+Z1)**2)
 IF(S12.GE.BCCFAK)THEN
      NW=.TRUE.
      I1=LSTAR
 ENDIF
 IF(S12.GE.T4)THEN
      NW=.TRUE.
      I2=LSTAR
 ENDIF
 IF(S12.GE.T5)THEN
      I3=LSTAR
      GOTO 703
 ENDIF
      IF(NW)GOTO 703
7023  IDMEXG(MVLA)=I2
      IDMEXL(MVLA)=I3
      IDMEX(MVLA)=I1
```

The NW logical variable is a trick that permits to decrease the number of operation when the number of stars giving a non-negligible overlap is smaller than the total number of stars for each couple.

The call to the *OLDNEW* routine is aimed to write the EXAD vector in the EXAOLD vector, in order to have the reference exponent available when it changes during the calculation (see the FIXINDEX keyword in ref.[4]).

In the calculation of the exchange terms we will have to evaluate overlaps as shown in eq.4.28. Due to the organization of the DO loops, we will know the sequential number of the **g**, **g'** and **h** vectors and their Bravais lattice vector indices (LG(1:3,I), see section 3.4); we will then need to know the sequential order of the sum vector $\mathbf{l} = \mathbf{h} + \mathbf{g}' - \mathbf{g}$, in order to get later on its cartesian components.

$$S^{\mathbf{l}}_{\nu\omega} =< \nu(\mathbf{r} - \mathbf{g})|\omega(\mathbf{r} - \mathbf{h} - \mathbf{g}') >=< \nu(\mathbf{r})|\omega(\mathbf{r} - \mathbf{l}) > \tag{4.28}$$

For this reason a matrix **KG123(I,J,K)** is built: for each set of three indices it produces the sequential order of the corresponding vector. How many positions must be allocated for KG123? The maximum LG index for the full set of vectors classified in the *GCALCO* subroutine along the three cartesian direction is calculated, and then multiplied by 3 (as in the worst case all three vectors can reach the maximum value for a given component, positive or negative) and then the I, J and K indices in **KG123(I,J,K)** can run from -3*KGMAX to +3*KGMAX (KGMAX1 is along x, KGMAX2 along y and KGMAX3 along z).

The quantities WEIGHT is set to 1 and **KPESO** to .TRUE. for all vectors. We will discuss the meaning of these vectors later on, for the selection of the coulomb bielectronic integrals. (SEE REF. SHELLXN quando finito spiegare Kpeso)

4.2.3 The *ALLOCATE_SCOST* subroutine

At the end of the *SHELLD* subroutine, *ALLOCATE_SCOST* is called in order to organize the calculated quantities according to shell criteria.

The first part of this subroutine, that is a DO-loop-chain over shells, couple sets and couples of each set, defines the quantity LOOP, that coresponds to the allocation size of three of the four vectors calculated in the subroutine:

IDMEXLIST, **IDMEXMIST** and **RDMEXLIST**. The fourth vector, **IDMEXBASE**, has inf(20)+1 size.

```
LOOP=0
DO L1=1,LAF
DO MVLU=1,MVLAF
IDM=IDMEX(MVLU)
IF(IDM.EQ.0)CYCLE
DO I=NCF(MVLU)+1,NCF(MVLU+1)
IF(LA3(I).EQ.L1)THEN
LOOP=LOOP+1
ELSE
IF(LA4(I).EQ.L1)LOOP=LOOP+1
ENDIF
ENDDO
ENDDO
ENDDO
```

We discuss now how these quantities are evaluated and what does they mean. At first, the LOOP counter is set to 0 again and the DO-loop-chain is repeated as in the first part.

```
LOOP=0
DO L1=1,LAF
IDMEXBASE(L1)=LOOP
DO MVLU=1,MVLAF
IDM=IDMEX(MVLU)
IF(IDM.EQ.0)CYCLE
RC=SCOST(MVLU)
DO I=NCF(MVLU)+1,NCF(MVLU+1)
LL3=LA3(I)
LL4=LA4(I)
IF(LL3.EQ.L1)THEN
LOOP=LOOP+1
IDMEXLIST(LOOP)=LL4
IDMEXMIST(LOOP)=IDM
RDMEXLIST(LOOP)=RC
ELSE
IF(LL4.EQ.L1)THEN
LOOP=LOOP+1
IDMEXLIST(LOOP)=LL3
IDMEXMIST(LOOP)=IDM
RDMEXLIST(LOOP)=RC
ENDIF
ENDIF
ENDDO
ENDDO
ENDDO
IDMEXBASE(LAF+1)=LOOP
```

IDMEXBASE is a vector that associates to each shell the number of shell couples in which it is contained. For example, given a shell L1, the number of couples in which it is contained is IDMEXBASE(L1+1)-IDMEXBASE(L1).

For a given shell L1, **IDMEXLIST** provides the "other" shell of the couples containing L1, **IDMEXMIST** indicates, for one of these couples, the number of **g** stars to be considerated and **RDMEXLIST** the constant of the shell overlap SCOST, see eq.4.20 in section 4.2.2. **IDMEXBASE**, then, points on the other three vector the starting point of each shell.

Consider, for example, the case of a 2D-graphite layer (4 shells, TOLINTEG 5 5 5 5 12): the four vectors are shown in table 4.8. We just remember the couple sets (see section 3.3):

MVLA		
1	11	33
2	21	43
3	22	44
4	13	31
5	32	14
6	42	24

L1	1			2					3			4					
IDMEXBASE	0			3					8			11					16
IDMEXLIST	1	2	4	1	2	3	4	4	3	4	2	3	4	1	2	2	
IDMEXMIST	1	2	3	2	4	3	4	4	1	2	3	2	4	3	4	4	
RDMEXLIST	0	-1.1	-1.1	-1.1	0	-1.1	0	0	0	-1.1	-1.1	-1.1	0	-1.1	0	0	

Table 4.8. The quantities calculated in the *ALLOCATE_SCOST* subroutine for the first two shells of a graphite layer. The first line lists the shells; IDMEXBASE, containing the number of couples containing the indicated shell, points the starting point of the four shells in the other three vectors. IDMEXLIST contains the second shell of the couple when the L1 is the first shell, and the first one when L1 is the second shell (for example, when L1=2, both 42 and 24 couples are considered, and the shell 4 appears twice in IDMXLIST). Consider, for example, L1=1: there are three couples containing this shell, they are 11, 21 and 14. IDMEXLIST([IDMEXBASE(L1)+1]:[IDMEXBASE(L1+1)-IDMEXBASE(L1)]) gives then 1,2 and 4. IDMEXMIST contains the number of **g** stars for each couple (note that couples belonging to the same set have the same value of IDMEXMIST, for example couples 32 and 14), and RDMEXLIST the term of the overlap independent of translation vectors (eq. 4.20). Note that the couple set 13,31 is not considered, as the number of **g** stars is null (overlap smaller than the tolerance).

4.2.4 The *ZFEASV* subroutine

This subroutine is called much later than the previous ones, in the "core" of the bielectronic integral calculation, the *SHELLXN* subroutine (or *SHELLX*

subroutine, we will see later on the differences). It is called when the L1 and L2 shells are already defined, and transfer data from general vectors (for all the couple sets) to vectors for the specific MVLA value (remember that MVLA runs from 1 to inf(56)). The calculated quantities, however, are strongly related to information given by the *SHELLD* and the *ALLOCATE_SCOST* subroutines.

The values of L1 and L2, the shells of the mother couple of a considered couple set, are required.

The main instructions of *ZFEASV* are shown in the following, and is repeated twice (for L1 and L2). Six vectors of inf(20) size are calculated for each L1L2 mother couples of the sets: **IDMEX1**, **IDMEX2**, **SCOST1**, **SCOST2**, **LA34V1** and **LA34V2**, where the numbers 1 and 2 means that they are related to L1 and L2. Note that the **IDMEX-** vectors have been set to zero in the *ALLOCATE_SCOST* subroutine.

```
DO K=IDMEXBASE(L1)+1,IDMEXBASE(L1+1)
I=IDMEXLIST(K)
IDMEX1(I)=IDMEXMIST(K)
SCOST1(I)=RDMEXLIST(K)
ENDDO
DO K=ICCT(L1)+1,ICCT(L1+1)
LA34V1(ILA12T(K))=LA34V(K)
ENDDO
```

Let us analyse the first DO-loop. It runs on the couples containing L1, and I is the other shell contained in the L1 couples. **IDMEX1** lists the **g** stars to be considered for all the couples containing L1. Its meaning is exactly the same discussed for the IDMEXMIST array in section 4.2.3, but whereas IDMEXMIST is calculated only once and contains information about all shells, **IDMEX1** refers only to the L1 shell, and it is refilled for each L1L2 couple at each MVLA loop. Similar considerations holds for the **IDMEX2** array. Why doing such an operation? It will be useful, for the calculation of the exchange integrals, having the quantities for each shell separated, in order to easily perform the "exchange" of the shells.

The **SCOST1** and **SCOST2** vectors lists the factors independent of the translation vectors of eq.4.20, as the RDMEXLIST one, but information for each shell is separated.

Consider the case of a graphite monolayer, and have a look to table 4.8, where the full vectors are shown. We take as an example the mother couple L1L2=32.

```
L1=3                                L2=2
K= 9,10,11                          K=4,5,6,7,8
I=3,4,2                             I=1,2,3,4,4,3
IDMEX1(3)=1                         IDMEX1(1)=2
IDMEX1(4)=2                         IDMEX1(2)=4
IDMEX1(2)=3                         IDMEX1(3)=3
IDMEX1(1)=0                         IDMEX1(4)=4
SCOST1(3)=0                         SCOST2(1)=-1.1
SCOST(4)=-1.1                       SCOST2(2)=0
SCOST(2)=-1.1                       SCOST2(3)=-1.1
SCOST(1)=0                          SCOST2(4)=0
```

Suppose we have the bielectronic integral $< L1L2|L3L4 >$ with L1=3, L2=2, L3=4, L4=2. The exchange integrals to be evaluated are $< L1L3|L2L4 >$ and $< L1L4|L2L4 >$. How many stars must be considerated, for example, for L1L3? IDMEX1(L3)=IDMEX1(4)=2. And for L1L4? IDMEX1(L4)=3. Which is the constant term of eq.4.20 for the couple L1L3? SCOST1(L3)=0

Given a shell L1 (or L2), **LA34V1** (**LA34V2**) gives the starting point in the NQGSHG list of translation vectors of all the mother couples containing L1 (or L2). These quantities are already contained in the LA34V vector calculated in the GV subroutine (see section 3.5, for ILA12T and ICCT see section 3.1), but information is here separated for faster access.

Consider again the couple L1L2=32.

```
L1=3                                L2=2
K=9,10,11,12                        K=5,6,7,8
ILA12T(K)=1,2,3,4                   ILA12T(K)=1,2,3,4
LA34V1(1)=86                        LA34V2(1)=0
LA34V1(2)=86                        LA34V2(2)=0
LA34V1(3)=0                         LA34V2(3)=129
LA34V1(4)=43                        LA34V2(4)=129
```

So, where does the **g** vector list start in NQGSHG for the couple 33? It starts in the LA34V1(3)+1=1 position.

4.2.5 The FIXINDEX strategy

During optimization or frequency calculations, sets of SCF calculations are performed at different geometries. As the selection of integrals, as shown before, is based on overlap criteria, it happens that the center of mono- and bielectronic integrals is varying by SCF to SCF, producing discontinuity on the energy surfaces that can be as large as 10^{-3}-10^{-4} hartree. This effect can be reduced by setting tolerances at higher values; however, the cost is very

high, the effect limited. A much better strategy is the FIXINDEX scheme (see the FIXINDEX keyword in ref.[4]). A set of SCF calculations are performed by using a reference geometry for the couple selection (for example, in frequency calculations, the selection is performed on the basis of the equilibrium geometry), whereas the calculation is performed with the "actual"geometry.

For these reasons, two geometries have to be stored: the reference geometry, stored as "OLD", and the actual geometry, stored as "NEW". The FIXINDEX option is set automatically by optimization and frequency calculations; in a optimization calculation, as the equivalent geometry ca be very different from the starting one, a refresh of the reference geometry is possible when $\Delta E >$XXXXXXXXXX.

The splitting between "OLD" and "NEW" coordinates is performed in the *OLDNEW* subroutine, called by the *SHELLD* subroutine (see section 4.2.2). The coordinates of atoms, shells and **g** vectors are stored into 3×N matrices, XA (N=inf(24), total number of atoms in the unit cell), XL (N=inf(20), total number of shells) and XG (N=inf(79), total number of **g** vectors) respectively.

The **XA** matrix is built in the *EQUPOS* subroutine (INT_SCREEN→ INPUT→ CRYSTA→ SYMMET→EQUPOS), the **XG** matrix in the *GCALCO* subroutine and the **XL** one is obtained from XA (the center of a shell of an atom coincide with the center of the atom) in the *INPBAS* subroutine (note that the *INT_SCREEN* subroutine, calling the *GCALCO* and the *EQUPOS* subroutines indirectly, is called at each SCF, and so at each optimization step or frequency point).

The reference geometry is then copied to **XOLD**, **XLOLD** and **XGOLD** matrices in the *OLDNEW* subroutine. These lasts matrices are shared with the other subroutines by the *BASOLD_MODULE*, XA and XL by the *BASATO_MODULE* and XG by the *GVECT_MODULE*. Obviously, in the first energy point calculation, the "OLD" coordinates are equal to the "NEW" coordinates.

4.2.6 The *VIC2* subroutine

The *VIC2* subroutine prepares some quantities necessary for the gaussian product and the bielectronic integrals calculation, and shares them by the *TRAVIC* module. This subroutine is called after *ZFEASV*, in the *SHELLXN* subroutine.

The structure, relatively simple, and some comments about the *VIC2* routine are shown in table 4.9.

X1NEW=XL(1,L1)	New X coordinates of the first shell
Y1NEW=XL(2,L1)	New Y coordinates of the first shell
Z1NEW=XL(3,L1)	New Z coordinates of the first shell
X2NEW=XL(1,L2)-X1NEW	New X distance between the shells of a couple
...	...
EX2NEW=EXAD(L2)	New exponent of the most diffuse gaussian of the L2 shell
EX1NEW=EXAD(L1)	New exponent of the most diffuse gaussian of the L1 shell
FA2NEW=EX2NEW/(EX1NEW+EX2NEW)	
X1OLD=XLOLD(1,L1)	Old X coordinates of the first shell
...	...
X2OLD=XLOLD(1,L2)-X1OLD	Old X distance between the shells of a couple
...	...
EX2OLD=EXAOLD(L2)	Old exponent of the most diffuse gaussian of the L2 shell
EX1OLD=EXAOLD(L1)	Old exponent of the most diffuse gaussian of the L1 shell
DDDDDD=EX1OLD+EX2OLD	$\alpha + \beta$
SIG12=1._FLOAT/DDDDDD	$1/(\alpha + \beta)$
SX12=SCOST(MVLA)	$0.75log(4\alpha\beta\frac{1}{(\alpha+\beta)^2})$
FA2OLD=EX2OLD*SIG12	$\beta/(\alpha + \beta)$
F12OLD=EX1OLD*FA2OLD	$\alpha\beta/(\alpha + \beta)$
DDDDDD=DDDDDD*0.5_FLOAT	$(\alpha + \beta)/2$
FATOLD=F12OLD*0.5_FLOAT	$\alpha\beta/2(\alpha + \beta)$
MVRSJ(1)=L1	First shell of the first electron in Coulomb terms
MVRSJ(2)=L2	Second shell of the first electron in Coulomb terms
MVRSK(1)=L1	First shell of the first electron in exchange terms
MVRSK(3)=L2	Second shell of the first electron in exchange term
L1PBAS=LAA(L1)	
JPRIMA=LAA(L1+1)	
NPRIMA=JPRIMA-1	
JPRIMA=JPRIMA-L1PBAS	Total number of gaussian functions in the L1 shell
...	...
JPRIMB=JPRIMB-L2PBAS	Total number of gaussian functions in the L2 shell
LAT1=LAT(L1)	0=s; 1=sp; 2=p; 3=d; 4=f
LAT2=LAT(L2)	0=s; 1=sp, 2=p; 3=d; 4=f
MU=LAT2+LAT2	
LOOP=LAT1*8+1	
ITYPK=LOOP+MU	
ITYPJ=ITYPK+MU	(LAT1*8+1)+4*LAT2
ID1=LATAO(L1)	Number of AO in the L1 shell
ID2=LATAO(L2)	Number of AO in the L2 shell
ID12=ID1*ID2	Size of the L1L2 block
MTYPEA=ITYPE(LAT1+1)-1	ITYPE=1, s; 2, sp; 2, p; 3, d; 4, f
MTYPEB=ITYPE(LAT2+1)	
NTYPEB=I4BASE(LAT2+1)	I4BASE=0, s; 5, sp; 10, p; 15, d; 20, f
NTYPEA=I4BASE(LAT1+1)	
LMAXP=MTYPEA+MTYPEB	$\ell_{L1}+\ell_{L2+1}$
MTYPEB=MTYPEB-1	
ITYPAB=NTYPEA+LAT2+1	
NMAXP=MIKY(LMAXP+1)	Number of elements of half a matrix of LMAXP+1 size
NDIMJ1=NMAXP*ID12	
LLOG12=ID12.NE.1	ID12=1 if both L1 and L2 are s shells
LOOP=1	
DO 3 MU=1,LMAXP	
DO 3 NU=1,MU	
DO 3 ITAU=1,NU	
PHAJ(LOOP)=PHAVRS(LMAXP-ITAU+1)	PHAVRS: logical array in the *CONGEN* subroutine, the odd elements are true, the even ones false
LOOP=LOOP+1	
3ENDDO	

Table 4.9. The *VIC2* subroutine; instructions that repeats for y and z directions and for both L1 and L2 are shown only once.

4.3 The *SHELLXN* subroutine

In this section we explore the macro structure of the *SHELLXN* subroutine, that is called by the *SCF* subroutine at each SCF cycle, in order to evaluate the selected coulomb and exchange integrals, and sum and store (or use, as in the case of direct calculations they are not stored) them as a single block to be multiplied by the same density matrix block .

Consider the bielectronc contribution to the Fock matrix:

$$F^{\mathbf{g}}_{\mu\nu}(biel.) = (I_C + I_X) = \qquad (4.29)$$
$$\sum_{\sigma,\omega,\mathbf{g}'} \left\{ \sum_{\mathbf{h}} < \mu^0 \nu^{\mathbf{g}} | \sigma^{\mathbf{h}} \omega^{\mathbf{h}+\mathbf{g}'} > -\frac{1}{2} \sum_{\mathbf{h}} < \mu^0 \sigma^{\mathbf{h}} | \nu^{\mathbf{g}} \omega^{\mathbf{h}+\mathbf{g}'} > \right\} P^{\mathbf{g}'}_{\sigma\omega}$$

where I_C and I_X are the coulomb and the exchange contribution, respectively. These two contributions are summed together in a quantity, stored as such or saturated with the density matrix.

The subroutine consists essentially in a chain of seven large DO-lops, three on the $\mu\nu^{\mathbf{g}}$ term of the bielectronic integral, and four on the $\sigma\omega^{\mathbf{g}'}$ term . The scheme is represented in fig.4.2, whereas the whole subroutine is shown in tables 4.10-4.17, where each instruction is labeled.

The description will frequently refer to quantities calculated in the first part of the CRYSTAL code; in particular the arrays discussed in chapter 3.1 will be used.

As we will see later on, when the implementation of the bielectronic integral calculation will be discussed, these seven DO-loops are often split into two sub-blocks, one containing instructions for coulomb term calculation and the other one for the exchange integral calculation.

For each cycle on the seven main do-loops (see scheme in figure 4.2), the *SHELLXN* subroutine calculates two blocks of coulomb and two blocks of exchange integrals.

For the coulomb terms consider the two sets of integrals:

$$\begin{aligned} a) \quad & < \mu\nu^{\mathbf{g}} | \sigma^{\mathbf{h}} \omega^{\mathbf{h}+\mathbf{g}'} > P^{\mathbf{h},\mathbf{h}+\mathbf{g}'}_{\sigma,\omega} \\ b) \quad & < \mu\nu^{\mathbf{g}} | \omega^{\mathbf{h}+\mathbf{g}'} \sigma^{\mathbf{h}} > P^{\mathbf{h}+\mathbf{g}',\mathbf{h}}_{\omega,\sigma} \end{aligned}$$

a) and b) integrals are actually the same, as they correspond simply to the change of the factors for the second electron. Note that the same exchange appears in the density matrix that saturates the integrals. These density

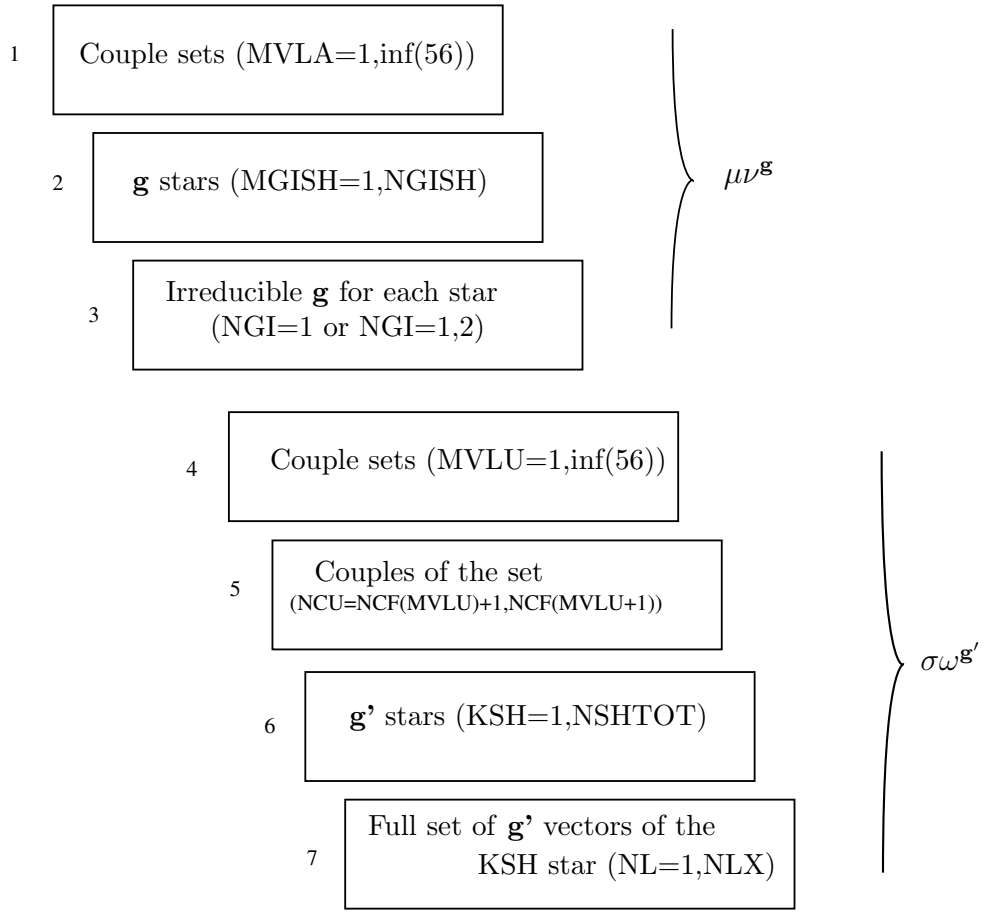

Figure 4.2. Schematic representation of the *SHELLXN* subroutine: the first three blocks run on the $\mu\nu^{\mathbf{g}}$ term of the bielectronic integral, blocks from 4 to 7 on the $\sigma\omega^{\mathbf{g}'}$ term. It should be noticed that the $\mathbf{h}$ summation in eq.4.26 does not appear in this scheme. It will be discussed later on.

matrix elements are never calculated as such; the one in a), in fact, by using translational invariance, becomes:

$$P^{\mathbf{h},\mathbf{h}+\mathbf{g}'}_{\sigma,\omega} = P^{0,\mathbf{g}'}_{\sigma,\omega} \equiv P^{\mathbf{g}'}_{\sigma,\omega}$$

As regards b), we have

$$P^{\mathbf{h}+\mathbf{g}',\mathbf{h}}_{\omega,\sigma} = P^{\mathbf{g}',0}_{\omega,\sigma} = \left[P^{0,-\mathbf{g}'}_{\omega,\sigma}\right] = P^{0,\mathbf{g}'}_{\sigma,\omega} \equiv P^{\mathbf{g}'}_{\sigma,\omega}$$

where in the first passage we used the translational invariance, in the second hermiticity. Note that when only the upper or the lower labels are interchanged, the "minus" signs appears.

Then the a) and b) bielectronic contribution coincide, that is, they are the same integral multiplied by the same density matrix. For this reason, in the code, only one set of integrals ($\lambda_3 \geq \lambda_4$, where $\sigma \in \lambda_3$ and $\omega \in \lambda_4$) is computed, because in this case the density matrix, built only for $\lambda_3 \geq \lambda_4$, is available .

It should be noticed that in eq. 4.29 all the selected integrals characterized by a different **h** are multiplied by the same density matrix. We can, then, saturate **h** before multipling by the density matrix.

The above analysis would permit to obtain the second set from the first one in the case of an infinite $\{\mathbf{h}\}$ set. However, this is not the case, as a different set is selected for a) and b) according to the overlap of λ_3 or λ_4 with the $\mu\nu^{\mathbf{g}}$ distribution.

We get then, for a given set of μ,ν,σ,ω,$\mathbf{g}$,$\mathbf{g}'$, the following term:

$$\left(\sum_{\mathbf{h}}^{\{\mathbf{h}_\sigma\}} < \mu^0 \nu^{\mathbf{g}} | \sigma^{\mathbf{h}} \omega^{\mathbf{h}+\mathbf{g}'} > \right) \cdot P_{\sigma\omega}^{0\mathbf{g}'} \tag{4.30}$$

where $\{\mathbf{h}_\sigma\}$ is the set of **h** vectors such that the λ_3 ($\sigma \in \lambda_3$) shell penetrates into the L1L2**g** ($\mu \in$L1, $\nu \in$L2) overlap distribution, on the basis of the T_2 tolerance (see section 1.1.10). The selection of the **h** vectors to be considered for the exact coulomb integrals calculation is performed in the *CLASSS* subroutine, and will be discussed in section 4.3.3.

For the second set of integrals, we have in general:

$$\left(\sum_{\mathbf{h}'}^{\{\mathbf{h}'_\omega\}} < \mu^0 \nu^{\mathbf{g}} | \omega^{\mathbf{h}'} \sigma^{\mathbf{h}'+\mathbf{g}''} > \right) \cdot P_{\omega\sigma}^{0\mathbf{g}''} \tag{4.31}$$

In order to have a set of integrals to be multiplied by the same density matrix, we set in eq.4.31 $\mathbf{g}'' = -\mathbf{g}'$, so that

$$P_{\omega\sigma}^{0\mathbf{g}''} = P_{\omega\sigma}^{0-\mathbf{g}'} = P_{\sigma\omega}^{0\mathbf{g}'} \equiv P_{\sigma\omega}^{\mathbf{g}'} \tag{4.32}$$

and then we get

$$\rightarrow \left(\sum_{\mathbf{h}'}^{\{\mathbf{h}'_\omega\}} < \mu^0 \nu^{\mathbf{g}} | \omega^{\mathbf{h}'} \sigma^{\mathbf{h}'-\mathbf{g}'} > \right) \cdot P_{\sigma\omega}^{\mathbf{g}'} \tag{4.33}$$

We can compare now, the set in eq.4.33 with the b) set: the two integrals coincide if **h'**=**h**+**g'**.

It is clear that the two set, $\{h'_\omega\}$ and $\{h_\sigma\}$, coincide only in the case of diagonal couples ($\sigma = \omega$). The question is then: how the $\{h'_\omega\}$ set can be

attributed to the couple $\sigma\omega^{\mathbf{g}'}$?
The same integral as eq.4.30 is obtained for the $\mathbf{h}'_i$ vectors of the $\{\mathbf{h}'_\omega\}$ set such that $\mathbf{h}'_i$-$\mathbf{g}'$=$\mathbf{h}_j$, where $\mathbf{h}_j \in\{\mathbf{h}_\sigma\}$; in such a case $\mathbf{h}_j$ can be attributed a WEIGHT=2. Otherwise, $\mathbf{h}'_i$-$\mathbf{g}'$ is added to the $\{\mathbf{h}_\sigma\}$ list. The $\{\mathbf{h}_\sigma\}$ set has then a weight 1 or 2 according to the above discussion.
We will come back to this point later on, in the fifth DO-loop of this subroutine (see section 4.3.9).

In the case of the exchange terms both the $< \mu\sigma^{\mathbf{h}}|\nu^{\mathbf{g}}\omega^{\mathbf{h}+\mathbf{g}'} >$ and the $< \mu\omega^{\mathbf{h}+\mathbf{g}'}|\nu^{\mathbf{g}}\sigma^{\mathbf{h}} >$ integrals must be calculated.

Even in this case they can be multiplied by the same density matrix. For the first integral we get:

$$P^{\mathbf{h},\mathbf{h}+\mathbf{g}'}_{\sigma,\omega} = P^{\mathbf{0},\mathbf{g}'}_{\sigma,\omega} \equiv P^{\mathbf{g}'}_{\sigma,\omega} \tag{4.34}$$

and for the second one:

$$P^{\mathbf{h}+\mathbf{g}',\mathbf{h}}_{\omega,\sigma} = P^{\mathbf{g}',\mathbf{0}}_{\omega,\sigma} = P^{-\mathbf{g}',\mathbf{0}}_{\sigma,\omega} = P^{\mathbf{0},\mathbf{g}'}_{\sigma,\omega} \equiv P^{\mathbf{g}'}_{\sigma,\omega} \tag{4.35}$$

The selection of translation vectors is performed on the basis of the three tolerances T_3, T_4 and T_5 discussed in section 1.1.10.

4.3.1 Packages for integral calculation

In CRYSTAL, two packages for the calculation of integrals are available: POPLE and ATMOL. The first one is more efficient, but limited to s and sp shells (LAT(λ)=0 and 1; LAT(λ) is an integer value, ranging from 0 to 4, associated to each shell in the *INPBAS* subroutine); ATMOL can treat s, sp, p d and f shells. Then, if a bielectronic integral contains one or more shells λ having LAT(λ)$\geq$2 the ATMOL package is used by default. However, for integrals among four shells of sp and s type, both the options can be used. By default, POPLE is chosen in these cases, but a different choice is possible, by using the INTGPACK keyword (see the CRYSTAL manual[4], which permits to set inf(26) according to the following (*READM2* subroutine):

