

*New Advances in Theoretical Understanding of Rare-Earth Emission.  
Fundamentals for Applications by Rational Property Design.  
Research Report, PCE-1881, Phase 1.*

**From models to materials. Mutations and extensions in DFT & LF (Density Functional Theory and Ligand Field) modeling strategies.**

**The working frame:** The body of the project, in the subsequently outlined data, continues our structural chemistry quests<sup>1</sup> using previous pioneering methodological advances,<sup>2,3</sup> applications<sup>4</sup> and prospects<sup>5,6</sup> concerning the magnetic and optical properties of lanthanide (Ln) compounds. As pointed in the initial project, the electronic configuration of lanthanides can be characterized as non-*aufbau*, with unpaired f-type electrons (determining spectra and magnetism) placed in lower energy orbitals, below many doubly occupied ones, a very non-routine situation, demanding special multi-configurational Wave Function Theory (WFT, see ref.[1] chapter 2) procedures. However, many interesting materials belong to solid-state regime, where the available methods are tributary to the Density Functional Theory (DFT, see ref.[1] chapter 3).

The frontal use of DFT in rare-earth compounds is not rigorously allowed,<sup>Error! Bookmark not defined.</sup> because most Ln ions have quasi-degenerate states (i.e. outside of founding DFT premises). Then, must use a combination of numeric experiments, where DFT is yet usable to provide parameters for the so-called Ligand Field (LF) phenomenological models, which are subsequently employed to simulate the spectra. The applications challenge the audacious aim of property-engineering for materials active as phosphors (scintillators), usable in energy-saving domestic lighting or as laser emitters.

### *AI.1 Non-standard plane-wave (PW) calculations by fractional populations of f-type bands.*

Aiming to work in the Plane-Waves (PW) based DFT, specific to solid-state problems, we will proceed systematically, assessing the quality of pseudo-potential data in accounting specific spectral parameters. This quest is a new perspective, since usually the pseudopotentials are tested by reproducing cell geometries or thermodynamic quantities. In this view we selected one<sup>7</sup> among the few complete repositories of lanthanide pseudopotentials. We then extracted from the large black box of each lanthanide pseudopotential data the components corresponding to all-electron (AE), core projected-augmented wavefunctions (PAW) and the pseudopotential tail (PS), fitting each by Slater-Type Orbital (STO) primitives. Thus, each radial profile of interest (labelled by  $n$  and  $l$  quantum numbers), picked up from the AE part of pseudopotential files, for each lanthanide element, is expanded as linear combination of six STO primitives (with  $c_i$  coefficients,  $i=1:6$ ). The STO components are characterized by exponential parameter ( $\zeta$ ) and the radial order  $k$  (related to a radial  $r^{k-1}$  preexponential factor).

The results for the 4f and 5d shells of the whole lanthanide series are comprised in Table 1. Having STO-based definition of orbitals, one may to advance towards analytical formulas of the Slater-Condon parameters of the valence shell. The general two-electron atomic integral can be ascribed, in terms of different ( $n, l$ ) quantum number couples of the four component orbitals and the  $k$ -th moment of the multipolar electrostatic function, as the following integral.

$$R^k(n_a l_a, n_b l_b, n_c l_c, n_d l_d) = \int_{r_1=0}^{\infty} \int_{r_2=0}^{\infty} R_{n_a l_a}(r_1) R_{n_c l_c}(r_1) R_{n_b l_b}(r_2) R_{n_d l_d}(r_2) \frac{\min(r_1, r_2)^k}{\max(r_1, r_2)^{k+1}} r_1^2 r_2^2 dr_1 dr_2 \quad (1)$$

For the spectral problems expected in the f-f and f-d spectra of lanthanide ions, there are the following integrals: intra-shell  $F^k(n_a l_a) = R^k(n_a l_a, n_a l_a, n_a l_a, n_a l_a)$ , for  $n_a l_a = 4f$  altogether with  $k=0, 2, 4, 6$  indices, while  $n_a l_a = 5d$  coupled with the  $k=0, 2, 4$  series; inter-shell  $F^k(n_a l_a, n_b l_b) = R^k(n_a l_a, n_a l_a, n_b l_b, n_b l_b)$  particularized for the 4f-5d case, with  $k=0, 2, 4$ ; inter-shell  $G^k(n_a l_a, n_b l_b) = R^k(n_a l_a, n_b l_b, n_a l_a, n_b l_b)$ , running on  $k=1, 3$  for the 4f-5d pair. The general  $R^k$  integrals can be rescaled to  $R_k = R^k D_k$ , with factors chosen to avoid fractional coefficients in the subsequent use in spectral terms formulas (switching to  $F_k$  and  $G_k$ , by corresponding factorization of  $F^k$  and  $G^k$ ). Using orbitals expanded in a general series of STOs, computing the particular  $F^k$  and  $G^k$  integrals implies the estimation of general  $R^k$  terms with Slater-type primitives as product integrand. The calculation of such quantities is documented in early literature,<sup>8</sup> in a style tributary to the dawn of computational quantum chemistry, old computers and codes. In order to be in line with the new computational frame, we prepared our own algorithms, running in *matlab-octave*, with analytical formulas obtained by the help of *Mathematica* algebraic facilities.

**Table 1.** The exponential parameters ( $\zeta_i$ ) and combination coefficients ( $c_i$ ) for the STO primitives fitting the 4f (left side) and 5d (right side) shells from discussed pseudopotential data.<sup>7</sup> The header line corresponds to the shell and radial order ( $k$ ) of the STO components. The  $c_0$  and  $c_q$  lines stand for neutral and tri-positive Ln.

Ln		4f						5d						
		$k$	4	4	4	5	5	6	3	4	5	5	5	5
Ce	$\zeta$		13.69	7.481	3.219	3.723	1.867	1.255	16.114	5.946	5