```
INF(26)=0     POPLE FOR COULOMB   POPLE FOR EXCHANGE
INF(26)=1     ATMOL FOR COULOMB   POPLE FOR EXCHANGE
INF(26)=2     POPLE FOR COULOMB   ATMOL FOR EXCHANGE
INF(26)=3     ATMOL FOR COULOMB   ATMOL FOR EXCHANGE
```

On the basis of the value of inf(26), some logical and integer variables are set:

```
SUBROUTINE SHELLXN                                                    1
USE NUMBERS                                                           2
USE LMAXXX                                                            3
USE PARAME_MODULE                                                     4
USE MEMORY_G                                                          5
USE MEMORY_BIPOBU                                                     6
USE MEMORY_QERVI                                                      7
USE MEMORY_SCREEN                                                     8
USE MEMORY_KG123                                                      9
USE MEMORY_SCOST                                                      10
USE BASOLD_MODULE                                                     11
USE PARAL1_MODULE                                                     12
USE PARINF_MODULE                                                     13
USE BASATO_MODULE                                                     14
USE GVECT_MODULE                                                      15
USE PESVRS_MEMORY                                                     16
USE ROTMATRIX                                                         17
USE CLADDD_MODULE                                                     18
USE ONE_SIDED_COMMS                                                   19
IMPLICIT REAL(FLOAT) (A-H,O-Z)                                        20
CHARACTER*7 ZNAMZ                                                     21
LOGICAL IMMUM,LLCOUL,LEXCH4A,MPESO,NPF22,L3GEL4,MOTUHF                22
LOGICAL LCOUL,LPINT,IROUR,ISOUR,MCOUL,NOTUHF,L3EQL4,LEXCH5            23
LOGICAL NW,LEXCH2,LEXCH4,LPOPLJ,LJINT,LSTA34,LATMOL                   24
LOGICAL BIPOC_ACTIVE,BIPOX_ACTIVE,IMMUMBI                             25
INCLUDE 'TRAVIC.INC'                                                  26
COMMON/DCOMM/DDD,DX,DY,DZ                                             27
COMMON/TCOMM/ACCFAC                                                   28
DIMENSION PTEM(LMAXX25*2),RJ(LMAXX2525),RJ2(LMAXX2525),               29
*RO(LMAXX2525),RK(LMAXX2525),MOVF34(48),
*SS34(LMAX_COUL25),TT12(LMAX_COUL25),
*SS12(LMAXX25*LMAX_COUL25),WW12(LMAXX35*MAX_PRIM100)
INTEGER,DIMENSION(:),ALLOCATABLE :: INZILA,ILA                        30
INTEGER,DIMENSION(:),ALLOCATABLE :: IDIPC_VRS,ILW,ILV,ILU,MGIPI       31
REAL(FLOAT),DIMENSION(:),ALLOCATABLE :: VDID_VRS                      32
DATA ZNAMZ/'SHELLXN'/                                                 33
MOUT10=IUNIT(40)                                                      34
BIPOC_ACTIVE=INF(41).LT.20000                                         35
BIPOX_ACTIVE=INF(43).LT.20000                                         36
IF(BIPOC_ACTIVE)CALL GENPOD(IDICOU,MOUT10)                            37
NPOLEJ=(IDICOU+1)**2                                                  38
NPOLEX=(IDIPEX+1)**2                                                  39
I=INF(26)/2                                                           40
J=INF(26)-I-I                                                         41
LAF1=INF(20)+1                                                        42
NSPSTA=INF(64)+1                                                      43
NUGIR=INF(79)                                                         44
NTUT=INF(19)                                                          45
NHSQ=INF(145)                                                         46
MVF=INF(2)                                                            47
NOTUHF=NSPSTA.EQ.1                                                    48
LPINT=I.NE.0                                                          49
LJINT=J.NE.0                                                          50
LPOPLJ=.FALSE.                                                        51
ALLOCATE(MGIPI(NUGIR),STAT=I)                                         52
IF(I.NE.0)CALL ERRVRS(0,ZNAMZ,'MGIPI ALLOCATION')                     53
ALLOCATE(ILU(NUGIR),STAT=I)                                           54
IF(I.NE.0)CALL ERRVRS(0,ZNAMZ,'ILU ALLOCATION')                       55
ALLOCATE(ILV(NUGIR),STAT=I)                                           56
IF(I.NE.0)CALL ERRVRS(0,ZNAMZ,'ILV ALLOCATION')                       57
ALLOCATE(ILW(NUGIR),STAT=I)                                           58
IF(I.NE.0)CALL ERRVRS(0,ZNAMZ,'ILW ALLOCATION')                       59
ALLOCATE(IDIPC_VRS(NUGIR),STAT=I)                                     60
IF(I.NE.0)CALL ERRVRS(0,ZNAMZ,'IDIPC_VRS ALLOCATION')                 61
ALLOCATE(VDID_VRS(NUGIR),STAT=I)                                      62
IF(I.NE.0)CALL ERRVRS(0,ZNAMZ,'VDID_VRS ALLOCATION')                  63
ALLOCATE(ILA(LIM025),STAT=I)                                          64
IF(I.NE.0)CALL ERRVRS(0,ZNAMZ,'ILA ALLOCATION')                       65
ALLOCATE(INZILA(LAF1),STAT=I)                                         66
IF(I.NE.0)CALL ERRVRS(0,ZNAMZ,'INZILA ALLOCATION')                    67
INZILA(1)=0                                                           68
MVLA=0                                                                69
998 CALL IGPVAL                                                       70
```

Table 4.10. Definitions before the DO-loop chain in the *SHELLXN* subroutine. The quantities are discussed in the text.

1 IF(MVLA.EQ.MVLAF)GOTO 997		71	Implicit **DO-loop** on the couple sets
MVLA=MVLA+1		72	
NSTC12=IDMCOU(MVLA)		73	Maximum number of **g** stars for the MVLA CS to be considerated in the Coulomb term ($GMFCAL$)
LPF11=IDMEXG(MVLA)		74	Maximum number of **g** stars for the MVLA CS to be considerated in the Exchange term ($SHELLD$)
NGISH=MAX(NSTC12,LPF11)		75	
IF(NGISH.EQ.0)GOTO 1		76	If no translation vectors, cycle
ITASK=ITASK+1		77	
IF(ITASK.NE.MPOINT)GOTO 1		78	Counter from the $IGPVAL$ subroutine
NYC=NCF(MVLA)		79	Mother couple of the set, $\mu\nu$
L1=LA3(NYC+1)		80	L1=μ, first shell
L2=LA4(NYC+1)		81	L2=ν, second shel
CALL ZFEASV(L1,L2)		82	See section 4.2.4
CALL VIC2(MVLA)		83	See section 4.2.6
NW=LAT1.GE.2.OR.LAT2.GE.2		84	L1 or L2 are p, d or f shells (ATMOL package, see section 4.3.1)
ISOUR=LJINT.OR.NW		85	See section 4.3.1
IROUR=LPINT.OR.NW		86	See section 4.3.1
IF(LPF11.NE.0.AND.BIPOX_ACTIVE) READEX(L1,L2)	CALL	87	Read multipolar components for the bipolar expansion of exchange terms, see section 4.3.4
IROSE=IROF(MVLA)		88	Starting point in the list of $\mathbf{g}^{irr}$ for the MVLA CS ($GMFCAL$)
ICOSTA=NSTATG(MVLA)		89	Starting point of the MVLA CS in $F^{\mathbf{g}}_{irr}$ ($GMFCAL$)
NUGIR=IROSE		90	Counter for the third DO-loop
DO 3 MGISH=1,NGISH		91	Maximum number of **g** stars to be considered (**2nd DO-loop**)
LEXCH2=MGISH.GT.LPF11		92	See text for explanation: if true the exchange terms for L1L2 are not calculated
MCOUL=MGISH.GT.NSTC12		93	See text for explanation: if true the coulomb terms for L1L2 are not calculated
DO 3 NGI=1,NSHGI(IROSE+MGISH)		94	Number of $\mathbf{g}^{irr}$ for each star (**3rd DO-loop**)
NUGIR=NUGIR+1		95	
MG=NNGI(NUGIR)		96	$\mathbf{g}^{irr}$ vector of the MGISH star
X21OLD=XGOLD(1,MG)+X2OLD		97	Reference distance $\mathbf{s}_a - \mathbf{s}_b - \mathbf{g}$ for selection, see sections 4.2.5 and 4.2.6
...		99	The same for y and z cartesian coordinates
CXBNEW(1)=XG(1,MG)+X2NEW		100	Actual $\lvert\mathbf{s}_a - \mathbf{s}_b - \mathbf{g}\rvert$
...		102	The same for y and z cartesian coordinates
XCK(1,3)=CXBNEW(1)		103	
...		105	The same for y and z cartesian coordinates
XCK_OLD(1,3)=x21old		106	
...		108	The same for y and z cartesian coordinates
IANUM=.TRUE.		109	Logical variable, re-set in the $VIC4$ routine
IF(MCOUL)GOTO 20		110	
CALL CLASSS(ILA,INZILA)		111	
IANUM=INZILA(LAF1).EQ.0		112	
IF(IANUM)GOTO 20		113	
CALL VIC3		114	
IF(IANUM)GOTO 20		115	
XPNEW=CXBNEW(1)*FA2NEW		116	
...		118	The same for y and z cartesian coordinates
S12=SX12-F12OLD*RSQOLD		119	
CALL POLIPO(L1,L2,IDICOU,SS12,XPNEW,YPNEW,ZPNEW)		120	
TT12(1:NPOLEJ)=0._FLOAT		121	
WW12(1:NMAXP*JANUM)=0._FLOAT		122	
LATMOL=.FALSE.		123	
GOTO 2020		124	
20 IF(LEXCH2)GOTO 3		125	
2020 IG1=LG(1,MG)		126	
IG2=LG(2,MG)		127	
IG3=LG(3,MG)		128	
IF(ONE_SIDED_OK)THEN		129	
JCOUNT=0		130	
NODE_TO_GET=0		131	
ELSE		132	
JCOUNT=LNGTH		133	
ENDIF		134	
LPF22=0		135	
ISTA34=0		136	

Table 4.11. The first three DO-loops of the chain. In parenthesis: the subroutine in which the quantities are calculated, in chapter 3.1.

DO MVLU=1,MVLAF	137	Loop on the second couple set (**4th DO-loop**)
IF(.NOT.IANUM)ISTA34=IDMCOU(MVLU)	138	If L1L2^g overlap is not negligible for the Coulomb term, ISTA34=maximum number of translation vectors for the couple MVLU
IF(.NOT.LEXCH2)LPF22=IDMEXL(MVLU)	139	If L1L2^g overlap is not negligible for the exchange term, IDMEXL=maximum number of translation vectors for the couple MVLU
NSHTOT=MAX(ISTA34,LPF22)	140	No translation vector for the MVLU set
IF(NSHTOT.EQ.0)CYCLE	141	
LSTA34=ISTA34.NE.0	142	
NPF22=LPF22.NE.0	143	
NYC=NCF(MVLU)+1	144	First couple of the MVLUset
NCFU=NCF(MVLU+1)	145	Last couple of the MVLU set
IPETAL=INZVLB(MVLU)	146	Starting point of irreducible translation vectors in NNGI for the MVLU set ($GMFCAL$)
NUMBO=NSTATG(MVLU)	147	Starting point of the set in the irreducible P and F matrices
LPESO=JNCDU(MVLU).NE.1	148	If JNCDU(MVLU)=2 the couple MVLU is related to another couple by hermiticity
L33=LA3(NYC)	149	First shell for the second electron, σ
L44=LA4(NYC)	150	Second shell for the second electron, ω
L3EQL4=L33.EQ.L44	151	
CALL VIC4	152	Quantities to be used in the "core" of the integral calculation
MOVF34(1:MVF)=NOVF34(1:MVF,LAT3,LAT4)	153	Starting point of each operator for the couple L3L4 in the ZO34 vector of direct products ($GSYM22$)
NW=LAT3.GE.2.OR.LAT4.GE.2	154	
ID34D=ID34+ID34	155	ID34=size of the matrix of L3L4 ($VIC4$)???
IF(LSTA34)THEN	156	If the coulomb term must be calculated
SIG34=1._FLOAT/(EX3OLD+EX4OLD)	157	$\frac{1}{\gamma+\delta}$
FA4OLD=EX4OLD*SIG34	158	$\frac{\delta}{\gamma+\delta}$
F34OLD=EX3OLD*FA4OLD	159	$\frac{\gamma\delta}{\gamma+\delta}$
FA4NEW=EX4NEW/(EX3NEW+EX4NEW)	160	Actual $\frac{\delta}{\gamma+\delta}$
BILBO=SCOST(MVLU)+S12	161	Overlap L1L2^g+constant term of the L3L4 overlap
T8CO=(SQRT(SIG12*SIG34)+(SIG12+SIG34)*0.5_FLOAT)*T7CO	162	
FRODO=PINF45*T8CO	163	
X7=XLOLD(1,L44)-XLOLD(1,L33)	164	
...	166	The same for y and z cartesian coordinates
LPOPLJ=.NOT.(ISOUR.OR.NW)	167	Condition for POPLE/ATMOL packages for coulomb integrals, discussed in section 4.3.1
NL8=ICCS2(NYC)	168	
DO KSH=1,ISTA34	169	
I=NQGSHG(NGSHG(NL8+KSH)+1)	170	
VDID=MIN((BILBO-((X7+XGOLD(1,I))**2+(Y7+XGOLD(2,I))**2	171	
* +(Z7+XGOLD(3,I))**2)*F34OLD)*CCCFAJ,1._FLOAT)	172	
IDIPC_VRS(KSH)=MAX(0,IDICOU-INT(ADICOU*VDID))	173	
VDID_VRS(KSH)=VDID*FRODO+T8CO	174	
ENDDO	175	
ENDIF	176	If the exchange term must be calculated
IF(NPF22)THEN	177	$\frac{\gamma}{\gamma+\alpha}$
G13NEW=EX3NEW/(EX3NEW+EX1NEW)	178	$\frac{\delta}{\delta+\beta}$
G24NEW=EX4NEW/(EX4NEW+EX2NEW)	179	
G1324N=G13NEW-G24NEW	180	$\frac{1}{\gamma+\alpha}$
BILBO=1._FLOAT/(EX1OLD+EX3OLD)	181	$\frac{1}{\delta+\beta}$
FRODO=1._FLOAT/(EX2OLD+EX4OLD)	182	$\frac{\gamma}{\gamma+\alpha}$
G13OLD=EX3OLD*BILBO	183	$\frac{\gamma\alpha}{\gamma+\alpha}$
F13OLD=EX1OLD*G13OLD	184	$\frac{\delta}{\delta+\beta}$
G24OLD=EX4OLD*FRODO	185	$\frac{\delta\beta}{\delta+\beta}$
F24OLD=EX2OLD*G24OLD	186	
G1324O=G13OLD-G24OLD	187	
T8EX=(SQRT(BILBO*FRODO)+(BILBO+FRODO)*0.5_FLOAT)*T7EX	188	
T8EX44=T8EX*PINF44	189	
NPOL13=ID13*NPOLEX	190	
NPOL24=ID24*NPOLEX	191	
LPOPLE=IROUR.OR.NW		Condition for POPLE/ATMOL packages for exchange integrals, discussed in section 4.3.1

Table 4.12. The fourth DO-loop on the couple sets for the second electron (it continues in tab.4.13). In parenthesis: the subroutine in which the quantities are calculated, in chapter 3.1. α, β, γ and δ are the exponents of the adjoined gaussians of L1, L2, L3 and L4 shells, respectively.

IF(LPESO)THEN	192	
G14NEW=EX4NEW/(EX4NEW+EX1NEW)	193	
G23NEW=EX3NEW/(EX3NEW+EX2NEW)	194	
G1423N=G14NEW-G23NEW	195	
BILBO=1._FLOAT/(EX1OLD+EX4OLD)	196	
FRODO=1._FLOAT/(EX2OLD+EX3OLD)	197	
G14OLD=EX4OLD*BILBO	198	
F14OLD=EX1OLD*G14OLD	199	
G23OLD=EX3OLD*FRODO	200	
F23OLD=EX2OLD*G23OLD	201	
G1423O=G14OLD-G23OLD	202	
U8EX=(SQRT(BILBO*FRODO)+(BILBO+FRODO)*0.5_FLOAT)*T7EX	203	
U8EX44=U8EX*PINF44	204	
NPOL14=ID14*NPOLEX	205	
NPOL23=ID23*NPOLEX	206	
ENDIF	207	
ENDIF	208	
DO NCU=NYC,NCFU	209	(**5th DO-loop**)
L3=LA3(NCU)	210	First shell of the couple
L4=LA4(NCU)	211	Second shell of the couple
N34V=ICCS2(NCU)	212	Starting point in the NGSHG vector for the couple (GV)
NSH13=IDMEX1(L3)	213	**g** stars to be considerated for the L1L3 couple ($ZFEASV$)
NSH23=IDMEX2(L3)	214	**g** stars to be considerated for the L2L3 couple ($ZFEASV$)
NSH24=IDMEX2(L4)	215	**g** stars to be considerated for the L2L4 couple ($ZFEASV$)
NSH14=IDMEX1(L4)	216	**g** stars to be considerated for the L1L4 couple ($ZFEASV$)
L3GEL4=LPESO.OR.L3.GE.L4	217	Two couples related by hermiticity or in the lower-half diagonal
LEXCH4A=NSH13.NE.0.AND.NSH24.NE.0	218	**g** stars for the $< L1L3\|L2L4 >$ integral non null
MPESO=NSH14.NE.0.AND.NSH23.NE.0.AND.LPESO	219	**g** stars for the $< L1L4\|L2L3 >$ integral non null, and twocouples related by hermiticity
X31OLD=XLOLD(1,L3)-X1OLD	220	Reference distance L3-L1 ($VIC2$)
...	222	It is repeated for y and z
CXCNEW(1)=XL(1,L3)-X1NEW	223	Actual distance L3-L1 ($VIC2$)
...	225	It is repeated for y and z
X41OLD=XLOLD(1,L4)-X1OLD	226	Reference distance L4-L1 ($VIC2$)
...	228	It is repeated for y and z
CXDNEW(1)=XL(1,L4)-X1NEW	229	Actual distance L4-L1 ($VIC2$)
...	231	It is repeated for y and z
IF(LSTA34)THEN	232	
XQNEW=XPNEW-CXCNEW(1)	233	
...	235	y and z
XQOLD=XPOLD-X31OLD	236	
...	238	y and z
INZCIL=INZILA(L4)	239	Starting point of the **h'** vectors for L4
ILTUTLVRS=INZILA(L4+1)-INZCIL	240	Total number of **h'** vectors, $\{\mathrm{H}\}_{L4}$
I1=INZILA(L3)	241	Starting point of the **h** vectors for L3
ILTUTK=INZILA(L3+1)-I1	242	Total number of **h** vectors, $\{\mathrm{H}\}_{L3}$
DO I=1,ILTUTK	243	
N=ILA(I1+I)	244	
KPESO(N)=.FALSE.	245	
ILW(I)=N	246	The **h** vectors are stored
ENDDO	247	
ENDIF	248	
IF(NPF22)THEN	249	If the exchange terms are present
IF(LEXCH4A)THEN	250	If $< L1L3\|L2L4 >$ is present
X31NES=CXCNEW(1)*G13NEW-CXBNEW(1)	251	Actual $\frac{\gamma}{\alpha+\gamma}(x(L3)-x(L1))-(x(L2)-x(L1)-x\mathbf{g})$
..	253	y and z
X31OLS=X31OLD*G13OLD-X21OLD	254	Reference $\frac{\gamma}{\alpha+\gamma}(x(L3)-x(L1))-(x(L2)-x(L1)-x\mathbf{g})$
...	256	y and z
BILBO=SCOST2(L4)	257	$\frac{3}{4}\left(\frac{4\beta\delta}{(\beta+\delta)^2}\right)$ $(ZFEASV)$
B24OLD=BILBO-BCCFAK	258	$\frac{3}{4}\left(\frac{4\beta\delta}{(\beta+\delta)^2}\right)$ + 2.303$ITOL(3)$ $(ZFEASV$ and $SHELLD)$
SX1324=SCOST1(L3)+BILBO	259	$\frac{3}{4}\left(\frac{4\alpha+\gamma}{(\alpha+\gamma)^2}\right)+\frac{3}{4}\left(\frac{4\beta\delta}{(\beta+\delta)^2}\right)$
N=LA34V2(L4)	260	Starting point in the NQGSHG list of translation vectors of the L2L4 couple ($ZFEASV$)

Table 4.13. Part of the fourth and the fifth DO-loop.

I1=INZL12(L4,2)	261	Number of multipoles for the L2L4 couple ($READEX$)
DO I=NGSHG(N+1)+1,NGSHG(N+NSH24+1)	262	
INVOR(NQGSHG(I))=I1	263	
I1=I1+NPOL24	264	
ENDDO	265	
ICAME=LA34V1(L3)	266	Starting point in the NQGSHG list of translation vectors of the L1L3 couple ($ZFEASV$)
I1=INZL12(L3,1)	267	Number of multipoles for the L1L3 couple ($READEX$)
DO I=NGSHG(ICAME+1)+1,NGSHG(ICAME+NSH13+1)	268	
JNVOR(NQGSHG(I))=I1	269	
I1=I1+NPOL13	270	
ENDDO	271	
ENDIF	272	
IF(MPESO)THEN	273	
X41NES=CXDNEW(1)*G14NEW-CXBNEW(1)	274	
...	276	
X41OLS=X41OLD*G14OLD-X21OLD	277	
...	279	
X32OLD=X31OLD-X21OLD	280	
...	282	
FRODO=SCOST2(L3)	283	
B23OLD=FRODO-BCCFAK	284	
SX1423=SCOST1(L4)+FRODO	285	
N=LA34V2(L3)	286	
I1=INZL12(L3,2)	287	
DO I=NGSHG(N+1)+1,NGSHG(N+NSH23+1)	288	
KNVOR(NQGSHG(I))=I1	289	
I1=I1+NPOL23	290	
ENDDO	291	
JCAME=LA34V1(L4)	292	
I1=INZL12(L4,1)	293	
DO I=NGSHG(JCAME+1)+1,NGSHG(JCAME+NSH14+1)	294	
LNVOR(NQGSHG(I))=I1	295	
I1=I1+NPOL14	296	
ENDDO	297	
ENDIF	298	
ENDIF	299	
JPETAL=IPETAL	300	Starting point of the irreducible **g'** vectors for the selected MVLU couple set
DO KSH=1,NSHTOT	301	(**6th DO-loop**)
LEXCH5=KSH.LE.LPF22	302	
LEXCH4=LEXCH5.AND.LEXCH4A	303	
LEXCH5=MPESO.AND.LEXCH5	304	
N34V=N34V+1	305	
IMMUM=L3GEL4.AND.KSH.LE.ISTA34	306	
IMMUMBI=IMMUM.AND.BIPOC_ACTIVE	307	
MOTUHF=LEXCH4.OR.LEXCH5	308	
NL8=NGSHG(N34V)	309	starting point of the translation vectors of the NCU couple in NQGSHG
NL9=NGSHG(N34V+1)	310	starting point of the translation vectors of the NCU+1 couple in NQGSHG
IF(IMMUM)THEN	311	
IF(L3EQL4.AND.KSH.EQ.1)THEN	312	
ILTUTL=0	313	
ELSE	314	
ILTUTL=ILTUTLVRS	315	
ENDIF	316	
LCOUL=(ILTUTK+ILTUTL).NE.0	317	
IF(LCOUL)THEN	318	
IDIPC=IDIPC_VRS(KSH)	319	
VDID=VDID_VRS(KSH)	320	
ENDIF	321	
ELSE	322	
IF(MOTUHF)THEN	323	
LCOUL=.FALSE.	324	
ELSE	325	
JPETAL=JPETAL+NL9-NL8	326	
CYCLE	327	
ENDIF	328	
ENDIF	329	
MOTUHF=(.NOT.MOTUHF).OR.NOTUHF	330	
DO NL=NL8+1,NL9	331	(**7th DO-loop**)
JPETAL=JPETAL+1	332	
MGP=NQGSHG(NL)	333	

Table 4.14. Part of the fifth, the sixth and part of the seventh DO-loops.

NU=INO(JPETAL)*ID34+NUMBO	334	
MV=INOIV(JPETAL)	335	
IF(MOTUHF)THEN	336	
IF(MV.EQ.1)THEN	337	
PTEM(1:ID34)=PG_IRR(NU+1:NU+ID34)	338	
ELSE	339	
PTEM(1:ID34)=0._FLOAT	340	
DO J=MOVF34(MV-1)+1,MOVF34(MV)	341	
I=NO34(J)	342	
PTEM(I)=ZO34(J)*PG_IRR(NOM34(J)+NU)+PTEM(I)	343	
ENDDO	344	
ENDIF	345	
ELSE	346	
IF(MV.EQ.1)THEN	347	
J=NU+NTUT	348	
DO I=1,ID34	349	
PTEM(I)=PG_IRR(NU+I)	350	
PTEM(ID34+I)=PG_IRR(J+I)	351	
ENDDO	352	
ELSE	353	
PTEM(1:ID34D)=0._FLOAT	354	
DO J=MOVF34(MV-1)+1,MOVF34(MV)	355	
I=NO34(J)	356	
N=NOM34(J)+NU	357	
PTEM(I)=ZO34(J)*PG_IRR(N)+PTEM(I)	358	
PTEM(ID34+I)=ZO34(J)*PG_IRR(NTUT+N)+PTEM(ID34+I)	359	
ENDDO	360	
ENDIF	361	
ENDIF	362	
IF(L3EQL4.AND.MGP.GT.NN1(MGP))THEN	363	
LLCOUL=.FALSE.	364	
ELSE	365	
LLCOUL=LCOUL	366	
IF(IMMUMBI)THEN	367	
JCOUNT=JCOUNT-NPOLEJ	368	
IF(ONE_SIDED_OK)THEN	369	
IF(JCOUNT.LT.0)THEN	370	
CALL ONE_SIDED_GET(BIPOJ,STATIC_BIPOJ(1),STATIC_LNGTH,NODE_TO_GET)	371	
JCOUNT=STATIC_LNGTH-NPOLEJ	372	
NODE_TO_GET=NODE_TO_GET+1	373	
ENDIF	374	
SS34(1:NPOLEJ)=BIPOJ(JCOUNT+1:JCOUNT+NPOLEJ)	375	
ELSE	376	
IF(JCOUNT.LT.0)THEN	377	
CALL RREAD(MOUT10,SS34,NPOLEJ)	378	
ELSE	379	
SS34(1:NPOLEJ)=BIPOJ(JCOUNT+1:JCOUNT+NPOLEJ)	380	
ENDIF	381	
ENDIF	382	
ENDIF	383	
ENDIF	384	
X9OLD=XGOLD(1,MGP)	385	
...	387	
X9NEW=XG(1,MGP)	388	
...	390	
X6OLD=X41OLD+X9OLD	391	
...	393	
CXFNEW(1)=CXDNEW(1)+X9NEW	394	
...	396	
IGP1=LG(1,MGP)	397	
IGP2=LG(2,MGP)	398	
IGP3=LG(3,MGP)	399	
IF(LLCOUL)THEN	400	
ILTUTY=0	401	
ILTUTX=ILTUTK	402	
DO I=1,ILTUTL	403	
N=ILA(INZCIL+I)	404	
N=KG123(LG(1,N)-IGP1,LG(2,N)-IGP2,LG(3,N)-IGP3)	405	
IF(N.EQ.0)CALL ERRVRS(0,ZNAMZ,	406	
*'DIRECT LATTICE VECTOR OUT OF RANGE - INCREASE LIM007')	407	
IF(KPESO(N))THEN	408	
ILTUTX=ILTUTX+1	409	Counter for the **h** vectors and the **h'** vectors not belonging to $\{H\}_{L3}$
ILW(ILTUTX)=N	410	$\mathbf{h} \in \{H\}_{L3} + \mathbf{h'} \notin \{H\}_{L3}$
ELSE	411	
ILTUTY=ILTUTY+1	412	Counter for the **h'**=**h** vectors having WEIGHT=2

Table 4.15. .

ILV(ILTUTY)=N	413	list of the **h**=**h'** vectors belonging both to $\{H\}_{L3}$ and to $\{H\}_{L4}$
WEIGHT(N)=2._FLOAT	414	
ENDIF	415	
ENDDO	416	
ILTUTC=0	417	
ILTUTM=0	418	
XX=(X31OLD-X6OLD)*FA4OLD+XQOLD	419	
...	421	
DO I=1,ILTUTX	422	
N=ILW(I)	423	
IF(VDID.LT.((XX-XGOLD(1,N))**2+(YY-XGOLD(2,N))**2	424	
*+(ZZ-XGOLD(3,N))**2))THEN		
ILTUTM=ILTUTM+1	425	Counter for the vectors for which the integral can be approximated
MGIPI(ILTUTM)=N	426	Vectors for which the integral can be approximated
ELSE	427	
ILTUTC=ILTUTC+1	428	Counts the vectors for the exact calculation
ILU(ILTUTC)=N	429	Vectors for the exact calculation
ENDIF	430	
ENDDO	431	
CXENEW(1)=CXCNEW(1)-CXFNEW(1)	432	
...	434	
LLCOUL=ILTUTC.NE.0	435	One or more vectors for which the integral must be calculated exactly
IF(ILTUTM.NE.0)THEN	437	
XX=CXENEW(1)*FA4NEW+XQNEW	438	
...	440	
N=MGIPI(1)	441	
DX=XX-XG(1,N)	442	
...	444	
DDD=DX*DX+DY*DY+DZ*DZ	445	
CALL COVVVV(IDIPC,WEIGHT(N))	446	
DO I=2,ILTUTM	447	
N=MGIPI(I)	448	
DX=XX-XG(1,N)	449	
...	451	
DDD=DX*DX+DY*DY+DZ*DZ	452	
CALL COVUVU(IDIPC,WEIGHT(N))	453	
ENDDO	454	
CALL ISYMD(IDIPC,TT12,SS34)	455	
ENDIF	456	
IF(LLCOUL)THEN	457	
IF(LPOPLJ)THEN	458	
RJ(1:MOF)=0._FLOAT	459	
CALL SHELLJ(RJ,ILU)	460	Integrals are calculated with the POPLE package
ELSE	461	
CALL VIC5JD(WW12,PTEM,ILU)	462	Integrals are calculated with the ATMOL package
LATMOL=.TRUE.	463	
ENDIF	464	
ENDIF	465	
DO I=1,ILTUTY	466	
WEIGHT(ILV(I))=1._FLOAT	467	
ENDDO	468	
ENDIF	469	
NCICLX=0	470	
IF(LEXCH4)CALL CLAKKK(RO,MGIPI)	471	
IF(LEXCH5)CALL CLALLL(RK,MGIPI)	472	
IF(LLCOUL.AND.LPOPLJ)THEN	473	
IF(NCICLX.EQ.0)GOTO 10095	474	
IF(MOTUHF)THEN	475	
IF(NCICLX-2)11091,11092,11093	476	
11091 CALL COMBJ1(RJ,RO)	477	
GOTO 10095	478	
11092 CALL COMBJ2(RJ,RK)	479	
GOTO 10095	480	
11093 CALL COMBJ3(RJ,RO,RK)	481	
ELSE	482	
IF(NCICLX-2)12091,12092,12093	483	
12091 CALL COMBX1(RJ2,RO)	484	
GOTO 12094	485	
12092 CALL COMBX2(RJ2,RK)	486	
GOTO 12094	487	

Table 4.16. .

```
I=INF(26)/2
```

```
12093 CALL COMBX3(RJ2,RO,RK)                                                    488
12094 CALL MXMB(RJ2,ID34,1,PTEM,                                                489
*1,ID34,FG_IRR(ICOSTA+1),1,NTUT,ID12,ID34,2)                                    490
ENDIF                                                                           491
10095 CALL MXMB(RJ,ID34,1,PTEM,1,1,FG_IRR(ICOSTA+1),1,1,ID12,ID34,1)            492
ELSE                                                                            493
IF(NCICLX.NE.0)THEN                                                             494
IF(NCICLX-2)10091,10092,10093                                                   495
10091 CALL COMBX1(RJ,RO)                                                        496
GOTO 10094                                                                      497
10092 CALL COMBX2(RJ,RK)                                                        498
GOTO 10094                                                                      499
10093 CALL COMBX3(RJ,RO,RK)                                                     500
10094 CALL MXMB(RJ,ID34,1,PTEM,                                                 501
*1,ID34,FG_IRR(ICOSTA+1),1,NTUT,ID12,ID34,NSPSTA)                               502
ENDIF                                                                           503
ENDIF                                                                           504
ENDDO                                                                           505
ENDDO                                                                           506
IF(LSTA34)THEN                                                                  507
DO I=1,ILTUTK                                                                   508
KPESO(ILW(I))=.TRUE.                                                            509
ENDDO                                                                           510
ENDIF                                                                           511
IPETAL=IPETAL+NHSQ                                                              512
ENDDO                                                                           513
ENDDO                                                                           514
IF(JCOUNT.LT.0)REWIND MOUT10                                                    515
IF(IANUM)GOTO 3                                                                 516
CALL MXMB(SS12,1,ID12,TT12,1,1,FG_IRR(ICOSTA+1),1,1,ID12,NPOLEJ,1)              517
IF(LATMOL)THEN                                                                  518
I1=0                                                                            519
N=0                                                                             520
DO I=1,JANUM                                                                    521
CALL MXMB(CFACJ(I1+1),NMAXP,1,WW12(N+1),1,1,                                    522
*FG_IRR(ICOSTA+1),1,1,ID12,NMAXP,1)                                             523
I1=I1+NDIMJ1                                                                    524
N=N+NMAXP                                                                       525
ENDDO                                                                           526
ENDIF                                                                           527
3 ICOSTA=ICOSTA+ID12                                                            528
GOTO 998                                                                        529
997 DEALLOCATE(INZILA)                                                          530
DEALLOCATE(ILA)                                                                 531
DEALLOCATE(VDID_VRS)                                                            532
DEALLOCATE(IDIPC_VRS)                                                           533
DEALLOCATE(ILW)                                                                 534
DEALLOCATE(ILV)                                                                 535
DEALLOCATE(ILU)                                                                 536
DEALLOCATE(MGIPI)                                                               537
IF(BIPOC_ACTIVE)DEALLOCATE(BIPOJ)                                               538
IF(ONE_SIDED_OK)CALL FREE_SHARED_BIPO                                           539
CALL IGPRST                                                                     540
RETURN                                                                          541
END                                                                             542
```

Table 4.17. .

```
J=INF(26)-I-I
LPINT=I.NE.0     [inf(26).GE.2]
LJINT=J.NE.0     [inf(26).EQ.1.OR.inf(26).EQ.3]
LPOPLJ=.FALSE.   [re-set later on]
```

(see instructions 40-41 and 49-50-51 in tab.4.10).

We underline these points, as many logical variables refers to the kind of shells involved in the calculation. When the first couple $\mu\nu$, corresponding to L1L2 in the code, is selected, the *VIC2* subroutine, described in section 4.2.6, sets LAT1=LAT(L1) and LAT2=LAT(L2), used in the condition NW=LAT1.GE.2.OR.LAT2.GE.2. Two condition, then, are defined in the

first DO-loop (instructions number 85-86 in tab.4.11):

```
ISOUR=LJINT.OR.NW  (ATMOL for coulomb  or orbitals with LAT.GE.2)
IROUR=LPINT.OR.NW  (ATMOL for exchange or orbitals with LAT.GE.2)
```

In the fourth DO-loop, when the mother couple L3L4 is selected, the NW logical variable is redefined as NW=LAT3.GE.2.OR.LAT4.GE.2, after LAT3 and LAT4 have been defined in *VIC4* as LAT(L1) and LAT(L2).

Two other conditions, that will be used directly in the calculation of the bielectronic integrals (*SHELLJ*, *VIC5JD*, *CLAKKK* and *CLALLL*), are defined for coulomb and exchange terms:

```
LPOPLJ=.NOT.(ISOUR.OR.NW)  (coulomb)
LPOPLE=.NOT.(IROUR.OR.NW)  (exchange)
```

(167 and 192 in tab.4.12).

They will be used in the routine when the package for integral calculation has to be chosen: for coulomb terms the Pople package is used when LPOPLJ is true (all the shells have LAT$\leq$1 and inf(26) is equal to 0 or 2), for the exchange terms it is used when LPOPLE is true (all the shells have LAT$\leq$1 and inf(26) is equal to 0 or 1).

4.3.2 The DO-lops on the $\mu\nu$ mother couples of the sets

The first implicit DO-loop runs on the couple sets MVLA, and operates on the mother couple of each set (L1,L2 in the code); the quantities indicating the number of **g** stars (IDMCOU, IDMEXG) and the starting point in the $\mathbf{g}^{irr}$ list (IROF) and in $F^{\mathbf{g}}_{irr}$ (NSTATG) of $\mu\nu$ are recalled. Each single instructions from number 71 to number 89 are shown and commented in tab.4.11.

In the second DO-loop, running on the maximum number of **g** stars (NGISH, instructions 91-93 in tab.4.11) two logical variables are defined:

- LEXCH2=MGISH.GT.LPF11, the **g** star number is greater than the maximum number of **g** stars for exchange integrals
- MCOUL=MGISH.GT.NSTC12, the **g** star number is greater than the maximum number of **g** stars for coulomb integrals

If LEXCH2 is true, instructions regarding the exchange term must be skipped for terms containing L1L2, whereas if MCOUL is true, the coulomb term calculation must be skipped .

The third DO-loop runs on the irreducible **g** vectors of a star (that are 1 or 2), stored in the NNGI array (see the *GILDA1* subroutine). The first part of this loop is shown and commented in tab.4.11, from number 94 to 109.

If the MCOUL condition is verified (number 110 in tab.4.11), and so the considered **g** star is not included in the stars selected for the coulomb integrals involving the considered L1L2 couple, the selection of **h** vectors is not performed. On the contrary, if the MCOUL condition is not verified, the *CLASSS* subroutine is called (see section 4.3.3), in order to select the **h** vectors for wich the coulomb integrals have to be calculated exactly.

The condition IANUM=INZILA(inf(20)+1).EQ.0 is true if, given a general L3 shell of the unit cell, there are not any **h** vectors for which the coulomb terms must be calculated exactly, that is, there are not any cases in which one of the L3 shells penetrates the L1L2**g** distribution. In such a case the L1L2 couple can be disregarded in the calculation of coulomb terms, and, if it is so also for the exchange term (LEXCH2 verified), L1L2 can be completely disregarded, the code exits from the third and second DO-lops and the first loop starts again with the couple MVLA=MVLA+1.

When IANUM is false, the *VIC3* subroutine is called: a component of the coulomb integrals is here calculated (the **CFACJ** vector, *DFAC4* subroutine) and is related to the coefficients by which the spherical armonics can be espressed as a linear combination of Hermite polinomia[3]. A first selection is performed, in order to consider only the couples of primitives of the L1 and L2 shells giving a non negligible overlap (on the basis of the T_1 threshold). In the *DFAC4* routine the overlap between two primitives belonging to the two shells of the couple L1L2 (see eq.4.20) is calculated and stored as EXPACC. If it is less than ACCFAC=$10^{-ITOL(1)}$, the LSCREE logical variable is true (LSCREE=EXPACC.LE.ACCFAC), the program exit from the *DFAC4* and the loop on the primitives cycle in the *VIC3* routine. LCNUM is there set to zero, and it is incremented only when LSCREE is false. As a consequence, when all the overlap between the primitives are negligible, meaning that the L1L2 couple is negligible, LCNUM is equal to zero. The condition IANUM=LCNUM.EQ.0 is then defined in the *VIC3* subroutine. If IANUM=.TRUE., the *SHELLXN* routine disregard the L1L2 couple for the coulomb integrals (instruction number 113 in tab.4.11).

On the contrary, when the product of the gaussian functions of the L1 shell by the ones of the L2 shell is non negligible (LSCREE=.FALSE. and

IANUM=.FALSE.), the x, y and z components of the vector $(\mathbf{s}_b - \mathbf{s}_a + \mathbf{g}) \cdot \frac{\beta}{\alpha+\beta}$ are calculated (instructions 116-118):

XPNEW=CXBNEW(1)*FA2NEW
YPNEW=CXBNEW(2)*FA2NEW
ZPNEW=CXBNEW(3)*FA2NEW

and the full overlap term of eq. 4.23 is evaluated (instruction 119):

S12=SX12-F12OLD*RSQOLD $\quad S12 = 0.75 * log\left(\frac{4\alpha\beta}{(\alpha+\beta)^2}\right) - \frac{\alpha\beta}{\alpha+\beta} \cdot |\mathbf{s}_a - \mathbf{s}_b - \mathbf{g}|^2$

The *POLIPO* subroutine is then called, in order to calculate the multipolar components for the bipolar expansion of the coulomb terms, whose first electron couple is L1L2.

Once the L1L2 couple has been selected for the coulomb calculation, we do not need to verify if it is selected also for the exchange terms, it will be done later on. The Bravais components of the considered irreducible **g** vector, LG, are then stored in IG1, IG2 and IG3. The if condition that follows (instruction 129-134) regards technical parameters about the parallel and serial calculations, and discussing about these details goes beyond our aim here.

DOPO VACANZE: (**LENGTH=lunghezza buffer, vedere poi bene dove LENGTH=MIN(IBUFFER
ISTART)

4.3.3 The selection of h vectors for exact coulomb integrals: the *CLASSS* subroutine

Consider the coulomb bielectronic integral

$$C = \sum_h \int \chi_\mu^0(\mathbf{r_1}) \chi_\nu^{\mathbf{g}}(\mathbf{r_1}) \widehat{I} \chi_\sigma^{\mathbf{h}}(\mathbf{r_2}) \chi_\omega^{\mathbf{h}+\mathbf{g}'}(\mathbf{r_2}) d\mathbf{r_1} d\mathbf{r_2} \tag{4.36}$$

As widely discussed in section 1.1.10 such an integral can be evaluated either exactly, or, when $\sigma^{\mathbf{h}}$ does not penetrate the charge distribution $\mu\nu^{\mathbf{g}}$, through a multipolar expansion (see equations from 1.44 to 1.46). For this reason **h** vectors in the previous equation must be split in two sets: {H} (finite set, exact integrals) and {H_A} (infinite multipolar expansion). The overlap between the

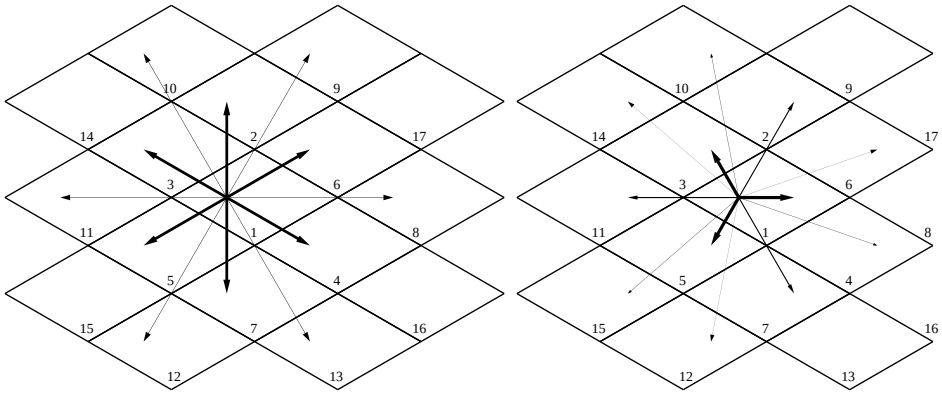

Figure 4.3. The first two stars of **g** for the 1,2 couple set type and the first three stars for the 1,3 couple set type.

first couple $\mu\nu$ and the adjoined gaussian of the σ shell is evaluated on the basis of the T_2 tolerance, in order to assigne the **h** vectors to one of the two sets.

This selection is performed in the *CLASSS* subroutine, called in the third "large" do-loop of the *SHELLXN* subroutine at each SCF step (instruction number 111 in tab.4.10). Given $\mu\nu$ (L1,L2 in the code), the **g** star (MGISH) and the irreducible **g** vector of the star (NGI), the *CLASSS* subroutine selects the **h** vectors (N) giving a non-null overlap between σ (J=1,LAF with LAF=inf(20) being the total number of shells) and $\mu\nu^{\mathbf{g}}$.

The **h** vectors belonging to the {H} set are then stored, in sequence (see *GCALCO* and *GV* subroutines), in the **ILA** vector and the starting point of each σ shell (J in the code) are given in the **INZILA** pointer. Tab.4.18 list ILA and INZILA for graphite; in fig.4.3 the scheme of the **g** stars for the couple 1,1 and 1,3 of graphite is shown. Note that a couple of vectors ILA/INZILA for each

$$[L1,L2,MGISH,NGI] = [\mu,\nu,star(\mathbf{g}),\mathbf{g}^{irr}]$$

set exists. As INZILA(1)=0 in any case, its value is assigned at the beginning of the *SHELLXN* subroutine (instruction number 68 in tab.4.10), whereas the *CLASSS* subroutine calculates only the elements from INZILA(2) to INZILA(LAF+1).

L1,L2	1,1
g star	1
$\mathbf{g}_{irr}$	1
NNGI(1)	1
INZILA	0 1 8 8 20 [note that the 13 couple is disregarded, INZILA(3)=INZILA(4)]
ILA	**1** **1** 2 3 4 5 6 7 **1** 2 3 4 5 6 7 9 10 13 17 18
L1,L2	2,1
g star	1
$\mathbf{g}_{irr}$	1
NNGI(1)	1
INZILA	0 1 8 11 23
ILA	**1** **1** 2 3 4 5 6 7 **1** 5 6 **1** 2 3 4 5 6 7 9 10 13 17 18
L1,L2	2,1
g star	2
$\mathbf{g}_{irr}$	1
NNGI(2)	2
INZILA	0 1 8 10 16
ILA	**1** **1** 2 4 6 8 10 14 **2** 6 **1** 2 6 10 14 18
L1,L2	2,2
g star	1
$\mathbf{g}_{irr}$	1
NNGI(1)	1
INZILA	0 7 26 38 56
ILA	**1** 2 3 4 5 6 7 **1** 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 **1** 2 3 4 5 6 7 9 10 13 17 18 **1** 2 3 4 5 6 7 8 9 10 11 13 14 15 17 18 27 31
L1,L2	2,2
g star	2
$\mathbf{g}_{irr}$	1
NNGI(2)	2
INZILA	0 8 22 30 47
ILA	**1** 2 4 5 6 7 8 10 **1** 2 3 4 5 6 7 8 10 12 13 14 16 18 **1** 2 4 5 6 10 13 18 **1** 2 3 4 5 6 7 8 9 10 13 14 17 18 24 28 31
L1,L2	2,2
g star	3
$\mathbf{g}_{irr}$	1
NNGI(3)	8
INZILA	0 4 18 23 35
ILA	**1** 2 4 8 **1** 2 3 4 5 6 7 8 10 12 14 16 20 22 **1** 2 4 6 10 **1** 2 3 4 5 6 7 8 10 13 14 18
L1,L2	2,2
g star	3
$\mathbf{g}_{irr}$	2
NNGI(4)	9
INZILA	0 4 18 23 35
ILA	**1** 3 5 9 **1** 2 3 4 5 6 7 9 11 13 15 17 21 23 **3** 5 9 13 17 **1** 3 5 6 9 13 15 17 21 23 27 35
L1,L2	2,2
g star	4
$\mathbf{g}_{irr}$	1
NNGI(5)	14
INZILA	0 7 14 17 29
ILA	**1** 2 4 6 8 10 14 **1** 2 4 6 8 10 14 **2** 6 10 **1** 2 4 5 6 8 10 13 14 18 24 28

Table 4.18. ILA and INZILA vectors for the couples 1,1, 1,2, and 2,2 of graphite (two shells for each atom, TOLINTEG 5 5 5 5 12; shells 1 and 2 on the same atom). The NNGI vector (MG in the code) is defined in the *GILDA1* subroutine, and lists the irreducible **g** vectors of a couple set type. The first vector of a shell is in bold size in ILA.

4.3.4 The bipolar expansion

In section 4.3.3 the **h** vectors have been split into two sub-families, the first one, {H}, for which the coulomb integrals are evaluated exactly, and the second one, $\{H_A\}$, for which they are calculated as three center integrals, evaluated through th multipolar expansion of the shell charge distributions. The exchange terms, on the contrary, are all calculated exactly (as bielectronic integrals). However, both the exact coulomb and exchange integrals can be approximated with a bipolar expansion of the two charge distributions, each one involving 3 coordinates, *when the charge distribution of the second electron is external to the*

charge distribution of the first electron.

Consider for example the coulomb integral $< \mu^0 \nu^{\mathbf{g}} | \sigma^{\mathbf{h}} \omega^{\mathbf{h}+\mathbf{g}'} >$; the value of the integral will go to zero when:

- **h** is very large
- $\nu^{\mathbf{g}}$ is far from μ^0 and/or $\omega^{\mathbf{g}'}$ is far from σ^0 (large **g** and **g'** vectors respectively)
- α, β, γ or δ exponents (one, some or all) are very large (in the case when the centers of two shells coincide, the overlap between two Gaussian functions decreases even if the two exponents are very different)

The second and the third factors of the list are much more important than the first one, as actually the charge involved in the interaction, $\int \mu^0 \nu^{\mathbf{g}} d\tau = \rho_{ab}$, is 1 for $\mu^0 = \nu^{\mathbf{g}}$ (remember that our AO are normalized in such a way that $\int \chi_\mu \chi_\mu d\tau = 1$), and decay exponentially as the distance or the exponents increase. For example, consider $\mathbf{s_a} = \mathbf{s_b}$ and $\mathbf{g} = 0$, $\alpha = 0.1$ and $\beta = 1000$: the charge decreases by a factor $\left(\frac{4\alpha\beta}{(\alpha+\beta)^2}\right)^{\frac{3}{4}} = 2\sqrt{2} \cdot 10^{-3}$ (see eq.4.21).

The reciprocal penetration between the two charges ρ_{ab} and ρ_{cd} is evaluated by performing the overlap $\int g_{ab} g_{cd} d\tau$ after having renormalized (VERO???) g_{ab} and g_{cd}, so that we get 1 for the complete penetration ($g_{ab} = g_{cd}$) and zero for distant charge distributions. We must, however, take into account that the integral itself is negligible for ρ_{ab} and ρ_{cd} very small, so that, in this case, we can accept a large penetration error.

Similar comments hold for the exchange terms, where the distance between the two distributions is given by $|\mathbf{g}|$ for terms $< \mu^0 \sigma^{\mathbf{h}} | \nu^{\mathbf{g}} \omega^{(\mathbf{h}+\mathbf{g}'-\mathbf{g})+\mathbf{g}} >$ and $< \mu^0 \omega^{\mathbf{h}+\mathbf{g}'} | \nu^{\mathbf{g}} \sigma^{(\mathbf{h}-\mathbf{g})+\mathbf{g}} >$.

The penetration thresholds s is set by default to $inf(41)$=14 (10^{-14}) and $inf(43)$=10 (10^{-10}) for coulomb and exchange terms respectively, whereas $inf(40)$=4 and $inf(42)$=2 indicate the order of the multipoles considered in the bipolar expansion for coulomb and exchange integrals respectively (different values can be set, see the CRYSTAL manual[4]). However, one can choose to avoid the bipolar approximation and prefer the exact calculation (see NOBIPOLA keyword in the CRYSTAL manual[4]), and the cost of the calculation increase up to a factor 3[4]. In this last case, $inf(41)$ and $inf(43)$ are set automatically to 200000, and two logical variables regulate the calculations (see

instructions 35 and 36 in tab.4.10):

```
BIPOC_ACTIVE=INF(41).LT.200000   (coulomb)
BIPOX_ACTIVE=INF(43).LT.200000   (exchange)
```

Note that the calculation can be performed exactly for one term and approximated for the other one: for example one could set by input $inf(41) = 2000000$ and run the calculation without the NOBIPOLA keyword, so that coulomb terms are calcuated exactly and exchange terms are approximated.

The structure of the *SHELLXN* subroutine, whose loops run over the $\mu\nu$ and $\sigma\omega$ couples, is not "natural" for the calculation of exchange terms, as the involved couples are $\mu\sigma$, $\nu\omega$, $\mu\omega$ and $\nu\sigma$. For a given couple $\mu\nu$, it is useful calculating the multipolar components before $SHELLXN$, in the *EXCBUF* subroutine, called by *INTCALC* when INF(43).LT.200000. The instruction number 87 in tab.4.11 means that if the actual exchange integral has to be calculated and BIPOLA is not switched off by input, the *READEX* subroutine reads and organises the multipoles of the couples containing μ (then, μ with all selected partners) and the ones containing ν calculated in the *EXCBUF* subroutine.
DOPO LE VACANZE SCRIVO UN COMMENTINO PER LA PARTE:
F(I)=SUM_J INTEGRAL(I,J)=SUM_J SUM_l1,l2 POLI(I,l1) R(l1,l2) POLI(l2,J) P_J
LIBRO ARANCIO PRIMO PP48:per i coulomb c'e' il trucchetto, per gli exchange no perche' h si mischia dappertutto eccetera. che si collega poi a NPOLj, NPOLX, dimensioni matrici e COVUVU e ISYMD.

4.3.5 Selection of exact and approximated bielectronic integrals

In section 4.2.1 we have shown how the overlap between the charge distributions of the two electrons can be obtained. For the present discussion, related to a general coulomb integral $< L1L2|L3L4 >$, we are interested in the natural logarithm of such a quantity, given in eq.4.24 and shown again in the following:

$$S_{12,34} = \frac{3}{4} ln \left(\frac{4(\alpha+\beta)(\gamma+\delta)}{(\alpha+\beta+\gamma+\delta)^2} \right) - \frac{(\alpha+\beta)(\gamma+\delta)}{\alpha+\beta+\gamma+\delta} \cdot \mathbf{D^2} \tag{4.37}$$

$$\mathbf{D^2} = |\mathbf{s_{12}} - \mathbf{s_{34}} - \mathbf{h}|^2 \tag{4.38}$$

with α, β, γ and δ being the exponent of the adjoined Gaussian of L1, L2, L3 and L4 respectively.

In order to apply the bipolar approximation to the calculation of the coulomb integrals, we need the two charge distributions being external with respect to each other, meaning that the overlap term $S_{12,34}$ must be very small (see section 4.3.4). The comparison between the $S_{12,34}$ term and a given threshold, is carried on. In a preliminary approximation, then, one could write:

$$S_{12,34} = \frac{3}{4} ln\left(\frac{4(\alpha+\beta)(\gamma+\delta)}{(\alpha+\beta+\gamma+\delta)^2}\right) - \frac{(\alpha+\beta)(\gamma+\delta)}{\alpha+\beta+\gamma+\delta} \cdot \mathbf{D^2} < -TOL \tag{4.39}$$

$$-\frac{(\alpha+\beta)(\gamma+\delta)}{\alpha+\beta+\gamma+\delta} \cdot \mathbf{D^2} < -TOL - \frac{3}{4} ln\left(\frac{4(\alpha+\beta)(\gamma+\delta)}{(\alpha+\beta+\gamma+\delta)^2}\right)$$

$$\mathbf{D^2} > \left[TOL + \frac{3}{4} ln\left(\frac{4(\alpha+\beta)(\gamma+\delta)}{(\alpha+\beta+\gamma+\delta)^2}\right)\right] \cdot \frac{\alpha+\beta+\gamma+\delta}{(\alpha+\beta)(\gamma+\delta)}$$

where $-TOL$ is the natural logarithm of a given overlap threshold, e^{-TOL} (in the code $TOL = inf(41)$, with default value $inf(41)$=14). In the following, $-TOL = s$.

We must remember, however, that the Gaussian functions associated to the overlap distributions are not normalized, and, then, even if their distance **h** is very small, the $S_{12,34}$ value could be considered negligible. For example, given two normalized Gaussian functions $g(L1)$ and $g(L2)$ associated to the L1 and L2 shells respectively, their overlap will be small if $g(L1)$ and a $g(L2)$ are different in position and exponent. The same holds for the L3 and L4 shells. As a consequence the $I_{12,34}$ value of the bielectronic integral depends not only on the distance between the two overlap distribution, but also on the value of S_{12} and S_{34}. For this reason, a corrective term f is added to eq.4.39, considering the charge associated to each overlap distribution:

$$|\mathbf{s_{12}} - \mathbf{s_{34}} - \mathbf{h}|^2 > \left[s + \frac{3}{4} ln\left(\frac{4(\alpha+\beta)(\gamma+\delta)}{(\alpha+\beta+\gamma+\delta)^2}\right)\right] \cdot \frac{\alpha+\beta+\gamma+\delta}{(\alpha+\beta)(\gamma+\delta)} \cdot \left(1 - f\frac{S12+S34}{2(-2.303TOL(1))}\right) \tag{4.40}$$

so that the larger the charge associated to the distribution, the more severe the tolerance. f is a percentage factor, optimized for coulomb integrals to 0.7 (-$\frac{inf(45)}{100}$ in the code, with $inf(45)$ being a parameter set to 70). In this way, the error introduced in evaluating terms involving large charges are minimized, whereas the approximation on terms involving small charges is less severe and the error, relatively large, is very small if compared to the total result of the integral.

Tab.4.19 summarizes the code instructions related to the selection of the **h** vectors. Note that from instruction number 419 in tab.4.19 we get the distance between the two centroids of the two electron distribution:

$$\frac{\delta}{\gamma+\delta}[x(L3)-x(L4)-x(\mathbf{g}')]+\frac{\beta}{\alpha+\beta}[x(L2)-x(L1)+x(\mathbf{g})]-[x(L3)-x(L1)]= \quad (4.41)$$

$$=\frac{\delta x(L3)}{\gamma+\delta}-\frac{\delta x(L4)}{\gamma+\delta}-\frac{\delta x(\mathbf{g}')}{\gamma+\delta}+\frac{\beta x(L2)}{\alpha+\beta}-\frac{\beta x(L1)}{\alpha+\beta}+\frac{\beta x(\mathbf{g})}{\alpha+\beta}-\frac{\delta x(L3)+\gamma x(L3)}{\gamma+\delta}+\frac{\alpha x(L1)+\beta x(L1)}{\alpha+\beta}=$$

$$=\frac{\alpha x(L1)+\beta(x(L2)+x(\mathbf{g}))}{\alpha+\beta}-\frac{\gamma x(L3)+\delta(x(L4)+x(\mathbf{g}'))}{\gamma+\delta}=$$

$$=\mathbf{s_{12}}-\mathbf{s_{34}} \quad (4.42)$$

Similar considerations can be done for the exchange series ($< L1L3|L2L4 >$ and $< L1L4|L2L3 >$, s=$inf(43)$=10 by default and f=$inf(44)/100$, $inf(44)$=90 by default), for which the distance between the two charge distributions is given by $|\mathbf{g}|$, so that equation 4.40 becomes:

$$|\mathbf{s_{13}}-\mathbf{s_{24}}-\mathbf{g}|^2>\left[s+\frac{3}{4}ln\left(\frac{4(\alpha+\gamma)(\beta+\delta)}{(\alpha+\beta+\gamma+\delta)^2}\right)\right]\cdot\frac{\alpha+\beta+\gamma+\delta}{(\alpha+\gamma)(\beta+\delta)}\cdot\left(1-f\frac{S13+S24}{2(-2.303TOL(1))}\right) \quad (4.43)$$

and

$$|\mathbf{s_{14}}-\mathbf{s_{23}}-\mathbf{g}|^2>\left[s+\frac{3}{4}ln\left(\frac{4(\alpha+\delta)(\beta+\gamma)}{(\alpha+\beta+\gamma+\delta)^2}\right)\right]\cdot\frac{\alpha+\beta+\gamma+\delta}{(\alpha+\delta)(\beta+\gamma)}\cdot\left(1-f\frac{S14+S23}{2(-2.303TOL(1))}\right) \quad (4.44)$$

4.3.6 The fourth DO-loop and the selection of bipolar approximated integrals

In the fourth DO-loop of the chain (see scheme 4.2), running on the couple sets (MVLU) for the second electron, some of the quantities to be used for the selection of vectors for which the integrals can be approximated with a bipolar expansion are calculated and stored. In tables 4.12 and 4.13 the instructions are shown and discussed (from number 137 to 208). It has to be noticed that the actual implemented selection formula does not corresponds exactly to the one obtained in section 4.3.5. It takes the following form for coulomb terms (and an analogous form for exchange terms):

$$|\mathbf{s_{12}}-\mathbf{s_{34}}-\mathbf{h}|=\left(\sqrt{\frac{1}{(\alpha+\beta)(\gamma+\delta)}}+\frac{1}{2}\cdot\frac{\alpha+\beta+\gamma+\delta}{(\alpha+\beta)(\gamma+\delta)}\right)\cdot s\cdot\left(1-f\frac{S_{12}+S_{34}}{-2\cdot 2.303TOL(1)}\right) \quad (4.45)$$

Two IF conditions split the instructions into two blocks: for a given set of L1, L2, L3 and L4 shells, the first block is executed if the **COULOMB**

SHELLE		
a1	IDICOU=INF(40)	Maximum order of multipoles for coulomb, default 4
a2	IDIPEX=INF(42)	Maximum order of multipoles for exchange, default 2
a3	ADICOU=IDICOU+1	
a4	T7CO=INF(41)	Tolerance for penetration between the two overlap distributions for coulomb, def 14
a5	T7EX=INF(43)	Tolerance for penetration between the two overlap distributions for exchange, default 10
a6	PINF44=-INF(44)*0.01_FLOAT	Parameter for exchange, default INF(44)=90
a7	PINF45=-INF(45)*0.01_FLOAT	Parameter for coulomb, default INF(45)=70
VIC2		
a8	X1OLD=XLOLD(1,L1)	Coordinates of the L1 shell
a9	X2OLD=XLOLD(1,L2)-X1OLD	$x(L2) - x(L1)$
a10	X21OLD=XGOLD(1,MG)+X2OLD	$x(L2) - x(L1) + x(\mathbf{g})$
a11	EX2OLD=EXAOLD(L2)	β
a12	EX1OLD=EXAOLD(L1)	α
a13	DDDDDD=EX1OLD+EX2OLD	$\alpha + \beta$
a14	SIG12=1._FLOAT/DDDDDD	$\frac{1}{\alpha+\beta}$
a15	SX12=SCOST(MVLA)	$0.75ln\left(\frac{4\alpha\beta}{(\alpha+\beta)^2}\right)$
a16	FA2OLD=EX2OLD*SIG12	$\frac{\beta}{\alpha+\beta}$
a17	F12OLD=EX1OLD*FA2OLD	$\frac{\alpha\beta}{\alpha+\beta}$
CLASSS		
a18	RSQOLD=X21OLD**2+Y21OLD**2+Z21OLD**2	$[x(L2) - x(L1) + x(\mathbf{g})]^2 + [y(L2) - y(L1) + y(\mathbf{g})]^2 + [z(L2) - z(L1) + z(\mathbf{g})]^2 = \lvert\mathbf{s}(L1) - \mathbf{s}(L2) - \mathbf{g}\rvert^2$
a19	XPOLD=X21OLD*FA2OLD	$[x(L2) - x(L1) + x(\mathbf{g})] \cdot \frac{\beta}{\alpha+\beta}$
VIC3		
a20	EX3OLD=EXAOLD(L33)	γ
a21	EX4OLD=EXAOLD(L44)	δ
SHELLXN		
119	S12=SX12-F12OLD*RSQOLD	$0.75ln\left(\frac{4\alpha\beta}{(\alpha+\beta)^2}\right) - \frac{\alpha\beta}{\alpha+\beta}\lvert\mathbf{s}(L1) - \mathbf{s}(L2) - \mathbf{g}\rvert^2$
157	SIG34=1._FLOAT/(EX3OLD+EX4OLD)	$\frac{1}{\gamma+\delta}$
158	FA4OLD=EX4OLD*SIG34	$\frac{\delta}{\gamma+\delta}$
159	F34OLD=EX3OLD*FA4OLD	$\frac{\gamma\delta}{\gamma+\delta}$
161	BILBO=SCOST(MVLU)+S12	$0.75log\left(\frac{4\gamma\delta}{(\gamma+\delta)^2}\right)$+S12
162	T8CO=(SQRT(SIG12*SIG34)+ +(SIG12+SIG34)*0.5_FLOAT)*T7CO	$\left[\sqrt{\frac{1}{(\alpha+\beta)(\gamma+\delta)}} + \frac{1}{2}\left(\frac{1}{\alpha+\beta} + \frac{1}{\gamma+\delta}\right)\right]\cdot$inf(41)
163	FRODO=PINF45*T8CO	$-\frac{inf(45)}{100}\left[\sqrt{\frac{1}{(\alpha+\beta)(\gamma+\delta)}} + \frac{1}{2}\left(\frac{1}{\alpha+\beta} + \frac{1}{\gamma+\delta}\right)\right]\cdot$inf(41)
164	X7=XLOLD(1,L44)-XLOLD(1,L33)	$x(L4) - x(L3)$
	RR=((X7+XGOLD(1,I))**2+(Y7+XGOLD(2,I))**2+ +(Z7+XGOLD(3,I))**2)	$\lvert\mathbf{s}(L3) - \mathbf{s}(L4) - \mathbf{g}'\rvert^2$
	S12+S34=BILBO-(RR)*F34OLD	S12+$\left[0.75log\left(\frac{4\gamma\delta}{(\gamma+\delta)^2}\right) - \frac{\gamma\delta}{\gamma+\delta}\lvert\mathbf{s}(L3) - \mathbf{s}(L4) - \mathbf{g}'\rvert^2\right]$
171	VDID=MIN((S12+S34)*CCCFAJ,1._FLOAT)	$\frac{S12+S34}{2T_1}$
172	IDIPC_VRS(KSH)=MAX(0,IDICOU-INT(ADICOU*VDID))	Maximum order of multipoles
173	VDID_VRS(KSH)=VDID*FRODO+T8CO	T8CO*$\left[1 - \frac{inf(45)}{100}\frac{S12+S34}{2T_1}\right]$
220	X31OLD=XLOLD(1,L3)-X1OLD	$x(L3) - x(L1)$
226	X41OLD=XLOLD(1,L4)-X1OLD	$x(L4) - x(L1)$
236	XQOLD=XPOLD-X31OLD	$\frac{\beta}{\alpha+\beta}[x(L2) - x(L1) + x(\mathbf{g})] - [x(L3) - x(L1)]$
318	IF(LCOUL)THEN	If the coulomb integral must be calculated
319	IDIPC=IDIPC_VRS(KSH)	
320	VDID=VDID_VRS(KSH)	
385	X9OLD=XGOLD(1,MGP)	$x(\mathbf{g}')$
391	X6OLD=X41OLD+X9OLD	$x(L4) - x(L1) + x(\mathbf{g}')$
419	XX=(X31OLD-X6OLD)*FA4OLD+XQOLD	$\frac{\delta}{\gamma+\delta}[x(L3) - x(L4) - x(\mathbf{g}')] + \frac{\beta}{\alpha+\beta}[x(L2) - x(L1) + x(\mathbf{g})] - [x(L3) - x(L1)]$
	LH=(XX-XGOLD(1,N))**2+(YY-XGOLD(2,N))**2+ +(ZZ-XGOLD(3,N))**2	
425	IF(VDID.LT.(LH)THEN	

Table 4.19. Instructions related to the selection of the **h** vectors appearing in the bipolar expansion. Instructions appearing in *SHELLE*, *VIC2*, *CLASSS* and *VIC3* subroutines are enumerated from from a1 to a21. The other numbers refer to the full set of *SHELLXN* instructions in tables 4.10 to 4.17. Some lines are not enumerated: they are used to divide long instructions into smaller and easier ones. When the same operation is repeated for the three cartesian coordinates, only the x instruction is shown in the present table.

terms have to be calculated (LSTA34=.TRUE.; see instructions from 156 to 175 in tab.4.12) and the second one if the **EXCHANGE** terms are present (NPF22=.TRUE.; see instructions from 176 in tab.4.12 to 208 in tab.4.13). The two blocks have a similar structure, and **T8CO** (coulomb), **T8EX** and **U8EX**. (exchange $< L1L3|L2L4 >$ and $< L1L4|L2L3 >$) are calculated:

$$T8CO = \left[\sqrt{\frac{1}{(\alpha+\beta)(\gamma+\delta)}} + \frac{\alpha+\beta+\gamma+\delta}{2(\alpha+\beta)(\gamma+\delta)}\right] \cdot s \tag{4.46}$$
$$T8EX = \left[\sqrt{\frac{1}{(\alpha+\gamma)(\beta+\delta)}} + \frac{\alpha+\beta+\gamma+\delta}{2(\alpha+\gamma)(\beta+\delta)}\right] \cdot s$$
$$U8EX = \left[\sqrt{\frac{1}{(\alpha+\delta)(\beta+\gamma)}} + \frac{\alpha+\beta+\gamma+\delta}{2(\alpha+\delta)(\beta+\gamma)}\right] \cdot s$$

In the coulomb block, after T8CO is estimated, the maximum order of multipoles for each star of the L3L4 couple, **IDIPC_VRS**, is calculated, and the right term of equation 4.45, **VDID_VRS**, is evaluated. Both these vectors have ISTA34 size, where ISTA34 is the maximum number of stars to be considered for the L3L4 couple (the first vector of each star is selected for the calculation of the L3L4**g'** overlap).

- VDID=min $\left(\frac{S_{12}+S_{34}}{-2\cdot 2.303 TOL(1)},1\right)$, where S_{12} and S_{34} are the natural logarithm of the overlap distribution of L1L2 and L3L4 respectively (see tab.4.12 and section 4.2.1). Note that VDID could be equal to 1 when $S_{12}+S_{34} \leq -2\cdot 2.303 TOL(1)$, and so when the overlaps $e^{S_{12}}$ and $e^{S_{34}}$ are both smaller than or equal to $10^{-TOL(1)}$. However, the case with $e^{S_{12}}$ and $e^{S_{34}}$ smaller than $10^{-TOL(1)}$ is never verified, as such couple sets are disregarded (see sections from 3.1 to 3.6: selection of the couples and of the **g** stars).

- **IDIPC_VRS**=max{0,$[inf(40)-(inf(40)+1)\cdot VDID]$}, where inf(40) is set to 4 by default (see section 4.2.1). A **IDIPC_VRS** element equal to zero means that only the charge is used in the bipolar expansion. This happens when VDID is greater or equal to $\frac{inf(40)}{inf(40)+1}$, and so when the overlap ρ_{12} and ρ_{34} are small. In particular, if the default value 4 is used for $inf(40)$, IDIPC_VRS(KSH)=0 when VDID$\geq$0.8

- **VDID_VRS**: the right term of equation 4.45.

For the exchange terms, only T8EX (and U8EX when LPESO is true, two couples related by hermiticity) is calculated, whereas the other quantities for

the selection are estimated later on in the *CLALLL* and *CLAKKK* subroutines (DA FINIRE, see section XX).

In the case of the exchange series the size of the blocks associated to each electron in the case of a bipolar approximation is calculated; it is given by the product of the number of atomic orbitals for each shell by the factor $(inf(42)+1)^2$, where $inf(42)$ is the maximum number of multipoles. Consider for example the exchange integral $< \lambda_p\lambda_d|\lambda_s\lambda_d >$, where d, p and s indicate $d-$ $p-$ and $s-$ type shells. The block corresponding to the first electron will have $NPOL13 = 3\cdot5\cdot(2+1)^2 = 15\cdot9 = 135$ size, and the one corresponding to the second electron will have $NPOL24 = 1\cdot5\cdot(2+1)^2 = 45$ size ($inf(42) = 2$ is the default value for the exchange).

4.3.7 Testing the bipolar approximation

The selection formula for the bipolar approximation implemented in the CRYSTAL code, different from the one that can be obtained by theory, and the s and f parameters were optimized for systems containing elements of the first three rows of the periodic table, such that the error introduced by the approximation were in the order of 10^{-5} Ha on the total energy. It has been recently noticed that the error introduced in the total energy by using the implemented selection formula increases with the number of core electrons of the elements of the system, so that it is better performing an "exact" calculation (NOBIPOLA[4] keyword) rather than applying the bipolar approximation.

In the case of ZrO_2, for example, the structure optimized with and without the bipolar approximation differ by about 10^{-2} Å on the cell parameters, 10^{-3} on fractional coordinates and 10^{-4} Ha on the total electronic energy. The situation is much more critical for systems containing heavier metals, that need to be optimized with the exact calculation of both coulomb and exchange terms. If we try, for example, to optimize La_2O_3 with the old bipolar approximation, the minimum of the energy surface in not reached, and the optimization points oscillates from one position to another one around the equilibrium structure. (PROVARE A OTTIMIZZARE CON LA NUOVA FORMULA!)

For all these reasons, several tests have been performed on a set of oxides of various elements belonging to different lines of the periodic table: BeO (second line), SiO_2 (third line), GeO_2 (fourth line), ZrO_2 (fifth line), La_2O_3 and PbO_2 (sixth line), UO_2 (seventh line); along the same line, tests have also

been carried out on C-diamond and two-dimensional C-graphite.

We show again the two expressions for the selection of vectors for which the bipolar approximation can be applied (see sections 4.3.4 and 4.3.5): in this section we will indicate as "new" all results obtained with the selection formula

$$|\mathbf{s_{12}} - \mathbf{s_{34}} - \mathbf{h}|^2 > \left[s + \frac{3}{4} ln\left(\frac{4(\alpha+\beta)(\gamma+\delta)}{(\alpha+\beta+\gamma+\delta)^2}\right)\right] \cdot \frac{\alpha+\beta+\gamma+\delta}{(\alpha+\beta)(\gamma+\delta)} \cdot \left(1 - f\frac{S12+S34}{2(-2.303TOL(1))}\right) \quad (4.47)$$

and as "old" the ones obtained with the selection formula

$$|\mathbf{s_{12}} - \mathbf{s_{34}} - \mathbf{h}|^2 > s \cdot \left[\sqrt{\frac{1}{(\alpha+\beta)(\gamma+\delta)}} + \frac{\alpha+\beta+\gamma+\delta}{2(\alpha+\beta)(\gamma+\delta)}\right] \cdot \left(1 - f\frac{S12+S34}{2(-2.303TOL(1))}\right) \quad (4.48)$$

Remember that analogous formula are obtained for the exchange terms. The s parameter corresponds to $inf(41)$ for coulomb and to $inf(43)$ for exchange selection; f corresponds to $0.01 \cdot inf(45)$ and $0.01 \cdot inf(44)$ for coulomb and exchange selection respectively.

The new formula has been implemented for both coulomb and exchange selection in a developing version of the CRYSTAL06 code (cvs17sept2007) and can be activated by the input keyword BIPONEW. We remember that, if nothing is indicated, the old selection formula is used, whereas the exact calculation can be performed with the NOBIPOLA[4] keyword. The "old" and "new" expressions have been tested on the selected set of systems and results have been compared to the exact calculation. Table 4.20 shows the systems that have been used for the present study, with their main features and the conditions in which the calculations have been performed.

The coulomb series The calculations have been performed by setting $inf(43)$ such that the exchange integrals are calculated exactly, so that differences between the approximated and the exact calculations are due to coulomb integrals only.

The effect of f and s parameters on the total electronic energy and the number of integrals to be calculated exactly has been investigated, and results summarized in tables from 4.22 to 4.24. Differences obtained with the default values for f and s are summarized in table 4.21.

Let us consider, first, the effect of the f value ($0.01 \cdot inf(45)$), that can range from 0.0 to 1.0 and represents the fraction of the charge term that has to be considered. The default value, optimized for the old formula and for elements

system	SGN	Z	el.	AN	TOLDEE	INTGPACK	TOLINTEG	OPT
BeO	186	2	24	4	8	A	6 6 6 6 12	OLD+NEW
C-graphite	77 (2D)	2	12	6	8	A	5 5 5 5 12	-
C-diamond	227		12	6	6	A	5 5 5 5 12	-
SiO_2	154	3	90	14	8	A+P	6 6 6 6 12	OLD
GeO_2	136	2	96	32	8	A+P	6 6 6 6 12	NOBIPOLA
ZrO_2	137	2	112	40	8	A+P	6 6 6 6 12	OLD+NEW
La_2O_3	164	1	138	57	8	A+P	7 7 7 7 14	NOBIPOLA
PbO_2	60	4	376	82				
UO_2	225	1	108	92				

Table 4.20. Systems considered in the present study. AN: atomic number (oxygen AN never shown); SGN: space group number; Z: number of formula unit per cell; el.: number of electron per cell; TOLDEE: tolerance on the SCF; INTGPACK: packages by which integrals are calculated, A=ATMOL and P=POPLE; TOLINTEG: tolerances in evaluating integrals; OPT: old/new or no approximation on bielectronic integrals for the optimization; OLD+NEW: the structure has been optimized both with the old and the new approximations.

system	Δ_{old}	Δ_{new}
BeO	5.2E-05	3.2E-06
C-graphite	-2.3E-06	-1.1E-05
C-diamond	8.8E-05	3.6E-06
SiO_2	3.5E-05	-9.6E-05
GeO_2	1.7E-04	-9.1E-05
ZrO_2	3.2E-03	-2.0E-05
La_2O_3	2.4E-03	-2.6E-04
PbO_2		
UO_2		

Table 4.21. Absolute energy difference (Δ; [Ha]) between the bipolar approximated calculation with the old (and new) formula and the exact calculation. Note that the error increase with the atomic number (or core electrons) of the elements.

of the first three periodic table lines, is 0.7. In the present study, values ranging from 0.5 to 1.0 (with step 0.1) have been considered: in 27 out of 36 cases the electronic energy difference between the exact and new-approximated calculations is smaller than the difference between the exact and old-approximated calculations (see table 4.22). Systems containing high atomic number elements, such as zirconium and lantanium, give electronic energy differences in the order of 10^{-3} Hartree for the old case, whereas the difference decreases to 10^{-4} (La) and 10^{-5} (Zr) when the new formula is applied. Differences due to the f

parameter are in the order of about 10^{-5} for the considered systems, and then the difference between the exact and the approximated calculations cannot be attributed to the effect of this parameter.

f	BeO		C-graphite		C-diamond		SiO_2	
	Δ_{old}	Δ_{new}	Δ_{old}	Δ_{new}	Δ_{old}	Δ_{new}	Δ_{old}	Δ_{new}
0.5	4.9E-5	2.6E-5	-7.0E-6	-1.3E-5	9.1E-5	-3.8E-6	7.4E-5	-7.7E-5
0.6	5.4E-5	2.7E-5	-4.3E-6	-1.1E-5	8.7E-5	-3.4E-6	6.5E-5	-9.1E-5
0.7*	5.2E-5	3.2E-5	-2.3E-6	-1.1E-5	8.8E-5	3.6E-6	3.5E-5	-9.6E-5
0.8	5.2E-5	3.2E-5	-4.5E-6	-6.8E-6	1.0E-4	-7.3E-7	4.7E-5	-8.7E-5
0.9	5.3E-5	3.1E-5	-1.0E-5	-7.2E-6	1.1E-4	2.0E-5	8.3E-5	-3.9E-5
1.0	5.3E-5	3.2E-5	-8.2E-6	-1.4E-5	1.0E-4	-1.3E-5	6.3E-5	-3.2E-5
	ZrO_2		La_2O_3		PbO_2		UO_2	
	Δ_{old}	Δ_{new}	Δ_{old}	Δ_{new}	Δ_{old}	Δ_{new}	Δ_{old}	Δ_{new}
0.5	3.3E-3	-1.6E-5	2.9E-03	2.6E-04				
0.6	3.3E-3	-1.6E-5	2.9E-03	2.6E-04				
0.7*	3.2E-3	-2.0E-5	2.4E-03	-2.6E-04				
0.8	3.3E-3	-2.5E-5	2.9E-03	2.6E-04				
0.9	2.3E-3	-1.8E-5	2.9E-03	2.5E-04				
1.0	2.3E-3	-2.8E-6	2.8E-03	2.5E-04				

Table 4.22. Energy difference (Δ_{old} and Δ_{new}, [Ha]) between the old-/new-approximated and the exact coulomb terms calculation for different values of the f parameter; *:default.

The effect of f on the total number of coulomb integrals to be calculated exactly at each SCF-cycle has been checked. In table 4.23 the exact coulomb integral percentage is shown, and it increases [from a minimum of 7 (C-diamond) to a maximum of 12% (BeO)] when passing from f=1.0 to f=0.5, indicating that the cost of the calculation does not increase dramatically. No relevant differences has been observed, for the considered systems, between the old and the new calculation costs (new is always 1-2% higher).

We consider, now, the effect of the s parameter ($inf(41)$); results are shown in table 4.24. The difference between the bipoalr approximated calculations and the exact calculation decreases when s is increased. As regards the old/new selection, it is observed that the new formula is more efficient than the old one in 28 out of 40 cases. In particular, for all the considered systems, differences become in the order of 10^{-6} Ha when s is set to 18 in the new formula. Note that the error on the lantanium oxide total energy is in the order of 10^{-4} Ha even when s=18 in the old formula.

f	BeO		C-graphite		C-diamond		SiO_2	
	C^{ex}_{old}%	C^{ex}_{new}%	C^{ex}_{old}%	C^{ex}_{new}%	C^{ex}_{old}%	C^{ex}_{new} %	C^{ex}_{old} %	C^{ex}_{new} %
0.5	25	26	43.5	45	22	24	22	23
0.6	23	24	41.5	43	21	23	20	21
0.7*	20.5	21.5	39.5	41	20	22	18	18.5
0.8	18.5	19	37	39	18.5	20	15.5	16.5
0.9	16	16.5	35	37	17.5	19	13.5	14
1.0	14	14	33	34	15	17	11	12

Table 4.23. Relative amounts of coulomb integrals to be calculated exactly at each SCF-cycle for different systems in the case of the old (C^{ex}_{old})and new (C^{ex}_{new}) approximations: effect of the f parameter. *:default.

s	BeO		C-diamond		SiO_2		ZrO_2		La_2O_3	
	Δ_{old}	Δ_{new}	Δ_{old}	Δ_{new}	Δ_{old}	Δ_{new}	Δ_{old}	Δ_{new}	Δ_{old}	Δ_{new}
12	2.6E-4	1.6E-4	9.3E-5	-7.1E-6	7.1E-4	3.8E-5	7.9E-3	1.3E-3	4.2E-2	2.3E-2
13	1.2E-4	7.4E-5	1.5E-4	-3.3E-6	3.4E-4	6.7E-7	2.5E-3	7.1E-4	1.6E-2	9.3E-3
14*	5.3E-5	3.2E-5	8.8E-5	-3.6E-6	3.5E-5	-9.6E-5	3.2E-3	-2.0E-5	2.4E-3	-2.6E-4
15	3.9E-5	6.7E-6	-6.4E-6	-1.4E-5	1.4E-4	6.6E-6	7.5E-4	-1.2E-4	2.0E-3	1.8E-4
16	6.5E-6	-5.0E-6	-1.4E-5	-3.6E-5	-2.5E-5	-1.5E-4	9.8E-4	-6.5E-5	2.1E-3	3.0E-5
17	1.2E-6	-3.0E-6	5.3E-6	-4.5E-6	-1.9E-5	-8.4E-5	3.8E-4	-2.0E-5	7.0E-4	-6.2E-6
18	2.4E-7	-2.6E-6	-3.2E-6	-8.4E-6	8.8E-6	8.9E-6	-1.1E-5	-1.6E-5	1.4E-4	-3.0E-6
30	-6.1E-7	-5.2E-7	1.9E-7	9.8E-7	5.8E-6	4.4E-6	-3.3E-6	-3.4E-6	5.4E-6	2.4E-6
10^3	-2.0E-11	-2.0E-11	0.0	0.0	4.0E-10	4.0E-10	0.0	0.0	0.0	0.0
10^4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	C^{ex}_{old}%	C^{ex}_{new}%	C^{ex}_{old}%	C^{ex}_{new}%	C^{ex}_{old}%	C^{ex}_{new} %	C^{ex}_{old} %	C^{ex}_{new} %	C^{ex}_{old} %	C^{ex}_{new} %
12	17	17	17	19	14.5	15	18	19	12	13
13	18.5	19	18	20	16	17	20	21	13.5	14.5
14*	20.5	21.5	20	22	18	18.5	22	23	15	16
15	22.5	23.5	21	23	19	20	24	25	16	17
16	24.5	25.5	22	24.5	21	22	26	27	17.5	19
17	27	28	23.5	25.5	22.5	24	28	29.5	19	20
18	29	30	24.5	27	24	25.5	30	31.5	20	22
30	51	52.5	42.5	47	41.5	44	52	53.5	37	40
10^3	99.998	99.998	100	100	99.95	99.97	99.999	99.999	99.5	99.5
10^4	100	100	100	100	100	100	100	100	100	100

Table 4.24. Effect of the s patameter on electronic energy difference, Δ and coulomb term to be calculated exactly, C^{ex}. *:default.

The exchange terms The scheme applied for investigating the effect of the f and s parameters on the coulomb series approximation has been utilized for the exchange series; remember that, now, f=0.01·$inf(44)$ and s=$inf(43)$. Results have been obtained by setting $inf(41)$ such that the coulomb terms are calculated exactly and differences between the approximated and the exact calculations are due to exchange terms only. Note that the present calculations have been performed with the B3LYP hybrid Hamiltonian, with 20% of exact HF exchange.

The effect of both parameters is shown in table 4.25 (compare to table 4.22). f does not influence significantly the total energy, and the old formula is slightly better than the new one in 16 out of 24 cases. Energy differences

decreases when s increases, and, as for the coulomb case, a difference value in the order of 10^{-6} Ha is obtained for all the studied systems with s=18; the old formula gives better results in 15 out of 32 cases.

f	C-diamond		SiO_2		ZrO_2		La_2O_3	
	Δ_{old}	Δ_{new}	Δ_{old}	Δ_{new}	Δ_{old}	Δ_{new}	Δ_{old}	Δ_{new}
0.5	5.9E-6	6.8E-6	-9.0E-6	6.2E-6	-5.7E-4	-6.5E-4	-5.6E-4	-5.9E-04
0.6	9.3E-6	9.2E-6	-9.8E-6	7.1E-6	-5.6E-4	-6.5E-4	-5.2E-4	-5.5E-04
0.7	9.4E-6	1.0E-5	-1.3E-5	-1.0E-6	-5.7E-4	-6.4E-4	-5.2E-4	-5.5E-04
0.8	8.9E-6	1.0E-5	-1.3E-5	-2.4E-6	-5.8E-4	-6.4E-4	-5.2E-4	-5.6E-04
0.9*	5.4E-6	8.8E-6	-1.5E-5	-3.3E-6	-5.6E-4	-6.4E-4	-5.2E-4	-5.6E-04
1.0	2.2E-6	5.5E-6	-7.2E-6	-5.6E-6	-5.6E-4	-6.5E-4	-5.3E-4	-5.6E-04
s								
10*	5.4E-6	8.8E-6	-1.5E-5	-3.3E-6	-5.6E-4	-6.4E-4	-5.2E-4	-5.6E-4
12	1.2E-5	2.3E-6	-2.9E-5	-3.2E-5	-3.0E-4	-3.1E-4	-3.1E-4	-2.8E-4
14	-7.1E-7	-3.1E-6	-1.1E-5	-1.4E-5	-6.1E-5	-1.0E-4	-1.3E-4	-1.3E-4
16	-6.1E-6	-6.3E-6	-1.8E-5	-1.5E-5	-8.9E-5	-9.0E-5	-3.0E-5	-1.5E-5
18	-1.5E-6	-1.1E-6	-4.9E-6	-6.3E-6	-1.5E-5	-1.2E-5	-9.9E-6	1.1E-5
20	-3.7E-7	-3.9E-7	-3.0E-6	-1.3E-6	-2.2E-6	-1.5E-6	9.3E-6	8.0E-6
22	2.9E-7	-4.1E-7	9.2E-7	5.4E-8	-1.0E-5	-1.0E-5	1.2E-6	-1.2E-6
24	-4.1E-7	1.0E-7	-1.1E-6	-1.1E-6	4.7E-8	-1.7E-7	2.9E-6	1.4E-6

Table 4.25. Effect of the f and s parameters on the exchange selection: Δ, [Ha], is the difference between the approximated and the exact calculation (B3LYP, 20%HF). *:default.

Conclusions For the f parameter the default values ($inf(45) = 70$ and $inf(44) = 90$) can remain unchanged, whereas the overlap thresholds s ($inf(41)$ and $inf(43)$) have been set to 18. The calculations have been performed in these conditions with the old and the new formula, and with hybrid B3LYP (containing the 20% of HF exchange) and HF Hamiltonians. Results are shown in table 4.26.

4.3.8 A "cut-factor" for the selection formula

It would be interesting, now, maintain the accuracy and reduce furtherly the cost of calculation. In particular, we could try to find a factor, depending on the adjoined Gaussian exponents of the four involved shells, which make the threshold less severe for valence shells, and more severe for core shells. Along

	Δ_{old}^{B3LYP}	Δ_{new}^{B3LYP}	Δ_{old}^{HF}	Δ_{new}^{HF}	$t_{old}^{HF}\%$	$t_{new}^{HF}\%$	$t_{nobipola}^{HF}$
BeO	-3.8E-6	-7.1E-6	-1.0E-5	-1.5E-5	52	53	493
C-diamond	-4.8E-6	-9.5E-6	-1.6E-5	-2.0E-5	61	62.5	53
C-graphite	-3.3E-6	-3.4E-6	-4.3E-6	-4.4E-6	71	71.5	16
SiO_2	3.9E-6	2.6E-6	-1.1E-5	-3.7E-6	48	49	8811
GeO_2	1.5E-4	6.9E-5					
ZrO_2	-2.3E-5	-2.6E-5	-3.2E-5	-6.4E-5	51	52.5	2701
La_2O_3	1.3E-4	7.7E-6	1.4E-4	5.4E-5	45.5	47	2317

Table 4.26. Total energy difference between the approximated (exchange and coulomb) and the exact calculation with B3LYP and HF Hamiltonians: $inf(41)$ and $inf(43)$ are set to 18. The last column shows the total time $t_{nobipola}^{HF}$, [s], required for the exact HF calculation (serial calculation); $t^{HF}\%$ indicates the total time required for the old and new approximated calculations as a fraction of $t_{nobipola}^{HF}$. Note that the bipolar approximation reduces the calculation cost of about 50% (it decreases to 30% with default values).

this line, the "cut" factor can be in the form

$$cf = 2 \cdot [ln(\alpha + \beta) + ln(\gamma + \delta)] \tag{4.49}$$

and added to s term of the selection formula in equation 4.50

$$|\mathbf{s_{12}} - \mathbf{s_{34}} - \mathbf{h}|^2 > \left[s + cf + \frac{3}{4} ln\left(\frac{4(\alpha+\beta)(\gamma+\delta)}{(\alpha+\beta+\gamma+\delta)^2} \right) \right] \cdot \frac{\alpha+\beta+\gamma+\delta}{(\alpha+\beta)(\gamma+\delta)} \cdot \left(1 - f \frac{S12+S34}{2(-2.303TOL(1))} \right) \tag{4.50}$$

Consider, for example, La_2O_3, whose adjoined Gaussian exponents are shown in table 4.27[1]. Consider, then, the case when $\alpha = \gamma$ =248.0800 and $\beta = \delta$ =39.8220: cf =22.6505, $s + cf = s$+22.6505> 30 and the selection threshold becomes very severe. Consider the case when $\alpha = \gamma$ =0.218 and $\beta = \delta$ =0.2537: cf =-3.0056, and the threshold is less severe.

```
EXAOLD          1 =   17.0178000000000  Si
EXAOLD          2 =   1.22543000000000
EXAOLD          3 =  0.302422000000000
EXAOLD          4 =  0.118000000000000
EXAOLD          5 =   2.55850000000000
EXAOLD          6 =  0.538800000000000

EXAOLD          1 =   8.32200000000000  Be
EXAOLD          2 =   2.33500000000000
EXAOLD          3 =  0.664000000000000

EXAOLD          1 =   3.18900000000000
```

[1]In CRYSTAL, at the moment, the adjoined Gaussian of a given shell is defined as the most diffuse Gaussian function of the shell.

ZrO_2		La_2O_3	
106.8939500	1s	248.0800	1s
13.0012280	2sp	39.8220	2sp
1.7492066	3sp	6.8213	3sp
0.7875000	sp*	4.8497	3d
1.9570000	4d	3.0748	4sp
0.3384800	sp*	1.5540	4d
0.9959000	d*	0.6145	d*
0.4135440	d*	1.3323	5sp
-	-	0.5917	sp*
-	-	0.2537	sp*
1.212	1s	1.212	1s
1.217	2sp	1.217	2sp
0.500	sp*	0.500	sp*
0.218	sp*	0.218	sp*
0.500	d*	0.500	d*

Table 4.27. Adjoined Gaussian exponents for ZrO_2 and La_2O_3. The five exponents in the second part of the table belong to oxygen. *: diffuse polarization functions.

	B3LYP		HF			
	Δ_{13}	Δ_{14}	Δ_{13}	$t_{13}\%$	Δ_{14}	t_{14}
ZrO_2	-3.4E-4	-2.6E-4	-4.7E-4	43	-1.9E-4	50.5
La_2O_3	-2.5E-5	2.4E-6	-1.6E-4	42	6.4E-5	43.5

Table 4.28.

```
EXAOLD          2 =   4.09550000000000
EXAOLD          3 =  0.917300000000000
EXAOLD          4 =  0.261200000000000
EXAOLD          5 =  0.931200000000000
```

4.3.9 Selection and calculation of the coulomb terms

It is useful, now, split the subroutine into two parts, in order to describe the implementation of the selection for Coulomb and Exchange terms and of their calculation. We begin, then, the discussion about the coulomb integrals and the related instructions in the three last DO-loop of the *SHELLXN* subroutine.

We must come back, first, to the *SHELLD* routine, in section 4.2.2: for all the N translation vectors selected in the *GCALCO* subroutine, a logical variable, KPESO(N)=.TRUE., and the weight of an integral, WEIGHT(N)=1, are set and stored in the *PESVRS* module.

If the coulomb integral for a given set of L1L2 and L3L4 mother couples must be calculated (LSTA34=.TRUE.), instructions from 232 to 248 of

tab.4.13 are executed (in the fifth DO-loop). Instructions from 233 to 238 corresponds to combination of exponents and coordinates, used later in the integral calculation.

The following quantities are here calculated:

- INZILA(L3), INZILA(L4): the starting point of the **h** and **h'** vectors for L3 and L4 respectively, see section 4.3.3, for the coulomb integrals $< L1L2^g|L3^hL4^{h+g'} >$ and $< L1L2^g|L4^{h'}L3^{h'-g'} >$
- ILTUTLVRS: total number of **h'** vectors, $\{\mathrm{H}\}_{L4}$
- ILTUTK: total number of **h** vectors, $\{\mathrm{H}\}_{L3}$

A short DO-loop on $\{\mathrm{H}\}_{L3}$ (I=1,ILTUTK) stores in the **ILW** array all the **h** vectors, namely N in the code (N=ILA(INZILA(L3)+I), and, for each of them, set the variable KPESO(N)=.FALSE., indicating that the N vectors belong only to $\{\mathrm{H}\}_{L3}$ (for the other translation vectors KPESO is true).

In the sixth DO-loop (301 in tab.4.14), running on the maximum number of **g'** stars between the number of stars for coulomb and exchange integrals (KSH=1,NSHTOT), a certain mumber of logical variables are defined for the coulomb terms:

- IMMUM=L3GEL4.AND.KSH.LE.ISTA34 (306): the L3L4 couple is in the low-half part of the matrix and the KSH star in smaller than the maximum star to be considered for the coulomb terms
- IMMUMBI=IMMUM.AND.BIPOC_ACTIVE (307): IMMUM is true and the bipolar approximation is never performed for the coulomb integrals.
- LCOUL=(ILTUTK+ILTUTL).NE.0 (317): it is true when the sum of the number of **h** and **h'** vectors is non-null.

If IMMUM is true, two cases are possible:

- ILTUTL=0 if L3=L4 (diagonal couple) and we are in the first star (so that the module of the **g'** (or **g"**) translation vector is null: $\sigma^{\mathbf{h}}\sigma^{\mathbf{h+0}}$). In this case the $\{\mathrm{H}\}_{L3}$ set is equal to the $\{\mathrm{H}\}_{L4}$ one, so that we need just one {H} set.
- ILTUTL=ILTUTVRS in all the other cases

If LCOUL is true, the IDIPC_VRS(KSH) and VDID_VRS(KSH) values are assigned to IDIPC and VDID, see sections 4.3.6 and 4.3.5. If IMMUM is false, but the exchange terms must be calculated (MOTUHF=.TRUE.) LCOUL is set to .FALSE. and the coulomb blocks after that will be disregarded; if it is true and the exchange terms must not be calculated, the loop cycles.

In the seventh DO-loop, running on the **g'** vectors (MGP=NQGSHG(NL) see the instructions 309, 310, 331 and 333 in tab.4.14), several blocks of instructions (sub-loops or IF-conditions) regards only the coulomb term calculation.

At first, we discuss the block going from number 363 to 384 in tab.4.15. The logical variable L3EQL4 is verified in the diagonal case, when L3 is equal to L4. For a diagonal couple, the **g'** stars (sixth DO-loop) are the ones of the $GCALCO$ set, as $(\mathbf{s}_3 - \mathbf{s}_4) = 0$. In each star, we have both the **g'** and the -**g'** vectors to be considered, and we need, then, to compute the integrals:

$$< \mu^0 \nu^{\mathbf{g}} | \sigma^{\{\mathbf{h}\}_\sigma} \sigma^{\{\mathbf{h}\}_\sigma + \mathbf{g}'} > \cdot P_{33}^{0\mathbf{g}'} \tag{4.51}$$

$$< \mu^0 \nu^{\mathbf{g}} | \sigma^{\{\mathbf{h}\}_\sigma} \sigma^{\{\mathbf{h}\}_\sigma - \mathbf{g}'} > \cdot P_{33}^{0\ -\mathbf{g}'} \tag{4.52}$$

Both the P blocks are available; however, we can transform the second set as follows:

$$< \mu^0 \nu^{\mathbf{g}} | \sigma^{\{\mathbf{h}\}_\sigma - \mathbf{g}'} \sigma^{\{\mathbf{h}\}_\sigma} > \cdot P_{33}^{-\mathbf{g}'\ 0} \tag{4.53}$$

and $P_{33}^{-\mathbf{g}'\ 0} = P_{33}^{0\ \mathbf{g}'}$ as in eq.4.51. In the seventh DO-loop, then, we consider only half of the $\mathbf{g}'$ stars and the others are generated as in eq.4.53. In the code this is done by the following instructions:

```
IF(L3EQL4.AND.MGP.GT.NN1(MGP))THEN
LLCOUL=.FALSE.
ELSE
LLCOUL=LCOUL
...
```

where NN1(MGP) corresponds to the inverse of MGP. LLCOUL will be used later on as a condition to execute or disregard the blocks regarding the coulomb calculation.

DOPO VACANZE: Se immumbi e bipoactive x coul, jcount va avanti a leggere nel buffer i multipoli, che sono ordinati per le coppie e sono di lunghezza Npolj=inf(40)+1 al quadrato. Poi fa ss34 che sono i multipoli per 34, mentre quelli s12 arrivano dal polipo. cercare mout 10 e one side ok dopo le vacanze. :-)

We want to know, now, which of the **h** vectors belonging to $\{H\}_{L3}$ appears also in the $\{H\}_{L4}$ set and have, then, a double weight, and which are the **h'** vectors belonging to the $\{H\}_{L4}$ set and not to $\{H\}_{L3}$. The whole set of **h** and **h'** vectors can then be assigned to the $L3^{\mathbf{h}}L4^{\mathbf{h}+\mathbf{g}'}$ couple.

Two variables are defined:

- ILTUTX: counter for the **h** vectors plus the **h'** vectors not belonging to $\{H\}_{L3}$;
- ILTUTY: counter for the **h'**=**h** vectors that are given WEIGHT=2.

This is done in the IF condition in the seventh DO-loop (instructions 400-415 in tab.4.15) if the coulomb term must be calculated (LLCOUL is true). In the DO-loop on the **h'** vectors (ILTUTL) the IF condition number 408 assigns the **h'** vectors either to the "ILTUTX" or to the "ILTUTY" list: for a given N vector, if KPESO(N)=.TRUE., the N vector has not been included in the $\{H\}_{L3}$ list, and so ILTUTX=ILTUTX+1 and **ILW(ILTUTX)**=N (list of the **h** vectors plus the **h'** vectors not belonging to $\{H\}_{L3}$), if KPESO(N)=.FALSE. the N vector belongs also to the $\{H\}_{L3}$ set, so WEIGHT(N)=2 and **ILV(ILTUTX)**=N (list of the **h**=**h'** vectors belonging both to $\{H\}_{L3}$ and to $\{H\}_{L4}$).

A DO-loop on ILTUTX (422 in tab.4.15) select then the coulomb terms to be calculated exactly and the ones that can be approximated through a bipolar expansion: ILTUTM is a counter for the vectors for which the integral can be approximated, and the vectors are stored in the **MGIPI** array, ILTUTC counts the vectors for the exact calculation, which are stored in the **ILU** array. Further details about the selection of **h** vectors are deeply discussed in section 4.3.5

LLCOUL is then re-set as LLCOUL=ILTUTC.NE.0, meaning that there are one or more **h** vectors for which the integral must be calculated exactly.

If ILTUTM is non-null, there are some vectors for which the integral can be approximated. A trick, that makes the calculation faster is used when such a calculation is performed: two subroutines are called, depending on which **h** vector is considered.

DOPO VACANZE: covuvu e covvvv fanno la somma su h della matrice dei multipoli, con riferimento alla parte che manca in the bipolar expansion section. ISYMD fa il triplo prodotto POLI_12 (sum h Rl1l2) POLI_34 e lo schiaffa

in tt2 che aveva gia previdentemente azzerato dopo il polipo. All the contributions are summed and stored into a matrix, that is summed to itself at each cycle; that is because the whole **h** summation is calculated and stored. Within such a logic, we should first define a null matrix and then sum each term corresponding to a different **h** value. It is faster, however, calculate the matrix for the first **h** vector (the *COVVVV* subroutine) and then, in a DO-loop running from the second to the last **h** vector, sum the other contribution one by one (the *COVUVU* subroutine).

If LLCOUL is true, the exact integral can be calculated either with the POPLE, or with the ATMOL package: the *SHELLJ* and the *VIC5JD* are called, respectively, in the two cases, and the result of the exact coulomb term is stored in RJ (POPLE) or in WW12 (ATMOL) quantities.

4.3.10 The exchange series

In the previou section we have disregarded for a while the exchange series in order to concentrate on the coulomb series. In this section, then, all the blocks of the *SHELLXN* routin regarding the exchange terms are discussed.

In the fifth DO-loop of the chain two logical variable related to the exchange series are defined (instructions number 218-219 in tab.4.13):

- LEXCH4A=NSH13.NE.0.AND.NSH24.NE.0: the translation vector stars for the couples L1L3 and L2L4 are non null;

- MPESO=NSH14.NE.0.AND.NSH23.NE.0.AND.JNCDU(MVLU).NE.1:

We know, from section 4.3.6, that the NPF22 condition is true when the exchange contributions for the L1L2 and L3L4 centers are present: a second large IF condition is executed when NPF22=.TRUE. Inside the IF condition, two sub-blocks, very similar with each other, are executed for $< L1L3|L2L4 >$ and the $< L1L4|L2L3 >$ integrals respectively. The instructions for the first one, controlled by the LEXCH4A condition, are commented in tables 4.13 and 4.14 (from number 250 to 272).

N and ICAME give the starting point of the translation vectors of all the couples containing L3 and L4, respectively. The **INZL12** vector, calculated in the *READEX* routine called a the beginning of *SHELLXN*, indicates the number of multipoles for a given couple, in this case L2L4 and L1L3. The I DO-loop runs on the translation vectors of the couple (**g-h-g' h**

```
IF(MPESO)THEN
...

N=LA34V2(L3)
I1=INZL12(L3,2)
DO I=NGSHG(N+1)+1,NGSHG(N+NSH23+1)
KNVOR(NQGSHG(I))=I1
I1=I1+NPOL23
ENDDO
JCAME=LA34V1(L4)
I1=INZL12(L4,1)
DO I=NGSHG(JCAME+1)+1,NGSHG(JCAME+NSH14+1)
LNVOR(NQGSHG(I))=I1
I1=I1+NPOL14
ENDDO
ENDIF
ENDIF
```

JNVOR e INVOR, LNVOR and KNVOR: vettori per le coppie L1L3 and L2L4, L1L4 and L2L3.
In pratica trova, dati i 4 shell, i vettori delle due coppie nel caso dei coulomb, degli scambi1324 e scambi 1423.

- LEXCH4=KSH.LE.LPF22.AND.LEXCH4A : the order number of the KSH star is smaller than the total number of the stars to be considered for the exchange terms, and LEXCH4A is true (see table **??**).
- LEXCH5=KSH.LE.LPF22.AND.MPESO : as the previous, but for the $< L1L4|L2L3 >$ exchange term.
- MOTUHF=(LEXCH4.OR.LEXCH5) : it is true if LEXCH4 or LEXCH5 are true (se sono vere entrambe??)

If MOTUHF is true, indicating that one or both the exchange contributions for the shells L1, L2, L3 and L4 exist, LCOUL is set to .FALSE., else the loop exit, the following KSH star is considered and JPETAL, indicating the starting point of irreducible vectors in NNGI for the mother couple L3L4, is increased by the number of irreducible vectors for the given couple in the KSH-1 star.

Then, the MOTUHF variable is re-set:
MOTUHF=(.NOT.(LEXCH4.OR.LEXCH5)).OR.NOT.NOTUHF (NOTUHF=(INF(64)+1).EQ.1, when it is true we are in the RHF case).

4.3.11 clakk and clalll

If the LEXCH4 logical variable is true, the *CLAKKK* subroutine is called, in order to calculate the exchange contribution $\sum_h < L1L3^{\mathbf{h}}|L2^{\mathbf{g}}L4^{\mathbf{h+g'}} >$.

At first the L2L4 couple is treated. From the *SHELLXN* and the *VIC2* routines we know that:
X6OLD=X(L4)-X(L1)+X(**h+g'**) (the same for Y and Z)
X21OLD=X(L2)-X(L1)+X(**g**) (the same for Y and Z)
Then:
XX=X(L4)-X(L2)+X(**h+g'-g**) (the same for Y and Z)

For a given L1 shell, ICAME, defined in the *SHALLXN* routine (instruction 266 of tab.4.14), indicates the starting point in the NQGSHG array for the translation vectors of the L1L3$^{\mathbf{h}}$ couple and NSH24 is the the irreducible vector of the first star of such couple.

A DO-loop running on the L1L3 stars

N=first vector of the first star of L1L3

s13=exponent of the product of the overlap 1324 : the two constant terms and the term sa-sb-g per L1L3, amnce il termine sc-sd-g per L2L4 (che viene calcolato dopo in s24).

NSH23: vettore irriducibile (o primo vettore) della star a cui sono arrivata

Loop su tutti i vettori dalla prima star a quella in cui sono arrivata.

N=vettore in NQGSHG

S24=sc-sd-g per L2L4

ITM:contatore, incrementa per ogni k che soddisfi la condizione

MGIPK(ITM)=N: stock vettori da considerare

azzero RO

4.3.12 Last parto of shellxn, non completa

- LLCOUL=ILTUTC.NE.0
- LPOPLJ=.NOT.(ISOUR.OR.NW)
- NCICLX.NE.0 verified only if the CLALLL or CLAKKK subroutines are called

Let us analize, now, the following if-then-else chain, which calls the COMB subroutines in order to add the bielectronic contributions of a given set of $\mu\nu$,$\sigma\omega$ atomic orbitals.

```
1) IF(LLCOUL.AND.LPOPLJ)THEN
    IF(NCICLX.EQ.0)GOTO 10095
2)  IF(MOTUHF)THEN
       IF(NCICLX-2)11091,11092,11093
       11091 CALL COMBJ1(RJ,RO)
       GOTO 10095
       11092 CALL COMBJ2(RJ,RK)
       GOTO 10095
       11093 CALL COMBJ3(RJ,RO,RK)
2)  ELSE
       IF(NCICLX-2)12091,12092,12093
       12091 CALL COMBX1(RJ2,RO)
       GOTO 12094
       12092 CALL COMBX2(RJ2,RK)
       GOTO 12094
       12093 CALL COMBX3(RJ2,RO,RK)
       12094 CALL MXMB(RJ2,ID34,1,PTEM,
        *1,ID34,FG_IRR(ICOSTA+1),1,NTUT,ID12,ID34,2)
2)  ENDIF
    10095 CALL MXMB(RJ,ID34,1,PTEM,1,1,FG_IRR(ICOSTA+1),1,1,ID12,ID34,1)
    CALL MXMB(RJ,ID34,1,PTEM,1,1,FG_IRR(ICOSTA+1),1,1,ID12,ID34,1)
1) ELSE
3)  IF(NCICLX.NE.0)THEN
       IF(NCICLX-2)10091,10092,10093
       10091 CALL COMBX1(RJ,RO)
       GOTO 10094
       10092 CALL COMBX2(RJ,RK)
       GOTO 10094
       10093 CALL COMBX3(RJ,RO,RK)
    10094 CALL MXMB(RJ,ID34,1,PTEM,
       CALL MXMB(RJ,ID34,1,PTEM,
        *1,ID34,FG_IRR(ICOSTA+1),1,NTUT,ID12,ID34,NSPSTA)
3)  ENDIF
1) ENDIF
```

If the Coulomb term is non-null and the shells are not d and f functions, the if-then-else number 1) enter in its first part. If no Exchange term is present (NCICLX.EQ.0 only) the instruction 10095 is directly executed whereas the bielectronic term contains only the Coulomb contribution. Se non ho uno dei

due contributi di scambio o se sono in un caso closed shell (MOTUHF), allora guardo quanto vale NCICLX: puo‘ valere 0,1,2,3. Se vale 0 abbiamo gia‘ visto che si passa direttamente all'istruzione 10095, in quanto non c'e' contributo di scambio. Se NCICLX=1, cioe' NCICLX-2¡0, liistruzione 11091 viene eseguita. Nella subroutine CLAKKK il valore di NCICLX viene infatti posto uguale a 1, e tale valore viene utilizzato che nessun integrale di scambio di tipo RK e' presente. La subroutine COMBJ1, dunque, somma in RJ il valore di RJ (integrale coulombiano) e quello di RO (scambio di tipo RO) scalato di -0.005*PAR(48), dove PAR(48) e' la percentuale di scambio esatto Hartree-Fock utilizzato (20 di default per il funzionale di tipo B3LYP).
Se NCICLX=2, NCICLX-2=0, e ricadiamo nel caso in cui solo integrali di scambio di tipo RK vengono calcolati. Nella subroutine CLAKKK il valore NCICLX viene posto uguale a NCICLX+2. Segue, dunque, che se sia integrali di tipo RK, sia integrali di tipo RO sono presenti quando NCICLX=3 (=1+2¿0, cioe' sia la subroutine CLALLL che la CLAKKK vengono chiamate). Le subroutine COMBJ2 e COMBJ3 vengono rispettivamente chiamate nei due casi, e sommano in RJ il valore di RJ iniziale piu' il valore degli integrali RK o RO+RK, scalati di-0.005*PAR(48).

If the condition number 2 is not satisfied (that is ..... open-shell system), the COMBX subroutines are called, the COMBX1 if NCICLX=1 (RO exchange term only), the COMBX2 if NCICLX=2 (RK exchange term only), the COMBX3 if NCICLX=3 (both RO and RK terms): they all stores the exchange term, scaled by -0.005*PAR(48), in RJ2, that is added, in the instruction 10094, to the Fock matrix.

At the end of the if-then-else number 2, the instruction 10095, adding the bielectronic contributions (apart from the RJ2 terms that, if present, have been already added in the second if-then-else) to the Fock matrix,is executed.

If there are not any Coulomb term or if the Pople package for integrals is not used, the third if-then-else cycle starts. The same options as the previous cycles based on the NCICLX value are present, and the COMBX subroutines are called. The result, RJ, is added to the Fock matrix.

The do-loops number 7, running on the irreducible **g** vectors of a star, and 6, running on the couples of the set ends after the if-then-else chain.

4.4 Deltap XXX

In section 4.3 the selection of bielectronic integrals to be calculated and their organization into blocks have been explored. It has to be reminded that about **the 80% of the calculation cost is due to the evaluation of bielectronic terms**. In particular, in geometry optimization and frequency calculation, when several SCF calculations must be performed at different geometries, the cost of calculation often increases dramatically. In order to decrease the heavy cost of geometry optimization and frequency calculation, then, one could try to act on the delicate bielectronic part of the code. In the case of frequency calculations, after the SCF calculation at the equilibrium geometry, a SCF is performed for each displacement of each irreducible atom. Then, only integrals with indices belonging to the displaced atom must be recalculated, whereas, when none of the μ,ν,σ,ω indices belongs to the displaced atom, the integral remains unchanged with respect to the equilibrium point.

As discussed in section 4.3, **for large systems, the bielectronic terms must be calculated at each SCF step**. The result of each single integral, then, cannot be stored: only the sum of such terms is stored in the Fock matrix. If (0) is the equilibrium point and (1) indicates the system in which one atom has been desplaced, we can write

$$F_{\mu\nu}^{\mathbf{g}} = F_{\mu\nu}^{(0)\mathbf{g}} - \left[\sum_{\sigma,\omega,\mathbf{g}'} I_{\mu\nu,\sigma\omega}^{\mathbf{g},\mathbf{g}'} P_{\sigma\omega}^{\mathbf{g}'}\right]^{(0)} + \left[\sum_{\sigma,\omega,\mathbf{g}'} I_{\mu\nu,\sigma\omega}^{\mathbf{g},\mathbf{g}'} P_{\sigma\omega}^{\mathbf{g}'}\right]^{(1)} = \tag{4.54}$$
$$= F_{\mu\nu}^{(0)\mathbf{g}} + \sum_{\sigma,\omega,\mathbf{g}'} \left[I_{\mu\nu,\sigma\omega}^{(1)\mathbf{g},\mathbf{g}'} P_{\sigma\omega}^{(1)\mathbf{g}'} - I_{\mu\nu,\sigma\omega}^{(0)\mathbf{g},\mathbf{g}'} P_{\sigma\omega}^{(0)\mathbf{g}'}\right]$$

where I indicates the sum of exchange and coulomb integrals with μ,ν,σ,ω,$\mathbf{g}$,$\mathbf{g}'$ indices (two coulomb and two exchange integrals)

$$I_{\mu\nu,\sigma\omega}^{\mathbf{g},\mathbf{g}'} = \sum_{\mathbf{h}} I_{\mu\nu,\sigma\omega}^{\mathbf{g},\mathbf{g}'}(\mathbf{h}) = \sum_{\mathbf{h}} \left[C_{\mu\nu,\sigma\omega}^{\mathbf{g},\mathbf{g}'}(\mathbf{h}) + C_{\mu\nu,\omega\sigma}^{\mathbf{g},\mathbf{g}'}(\mathbf{h}) + X_{\mu\sigma,\nu\omega}^{\mathbf{g},\mathbf{g}'}(\mathbf{h}) + X_{\mu\omega,\nu\sigma}^{\mathbf{g},\mathbf{g}'}(\mathbf{h})\right] \tag{4.55}$$

$$C_{\mu\nu,\sigma\omega}^{\mathbf{g},\mathbf{g}'}(\mathbf{h}) = P_{\sigma\omega}^{\mathbf{g}'} < \mu^0\nu^{\mathbf{g}} \,|\widehat{C}|\sigma^{\mathbf{h}}\omega^{\mathbf{h}+\mathbf{g}'} > \tag{4.56}$$

$$X_{\mu\sigma,\nu\omega}^{\mathbf{g},\mathbf{g}'}(\mathbf{h}) = -\frac{1}{2} P_{\sigma\omega}^{\mathbf{g}'} < \mu^0\sigma^{\mathbf{h}} \,|\widehat{X}|\nu^{\mathbf{g}}\omega^{\mathbf{h}+\mathbf{g}'} > \tag{4.57}$$

In section 4.3 the reason why **given a μ,ν,σ,ω,$\mathbf{g}$,$\mathbf{g}'$ set, the corresponding exchange and coulomb integrals can be added and multiplied by the same bloch of the density matrix** has been discussed. It is clear,

from equation 4.54, that the set $I^{\mathbf{g},\mathbf{g}'}_{\mu\nu,\sigma\omega}$ must be calculated for the equilibrium geometry and the symmetry reduced system only.

The density matrix is more sensitive than bielectronic integrals to geometry displacements, as its general $P^{\mathbf{g}}_{\mu\nu}$ element is defined as a sum over all the occupied states and the reciprocal lattice $\mathbf{k}$ vectors of the c_μ and c_ν coefficients of the wave function $\psi = \sum_i c_i\chi_i$, where $i = 1...\mu...\nu...N_{AO}$ and N_{AO} is the total number of atomic functions. In the present case we have to consider matrix elements such as

$$P^{\mathbf{g}'}_{\sigma\omega} = \sum_{\mathbf{k}} \sum_{n}^{Nocc} exp(i\mathbf{k}\mathbf{g}')c^*_{\sigma n}(\mathbf{k})c_{\omega n}(\mathbf{k}) \tag{4.58}$$

For this reason, a certain number of $P^{\mathbf{g}'}_{\sigma\omega}$ terms could be affected by the geometry difference even if σ and ω do not belong to the displaced atoms (see figure 4.4). We must be able, then, to identify wich are the density matrix blocks that must be recalculated for the symmetry reduced system, and, starting from equation 4.54, we get:

$$F^{\mathbf{g}}_{\mu\nu} = F^{(0)\mathbf{g}}_{\mu\nu} + \sum_{\sigma,\omega,\mathbf{g}'} \left[\left(I^{(1)\mathbf{g},\mathbf{g}'}_{\mu\nu,\sigma\omega} P^{(1)\mathbf{g}'}_{\sigma\omega} - I^{(0)\mathbf{g},\mathbf{g}'}_{\mu\nu,\sigma\omega} P^{(0)\mathbf{g}'}_{\sigma\omega} \right) + I^{(0=1)\mathbf{g},\mathbf{g}'}_{\mu\nu,\sigma\omega} \Delta P^{(1-0)\mathbf{g}'}_{\sigma\omega} \right] \tag{4.59}$$

$$\Delta P^{(1-0)\mathbf{g}'}_{\sigma\omega} = P^{(1)\mathbf{g}'}_{\sigma\omega} - P^{(0)\mathbf{g}'}_{\sigma\omega}$$

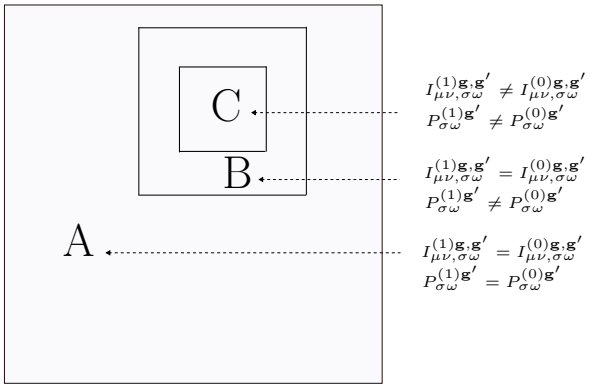

Figure 4.4. Schematic representation of the density matrix. A: region in which both the density matrix elements and the bielectronic integrals remains unchanged when passing from the equilibrium geometry (0) to the symmetry reduced system (2); ,B: region where only density matrix elements are affected by the geometry displacement, C: the $\sigma\omega$ block.

The density matrix is calculated at each SCF calculation, so that the $\Delta P^{(1-0)\mathbf{g}'}_{\sigma\omega}$ difference can be calculated (first part of the *DELTAP* subroutine). When this difference is smaller than a given threshold, the density

matrix calculated for the equilibrium point (0) and the one calculated for the symmetry reduced system (1) can be considered equal. On the contraty, when the difference is larger than a given threshold, the new density matrix and the corresponding $I_{\mu\nu,\sigma\omega}^{(0=1)\mathbf{g},\mathbf{g}'}$ integrals must be calculated.

The selection of density matrix blocks for which the bielectronic integrals in the

to be calculated in the symmetry

4.4.1 The *DELTAP* subroutine

This subroutine is called in the SCF routine in order to estimate the variation of the density matrix between two subsequent cycles. In particular, the part of the SCF routine in which $DELTAP$ is called is shown in the following (note that allocations/deallocations of variables are not shown here).

```
   ...
   DIRECTSCF=inf(114).EQ.1
   ...
   DEP=inf(100).NE.0
   DEPACT=.false.
   ...
   RESTDP=inf(57).NE.0.AND.INF(57).NE.4.AND.inf(16).NE.0
1  IF(DEP)THEN
   2 IF(INF(16).GT.1.OR.RESTDP) THEN
      3 IF (INF(35).NE.0) THEN
        DEPACT=.FALSE.
      3 ELSE
        4 IF ((.NOT.DIRECTSCF).OR.INF(101).EQ.0) THEN
          DEPACT=.FALSE.
          CALL DELTAP(INF(16),TOLDP,DP,DPMAX)
        4 ELSE
          DEPACT=.TRUE.
          CALL DELTAP(INF(16),TOLDP,DP,DPMAX)
        4 ENDIF
      3 ENDIF
    2 ENDIF
1  ENDIF
   ...
```

Let us describe, first, the logical variables appearing in the four conditions.

- inf(100) is set equal to 1 by default, so that DEP is true: the difference between the P matrices of two subsequent cycles is calculated.
- inf(16) indicates the number of SCF cycles that have been performed.
- RESTDP is true when GUESSP or GUESSF are set (see manual)

- inf(35) is 3 when the convergence on energy is satisfied, 2 when the convergence on eigenvalues is satisfied, 1 when the number of SCF cycles exceeds, 0 in all the other cases.

- inf(101) is equal to 1 if the DELTAP option is set in input, 0 when it is not activated.

- DIRECTSCF is true when inf(114)=1, corresponding to the SCFDIR input keyword (integrals are computed at each cycle and not stored)

The first condition is then satisfied by default; the second one means "if we are neither in the 0, nor in the 1st cycle, or, even if we are in the 1st cycle, the F or P initial guess is activated, then ..". The third condition is verified both if we are not converged on energy or eigenvalues, and if we do not exceed the maximum number of SCF cycles: in these cases the DEPACT logical variable is set to .false.; else, if we are not in direct conditions or the DELTAP option is not activated, DEPACT is set to .false. and the *DELTAP* routine is called. If the DELTAP option is activated and we are in direct conditions, DEPACT is set to true and the *DELTAP* routine is called.

This part of the *SCF* subroutine can be summarized as follows: the call to the *DELTAP* routine is controlled by the inf(100) parameter, and it is always called by default. The DEPACT logical variable is set to .true. only when the DELTAP option is active and the calculation is a direct calculation. Note that *DELTAP* corresponds to the name of the subroutine, and DELTAP is an input keyword by which the selection of P blocks that can be re-utilized is activated.

We come back, now, to the *DELTAP* subroutine; the difference between the previous (PG_IRR) and the actual (TMPG_IRR) irreducible elements of the density matrix is calculated when inf(35)=0. Let us have a look at the corresponding instructions in the code:

```
DP=0._FLOAT
DO  I=1,NTUT
DPG_IRR(I)=PG_IRR(I)-TMPG_IRR(I)
DP=DP+DPG_IRR(I)**2
ENDDO
DPMAX=MAXVAL(ABS(DPG_IRR(1:NTUT)))
DP=SQRT(DP/NTUT)
```

DP is first set to zero. A DO-loop on the total number of irreducible elements, NTUT, begins. NTUT has been set equal to [(inf(64)+1)*inf(19)],

so that the number of elements is doubled in the UHF case (remember that inf(64)=0 corresponds to RHF, inf(64)=1 to UHF; inf(19) is set in the $GMFCAL$ subroutine and corresponds to the total number of irreducible elements of the Fock and density matrices). The difference DPG_IRR(1,NTUT) is then calculated for each element of the irreducible density matrix. The mean square difference, DP, and the maximum difference, DPMAX, are then calculated.

The second part of the *DELTAP* subroutine is executed only when DELTAP and SCFDIR (and, at the moment, NOBIPOLA) are set in input. Table 4.29 shows and comments the second part of this routine.

	MEMORY_G	module for DELTAP
	inf(101)	case('DELTAP'), inf(101)=1, DEPACT=.true. (SCF)
	inf(19)	size of the vector PG_IRR ($GMFCAL$)
	NTUT	NTUT=inf(19)*(inf(64)+1), single or double size of PG_IRR for RHF or UHF calculation respectively
	LDEP	If true, the calculation of the block of the PG_IRR matrix is disregarded (SIZE:total number of the blocks)
	ICOC	Starting block for a given couple set (SIZE:MVLAF)
	itol(19)	tolerance on the ΔP integral screening, $10^{-itol(19)}$;itol(19)=20 by default
	TOLDP	TOLDP=$0.1^{itol(19)}$
1	IF(INF(114).EQ.1.AND.INF(101).NE.0)THEN	This condition is satisfied when DELTAP is set in input
	QVRS(1:LAFPOL)= QVRS(1:LAFPOL) - -QVRSO(1:LAFPOL)	
	QTOT(1:NAFPOL)= QTOT(1:NAFPOL) - -QTOTO(1:NAFPOL)	
	NUMBO=0	
	ICOCO=0	
	ILDEP=0	
	JLDEP=0	
1A	DO MVLU=1,MVLAF	Loop on the couple sets
	ICOC(MVLU)=ICOCO	Starting block in PG_IRR for the MVLU couple set
	NSHH= =MAX(IDMCOU(MVLU),IDMEXL(MVLU))	Maximum number of $\mathbf{g}^{irr}$ stars between the C and X terms
	IF(NSHH.EQ.0)CYCLE	No vectors, no element for the couple set
	MOC=NCF(MVLU)	First couple of the set
	IROSE=IROF(MVLU)	First $\mathbf{g}^{irr}$ of the couple set
	ID34=LATAO(LA3(MOC+1))* *LATAO(LA4(MOC+1))	Size of the block
1B	DO KSH=1,NSHH	Loop on $\mathbf{g}^{irr}$ stars
	NTT=NSHGI(IROSE+KSH)*ID34	Total number of elements for the KSH star of a couple of the set
	ZZZ=0._FLOAT	
1C	DO MOC=1,NTT	Loop on the elements
	NUMBO=NUMBO+1	Counter on TOTAL elements of the PG_IRR matrix
	ZZZ=ZZZ+ABS(DPG_IRR(NUMBO))	Sum of the absolute value of DPG_IRR for all the block
1C	ENDDO	
	ZZZ=ZZZ/NTT	Mean absolute difference on the elements of a block
	ICOCO=ICOCO+1	Counter of the blocks
	ADEP(ICOCO)=ZZZ	Mean absolute difference on the elements of the ICOCO block
2	IF(ZZZ.LT.TOLDP)THEN	Condition: if the difference is smaller than the threshold
	LDEP(ICOCO)=.TRUE.	LDEP for that block is true
	ILDEP=ILDEP+1	Counter on the blocks that do not need to be recalculated
	JLDEP=JLDEP+NTT	Counter on elements that do not need to be recalculated
	else	
	LDEP(ICOCO)=.FALSE.	The condition is not satisfied, the block has to be recalculated
2	ENDIF	
1B	ENDDO	
1A	ENDDO	
	...	Printing options
1	ENDIF	

Table 4.29. The part of the *DELTAP* subroutine activated by the DELTAP keyword (+ SCFDIR and NOBIPOLA).

(1) Loop on the couple sets for L1,L2 (MVLA)	
(2) Loop on the g stars of L3,L4 (MGISH)	
(3) Loop on the girr of a star (NGI)	
(4) Loop on the couple sets for L3,L4 (MVLU)	
NYC=NCF(MVLU)+1	First couple of the MVLU couple set.
NCFU=NCF(MVLU+1)	Last couple of the MVLU couple set.
IPPEP=ICCS2(NYC)	Starting point of the g stars of the first couple of the MVLU 06
(5) Loop on the g stars of L3,L4 (KSH)	
IPPEP=IPPEP+1	IPPEP is incremented by one in the KSH loop; it is used as a counter of the stars for the first couple of the MVLU couple set.
NLX=NGSHG(IPPEP+1)-NGSHG(IPPEP)	Total number of g in the KSH star for the NYC couple.
IBNUM=KSH.LE.ISTA34	the number of the star (KSH) is less or equal to the maximum number of the stars of the MVLU set for coulomb terms
IF(DEPACT) THEN	If the DELTAP case is active
ICOCO=ICOC(MVLU)+KSH	P Block of the MVLU set in the KSH star
IF(LDEP(ICOCO)) THEN	If true, the calculation of the ICOCO block can be disregarded
DO NCU=NYC,NCFU	Loop on the couples of the set
L3=LA3(NCU)	First member of the couple, L3
L4=LA4(NCU)	Second member of the couple, L4
IF(IBNUM.AND.(LPESO.OR.L3.GE.L4))THEN	If IBNUM and if two couples are generated by hermiticity or the couple is in the low half-part of the matrix
NNL=NGSHG(ICCS2(NCU)+KSH)	First g of the KSH star of the NCU couple
DO NL=1,NLX	Loop on the number of g vectors
MGP=NQGSHG(NNL+NL)	g vector
IF(L3EQL4.AND.MGP.GT.NN1(MGP))CYCLE	If the couple is diagonal and the g vector label is greater than the label of its reverse
ENDDO	
ENDIF	
ENDDO	
GOTO 888	The 6^{th} and the 7^{th} loops are overstep if LDEP(ICOCO)=.true.
ENDIF	
ENDIF	
(6) Loop on the couples of the L3,L4 couple set (NCU)	
(7) Loop on the g vectors (NL)	

Table 4.30.

(1) Loop on the couple sets for L1,L2 (MVLA)	
(2) Loop on the g stars of L3,L4 (MGISH)	
(3) Loop on the girr of a star (NGI)	
(4) Loop on the couple sets for L3,L4 (MVLU)	
(5) Loop on the couples of the L3,L4 couple set	
NSHTOT=MAX(ISTA34,LPF22)	
IF(DEPACT) THEN	If the DELTAP case is active
DO KSH=1, NSHTOT	
ICOCO=ICOC(MVLU)+KSH	P Block of the MVLU set in the KSH star
IF(LDEP(ICOCO)) THEN	If true, the calculation of the ICOCO block can be disregarded
L3=LA3(NCU)	First member of the couple, L3
L4=LA4(NCU)	Second member of the couple, L4
IBNUM=KSH.LE.ISTA34	
IF(IBNUM.AND.(LPESO.OR.L3.GE.L4))THEN	If IBNUM and if two couples are generated by hermiticity or the couple is in the low half-part of the matrix
NNL=NGSHG(ICCS2(NCU)+KSH)	First g of the KSH star of the NCU couple
NLX=NGSHG(ICCS2(NCU+1))-NNL	
DO NL=1,NLX	Loop on the number of g vectors
MGP=NQGSHG(NNL+NL)	g vector
IF(L3EQL4.AND.MGP.GT.NN1(MGP))CYCLE	If the couple is diagonal and the g vector label is greater than the label of its reverse
ENDDO	
ENDIF	
ENDDO	
GOTO 888	The 6^{th} and the 7^{th} loops are overstep if LDEP(ICOCO)=.true.
ENDIF	
ENDIF	
(6) Loop on the couples of the L3,L4 couple set (NCU)	
(7) Loop on the g vectors (NL)	

Table 4.31.

JPETAL=JPETAL+1	Counter on the g vectors
MGP=NQGSHG(NL)	g vector
NU=INO(JPETAL)*ID34+NUMBO	Starting point of the couple L3,L4,g
MV=INOIV(JPETAL)	Symmetry operator generating the g vector by the corresponding girr
IF(MOTUHF)THEN	Vedere meglio la condizione
IF(MV.EQ.1)THEN	Se il g che sto considerando e' irriducibile
PTEM(1:ID34)=PG_IRR(NU+1:NU+ID34)	Trasferisco il blocco di PG_IRR cosi' com'e' in PTEM
ELSE	
PTEM(1:ID34)=0._FLOAT	
DO J=MOVF34(MV-1)+1,MOVF34(MV)	Elementi dell'operatore per cui devo ruotar il blocco
I=NO34(J)	
PTEM(I)=ZO34(J)*PG_IRR(NOM34(J)+NU)+PTEM(I)	Ottengo il blocco L3,L4,g
ENDDO	
ENDIF	
ELSE	
IF(MV.EQ.1)THEN	Come sopra ma per UHF, ho due P
J=NU+NTUT	
DO I=1,ID34	
PTEM(I)=PG_IRR(NU+I)	
PTEM(ID34+I)=PG_IRR(J+I)	
ENDDO	
ELSE	
PTEM(1:ID34D)=0._FLOAT	
DO J=MOVF34(MV-1)+1,MOVF34(MV)	
I=NO34(J)	
N=NOM34(J)+NU	
PTEM(I)=ZO34(J)*PG_IRR(N)+PTEM(I)	
PTEM(ID34+I)=ZO34(J)*PG_IRR(NTUT+N)+PTEM(ID34+I)	
ENDDO	
ENDIF	
ENDIF	

Table 4.32.

4.5 The FROTA subroutine

The *FROTA* subroutine is called by the *FDIK* subroutine at each SCF step in order to transform the irreducible Fock matrix, $F^{\mathbf{g}}_{irr}$, into the reducible Fock matrix, $F^{\mathbf{g}'}_{red}$. This implies the rotation of a $F^{\mathbf{g}}(\lambda_1,\lambda_2)$ block into a $F^{\mathbf{g}'}(\lambda'_1,\lambda'_2)$ block, as shown by equations 4.8 and 4.11. In the case of a pp block, for example, the nine elements of $F^{\mathbf{g}}_{irr}$ are linearly combined to give the nine elements of the resulting $F^{\mathbf{g}'}_{red}$ block.

$$F^{\mathbf{g}'}_{red}(\lambda'_1,\lambda'_2) = \widehat{R} F^{\mathbf{g}}_{irr}(\lambda_1,\lambda_2) \tag{4.60}$$

The $F^{\mathbf{g}}_{irr}$ and $F^{\mathbf{g}'}_{red}$ sizes are calculated in the *GMFCAL* subroutine and stored, respectively, as inf(19) and inf(11). We must remember that only the upper-half reducible Fock matrix is stored.

The case of the ammonia molecule (one s and one sp shells on nitrogen atom - 1,2 - and one s shell on each hydrogen atom - 3,4,5) will be used as an example in the following discussion; the calculation can be followed step by step as the molecular case is much simpler than a periodic system, and the number of terms is relatively small (note that, obviously, only one **g** vector for each couple exist, $\mathbf{g}_1$=0,0,0). We are going to extend our considerations later on to periodic systems.

The NH_3 Fock matrix scheme is shown in figure 4.5. Actually, both the matrices are stored in vectors (FG_IRR and FG_RED):

```
FG_IRR:
   0      1       2         3         4         5        6       7        8        9      10     11     12         13
-14.6   -3.9     2.5E-19  -6.5E-18  -5.6E-02  -0.70     0.0   -3.2E-17  8.2E-02  0.0    0.8    0.0    5.0E-19  -4.3E-17

   14     15      16        17        18        19       20     21       22       23     24     25     26         27
  0.0    0.78   -4.4E-17   8.2E-02   5.0E-19  -3.5E-17  0.41  -0.44     -0.55     0.0   -0.45  -0.19  9.5E-02   -7.8E-02

FG_RED:
   0         1         2          3          4          5         6        7(10)     8(7)    9(11)     10(15)     11(8)
 -14.6      -3.9       2.5E-19   -6.6E-18   -5.6E-2    -4.4E-1   -4.4E-1  -4.4E-1   -7.0E-1   0.0       -4.3E-17   8.2E-2

  12(12)     13(16)    14(20)      15(21)     16(21)     17(21)   18(22)     19(23)    20(24)   21(25)    22(22)     23(23)
 -5.5E-1    -5.5E-1   -5.5E-1      7.8E-1     0.0        5.0E-19  0.0        3.9E-1    3.9E-1   7.8E-1   -3.5E-17   -4.5E-1

   24(24)    25(25)    26(22)      27(23)     28(24)     29(25)   30(26)    31(26)    32(26)   33(27)    34(27)     35(27)
  2.3E-1    2.3E-1    4.1E-1     -1.9E-1    -1.9E-1    -1.9E-1   9.5E-2   -7.8E-2   -7.8E-2   9.5E-2   -7.8E-2     9.5E-2
```

(elements are enumerated in the sequential order of storage; parenthesis in the reducible fock matrix FG_RED contain the position of the FG_IRR element from which the FG_RED element is generated by rotation- FG_RED=$\widehat{R}$·FG_IRR)

In order to understand the pointers of the Fock matrix, we must come back to the first five CRYSTAL subroutine (ref.unire documenti): in table 4.34 the meaning of pointers used in the present subroutine is discussed.

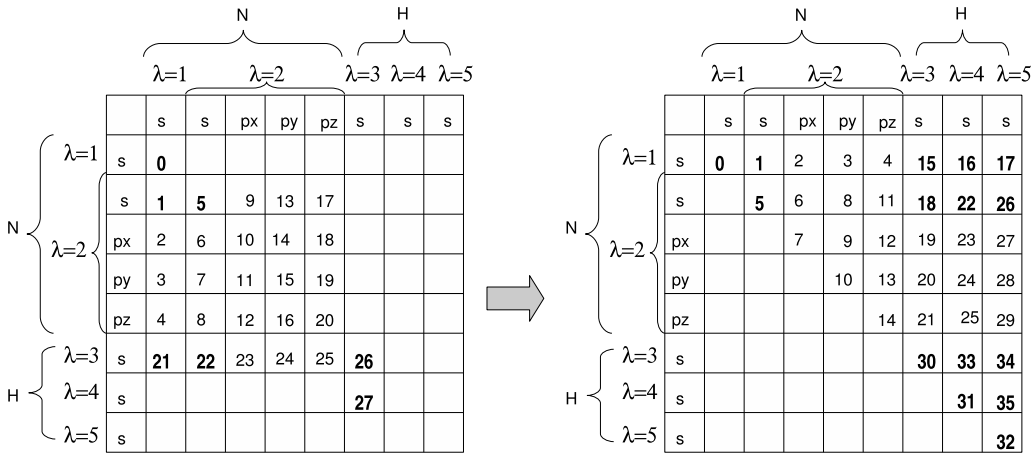

Figure 4.5. Labels of the $F^{\mathbf{g}}_{irr}$ (left) and $F^{\mathbf{g}}_{red}$ (right) matrix elements for ammonia: note that inf(19)=28 and inf(11)=36; in $F^{\mathbf{g}}_{irr}$ the whole diagonal blocks are stored, whereas $F^{\mathbf{g}}_{red}$ contains only half the diagonal blocks. Bold-sized numbers are the elements of the NSTATG ($F^{\mathbf{g}}_{irr}$) and NNNC2 ($F^{\mathbf{g}}_{red}$) vectors indicating the starting points of the corresponding λ_1,λ_2 block in the MVLU list. For example NSTATG(MVLU=5)=22, and from the MVLU list we know that the first couple of MVLU=5 is $\lambda_1 = 3$ and $\lambda_2 = 2$. Compare to table 4.33.

MVLU	1	2	3	4	5	6	7
NSTATG	0	1	5	21	22	26	27
NCF	1-0	2-1	3-2	6-3	9-6	12-9	18-12
λ_1,λ_2	1,1	2,1	2,2	3,1; 4,1; 5,1	3,2; 4,2; 5,2	3,3; 4,4; 5,5	4,3; 3,4; 5,3; 3,5; 5,4; 4,5
NNNC2	0	1	5	15; 16; 17	18; 22; 26	30; 31; 32	—; 33; —; 34; —; 35
INOIV	1	1	1	1; 2; 3	1; 2; 3	1; 2; 3	—; 6; —; 3; —; 4

Table 4.33. Couple sets (MVLU), couples of the set (λ_1,λ_2), pointers on the CS (NCF=NCF(MVLU+1)-NCF(MVLU)), $F^{\mathbf{g}}_{irr}$ (NSTATG), $F^{\mathbf{g}}_{red}$ (NNNC2) and on the symmetry operator generating the couple (INOIV). The couples in each column are generated by the first couple in the set, related to the others by the corresponding symmetry operator in the INOIV vector ; for example the couple 3,1 generates the couples 4,1 and 5,1. Compare to figure 4.5.

In the transformation from $F^{\mathbf{g}}_{irr}$ to $F^{\mathbf{g}}_{red}$ we must know, preliminarily, the starting point of the blocks involved in equation 4.60: they are given by the vectors NSTATG and NNNC2 (compare figure 4.5 to table 4.33). We know, from the *GORDSH* subroutine, that shell couples can be organised in "couple sets" (CS): it is clear that some couples are "unique", others are symmetry related, as shown by the NCF vector (see table 4.33). For each couple to be

inserted in $F^{\mathbf{g}}_{red}$, then, we must know the operator to be used in equation 4.60 (INOIV vector, table 4.33) .

The case of ammonia, however, is very simple, as only one **g** vector is present. Our discussion, then, must be accomplished by extending the previous considerations to a periodic system, such as graphite, a closed-shell system with a s (shells 1 and 3) and a sp (shells 2 and 4) on each carbon atom. In such cases also the starting point for the **g** cell in the $F^{\mathbf{g}}_{red}$ is required: all of the **g** cells are generated and the starting point of each one is obtained by the number of elements of the block. The $\mathbf{g}_{irr}$ generating the **g** vector we are interested in is identified by the INO vector.

In the next subsection the structure of the subroutine is analysed, in order to show how the equations are implemented and the pointers utilised; the example of graphite and ammonia will be discussed in detail.

4.5.1 Structure of the code

The *FROTA* subroutine is divided in two parts: the first refers to inf(64)=1, that is to open-shell systems, and the second to inf(64)=0, that is to closed-shell systems. In the UHF case a "double" $F^{\mathbf{g}}_{red}$ matrix is built ($F^{\mathbf{g}}_{\alpha}$ and $F^{\mathbf{g}}_{\beta}$, where the only difference is the exchange contribution), so that a double $F^{\mathbf{g}}_{irr}$ is present (two instructions instead of one in the most internal do-loop): the sizes of the reducible and irreducible Fock matrices are then, respectively, 2×inf(11) and 2×inf(19).

In a do-loop chain running on the CS (loop **(a)** in table 4.34), the couples c of the CS (loop **(b)**) and the **g** vectors of each c (loop **(c)**), the starting point of the block to be generated is identified. If $\lambda_1 = \lambda_2$, only the upper half part of the corresponding block in the reducible Fock matrix is stored (IKY(i) gives the number of elements of half a matrix with i size), whereas the whole block is stored when $\lambda_1 \neq \lambda_2$ (see instruction from 16 to 23 in table 4.34).

For each block, the starting point of the NCU couple in $F^{\mathbf{g}}_{red}$ is given by NNNC2(NCU) (see line 29 in table 4.34). In $F^{\mathbf{g}}_{irr}$, the starting point for each CS is given by NSTATG(MVLU) (line 07); note that it is a CS label, not a couple label, as in a set the first couple generates all the other couples in the set (for example, in table 4.35, the "mother" couple 1,1 generates the couple 3,3, and only NSTATG(1) is required).

(a) DO MVLU=1,MVLAF	Total number of couple sets CS ($GORDSH$)
02 NSHH=IDIMF(MVLU)	Total number of **g** stars for each CS ($GMFCAL$)
03 IF(NSHH.EQ.0)CYCLE	Maximum number of **g** vectors allocated on each couple type
04 NYC=NCF(MVLU)+1	For each CS NCF gives the starting point in the LA3,LA4 vectors ($GORDSH$)
05 L3=LA3(NYC)	Shell label λ_1 ($GORDSH$, NYC=1:NCF(MVLAF+1))
06 L4=LA4(NYC)	Shell label λ_2
07 NUMBO=NSTATG(MVLU)	Starting point of a couple set in F_{irr} ($GMFCAL$)
08 IPETAL=INZVLB(MVLU)	Starting point of the **g** vectors for each CS in the NQGSHG vector, listing all the **g** vectors of all the CS type ($GMFCAL$)
09 NLX=ICCS2(NYC)	Starting point of the stars of **g** of a couple in NGSHG (GV)
10 NLX=NGSHG(NLX+NSHH+1)-NGSHG(NLX+1)	Total number of **g** vectors for each CS (GV)
11 INX4=LAT(L4)	List of the shell labels of each shell ($INPBAS$)
12 ID3=LATAO(L3)	List of the number of AO for each shell ($INPBAS$)
13 ID4=LATAO(L4)	
14 ID34=ID3*ID4	
15 ICAS1=JNCDU(MVLU).EQ.1	Number of CS that can be generated by hermiticity ($GMFCAL$)
16 IF(L3.EQ.L4)THEN	
17 ID34N=IKY(ID4)	Number of elements of half a matrix with ID4 size (SHELL_INFO module)
18 MOVF34(0:MVF)=LOVF34(0:MVF,INX4)	
19 ELSE	
20 INX3=LAT(L3)	
21 ID34N=ID34	
22 MOVF34(0:MVF)=NOVF34(0:MVF,INX3,INX4)	
23 ENDIF	
(b) DO 800 NCU=NYC,NCF(MVLU+1)	
25 L3=LA3(NCU)	
26 L4=LA4(NCU)	
27 ICASLT=L4.LT.L3	
28 IF(ICASLT.AND.ICAS1)GOTO 800	
29 MG=NNNC2(ICCS1(NCU))	Starting position of each couple in F_{red} (NNNC2, $GMFCAL$) and index of a couple in ILA12T (natural order of the couples, ICCS1 vector in $GORDSH$)
(c) DO NL=1,NLX	
31 J=IPETAL+NL	
32 NU=INO(J)*ID34+NUMBO	Sequence of the irreducible **g** vectors generating the reducible ones ($GILDA1$)
33 MV=INOIV(J)	Sequence of the operators generating the reducible **g** vectors by the corresponding irreducible **g** vectors in INO ($GILDA1$)
34 IF(ICASLT)THEN	
(d1) DO J=MOVF34(MV-1)+1,MOVF34(MV)	
36 I=KO34(J)+MG	
37 FG_RED(I)=FG_IRR(NOM34(J)+NU)* *ZO34(J)+FG_RED(I)	
(d1) ENDDO	
39 ELSE	
(d2) DO J=MOVF34(MV-1)+1,MOVF34(MV)	
41 I=NO34(J)+MG	
42 FG_RED(I)=FG_IRR(NOM34(J)+NU)* 43 *ZO34(J)+FG_RED(I)	
(d2) ENDDO	
45 ENDIF	
46 MG=MG+ID34N	
(c) ENDDO	
(b) 800 IPETAL=IPETAL+NHSQ	
(a) ENDDO	

Table 4.34. The $FROTA$ structure for the RHF case. In the right column: pointers used in the $FROTA$ subroutine; into parenthesis: the subroutine where they are defined. The pointers defined in the $GSYM11$ and $GSYM22$ subroutines are not described here, as they have already been discussed in the previous section.

As loop **(c)** spans sequentially all the **g** vectors, all of the **g** cells are generated, and the starting point for the **g** cell in the $F^{\mathbf{g}}_{red}$ is obtained by the instruction number 46 in table 4.34 (the **g** vectors are sequentially identified by the J label). INO(J) identifies the $\mathbf{g}_{irr}$ generating the **g** vector of interest and the starting point of the irreducible "**g**" block is given by instruction number 32 in table 4.34 (note that the diagonal blocks in $F^{\mathbf{g}}_{irr}$ are complete).

instruction	variable	values			
(a)	MVLU	1		2	
02	NSSH	1		2	
04	NYC	1		3	
05,06	L3,L4	1,1		2,1	
07	NUMBO	0		1	
08	IPETAL	0		0	
09	ICCS2(NYC)	0		0	
10	NLX	1		7	
20,11	INX3,INX4	0,0 (s,s)		1,0 (sp,s)	
12,13	ID3,ID4	1,1		4,1	
		couples of the CS		couples of the CS	
(b)	NCU	1	2	3	4
25,26	L3,L4	1,1	3,3	2,1	4,3
29	ICCS1(NCU)	1	11	5	15
29	MG	0	1	2	30
(c),31	J	1	43	1 2 3 4 5 6 7	43 44 45 46 47 48 49
32	INO(J)	0	0	0 1 1 1 1 1 1	0 1 1 1 1 1 1
32	NU	0	0	1 5 5 5 5 5 5	1 5 5 5 5 5 5
33	INOIV(J)	1	2	1 1 9 8 3 7 4	2 10 2 6 11 5 12

Table 4.35. Example of graphite pointers used in the *FROTA* subroutine (only the first two couple sets are shown and TOLINTEG 5 5 5 5 12). The meaning of each quantity is explained in table 4.34, where one can found also the instruction corresponding to the left column of this table. Note that the first $\mathbf{g}_{red}$ is generated by the first $\mathbf{g}_{irr}$, whereas the other 6 by the second $\mathbf{g}_{irr}$; note also that all the twelve symmetry operators are present in INOIV(J) (see figure grafite in prima parte, collegare)

Let us discuss, now, the IF conditions inside the **(b)** and **(c)** loops (lines 28, 34 and 39). Line 38 says that if in the couple λ_1,λ_2 we have $\lambda_1 > \lambda_2$ (low half part of the Fock matrix) AND if the λ_1,λ_2 block does not generate any other block by hermiticity, such a block is not generated in $F^{\mathbf{g}}_{red}$. If $\lambda_1 > \lambda_2$ but the λ_1,λ_2 block generates another block by hermiticity (JNCDU(MVLU)=2, line 15), the **(d1)** loop instructions are executed: the KO34 indices (see the *GSYM22* subroutine) are used to generate the "transposed" block, corresponding to the couple λ_2,λ_1. On the contrary, if $\lambda_1 \leq \lambda_2$ (upper-half matrix) the **(d2)** loop instructions are executed, and the block (or half the block in diagonal cases) is generated, using now the NO34 vector (*GSYM22* subroutine). The example of graphite is useful and easy to clarify these three conditions. The couples with non negligible overlap (with TOLINTEG 5 5 5 5 12, couples 1,3 and 3,1 are disregarded) are listed in the following, with 1 and 2 being the s and sp shells on the first carbon atom, and 3 and 4 the s and sp shells on the second carbon atom. Couples into parenthesis are generated by hermiticity. Consider NCU=7: the corresponding couple is 3,2, belonging to the CS MVLU=4. As JNCDU(4)=2, the couple set generates another couple by hermiticity (3,2→2,3;

1,4→4,1), so that we are in the condition $\lambda_1 > \lambda_2$ with JNCDU(4)≠1. The **(d1)** is executed, but the block 2,3 instead of the 3,2 one is generated, so that it falls it the upper-half part of the matrix.

MVLU	1	2	3		4		5
	1 1	**2 1**	**2 2**	(2 3)	**3 2**	(1 2)	**4 2**
	3 3	**4 3**	**4 4**	(4 1)	**1 4**	(3 4)	**2 4**

We can come back, now, to the ammonia example and discuss tables 4.36 and 4.37: they show the whole scheme of how F_{red} is generated by F_{irr} for couples 5,6 and 7. The three horizontal lines in each table divide the quantities recalled in the **(a)** do-loop, the ones in the **(b)** and **(c)** do-loops and the ones calculated in the **(d1)** (or **(d2)**) do-loop, respectively. In table 4.36 not all the elements are shown, but one can note the effect of non diagonal symmetry operators on the Fock matrix elements (see couples 8 and 9). For couples 13, 15 and 17 in table 4.37, the **(c)** and **(d)** loops are not executed, as $\lambda_1 > \lambda_2$ and JNCDU(13,15,17)=1; that means we are in the low-half part of the matrix and no block in the upper part can be generated by hermiticity.

variable	values												
MVLU	5												
NSSH	1												
NYC	7												
L3,L4	3,2												
NUMBO	22												
IPETAL	1												
ICCS2(NYC)	4												
NLX	1												
INX3,INX4	0,1												
ID3,ID4	1,4												
NCU	7			8					9				
L3,L4	3,2			4,2					5,2				
ICCS1(NCU)	12			17					22				
MG	18			22					26				
J	2			3					4				
INO(J)	0			0					0				
NU	22			22					22				
MV	1			2					3				
I	19	...	22	23	...	24	25	26	27	...	28	...	30
J	7	...	10	11	...	14	15	16	17	...	20	...	22
NOM34(J)	1	...	4	1	...	3	3	4	1	...	3	...	4
FG_IRR	-0.54	...	-0.19	-0.55	...	-0.45	-0.45	-0.19	-0.55	...	-0.45	...	-0.19
ZO34(J)	1.0	...	1.0	1.0	...	-0.866	-0.5	1.0	1.0	...	0.866	...	1.0
FG_RED(I)	-0.54	...	-0.19	-0.55	...	0.39	0.23	-0.19	-0.55	...	-0.39	...	-0.19

Table 4.36. The ammonia CS number 5 contains three couples: the first one has four non-null elements, the others have 6 non-null elements (non-diagonal operators).

variable	values								
MVLU	6			7					
NSSH	1			7					
NYC	10			13					
L3,L4	3,3			4,3					
NUMBO	26			27					
IPETAL	4			7					
ICCS2(NYC)	0			12					
NLX	1			1					
INX3,INX4	0,0			0,0					
ID3,ID4	1,1			1,1					
NCU	10	11	12	13	14	15	16	17	18
L3,L4	3,3	4,4	5,5	4,3	3,4	5,3	3,5	5,4	4,5
condition 28	false	false	false	true	false	true	false	true	false
ICCS1(NCU)	13	19	25	-	14	-	15	-	20
MG	30	31	32	-	33	-	34	-	35
J	5	6	7	-	9	-	11	-	13
INO(J)	0	0	0	-	0	-	0	-	0
NU	26	26	26	-	27	-	27	-	27
MV	1	2	3	-	6	-	3	-	4
I	31	32	33	-	34	-	35	-	36
J	247	248	249	-	6	-	3	-	4
NOM34(J)	1	1	1	-	1	-	1	-	1
FG_IRR	9.5E-2	9.5E-2	9.5E-2	-	-7.8E-2	-	-7.8E-2	-	-7.8E-2
ZO34(J)	1.0	1.0	1.0	-	1.0	-	1.0	-	1.0
FG_RED(I)	9.5E-2	9.5E-2	9.5E-2	-	-7.8E-2	-	-7.8E-2	-	-7.8E-2

Table 4.37. Ammonia CS 6 and 7. Note that for CS=7 the condition 28 in table 4.34 is true and the instructions from 29 to 46 are not executed.

Achronims

Achronim	inf	Description
MVF	inf(2)	Total number of symmetry operators of the group
LAF	inf(20)	Total number of shells
NAF	inf(24)	Total number of atoms in the unit cell
MAX_ATOM_COUPLES		Total number of atom couples
MAX_G_COUPLES	inf(191)	Total number of shell couples
MVLAF	inf(56)	Total number of couple sets
LA34F	inf(73)	Total number of type of independent couples of shells (CT)
ICOF	inf(133)	Number of type of couple sets
NSTAR		Number of stars in which the lattice vectors are subdivided
MGFDMF	inf(145)	Maximum number of **g** vectors allocated on each CT
LSTDMF	inf(37)	Maximum number of **g** vectors per couple set
NGIDMF	inf(39)	Maximum number of irreducible **g** vectors
MGFBI		Total number of **g** for each CST originated by symmetry from the irreducible set of **g** vectors (NGDMF×MVF)
NFGT	inf(11)	Size of the lower half part of the Fock matrix (LIM021)
ICONTA	inf(19)	Size of the irreducible part of the Fock matrix
NPGT	inf(12)	Size of the density matrix (LIM022)

Table 4.38. Numerical quantities defined by CRYSTAL and used in the present subroutine description; inf is the label by which these quantities are stored in the code.

Bibliography

[1] Dovesi, R.; Roetti, C.; Civalleri, B.; Orlando, R.; Saunders, V. *Ab-initio quantum simulation in solid state chemistry*, Vol. 21 of *Rewiews in computational chemistry - Chapiter 1;* John Wiley and Sons, Inc, New York - Editors Kenny B. Lipkowitz, Raima Larter, Thomas R. Cundari, 2005.

[2] La polarizzazione spontanea nei materiali ferroelettrici. Dall'Olio, S.; Dovesi, R. Tesi di laurea, AA. 1995/1996.

[3] Pisani, C.; Dovesi, R.; Roetti, C. *Hartree-Fock ab-initio treatment of crystallyne systems;* Springer-Verlag, 1988.

[4] Crystal 2003 user's manual. Saunders, V. R.; Dovesi, R.; Roetti, C.; Orlando, R.; Zigovich-Wilson, C. M.; Harrison, N. M.; Doll, K.; Civalleri, B.; Bush, I. J.; D'arco, P.; Llunell, M. `www.crystal.unito.it/Manuals/crystal06.pdf`.

[5] Pisani, C.; Dovesi, R.; Roetti, C.; Causà, M.; Orlando, R.; Casassa, S.; Saunders, V. *Int. J. Quantum Chem.* **2000**, *77*, 1032–1048.

[6] Pisani, C. *Quantum-Mechanical Ab-initio calculation of the properties of crystalline materials;* Springer, 1994.

[7] Kohn, W.; Beke, A. D.; Parr, R. G. *J. Phys. Chem.* **1996**, *100*, 12974.

[8] Hoenberg, P.; Kohn, W. *Phys. Rev. B* **1964**, *136*, 864.

[9] Kohn, W.; Sham, L. J. *Phys. Rev. A* **1965**, *140*, 1133.

[10] Il legame ad idrogeno in sistemi molecolari e cristallini: studio computazionale ab-initio. Civalleri, B. Tesi di Dottorato, 1998/1999.

[11] Dispense per il corso di chimica computazionale. C. Roetti, B. C. Unpublished.

[12] Saunders, V. R.; Freyria, C.; Dovesi, R.; Salasco, L.; Roetti, C. *Mol. Phys.* **1992**, *77*, 629.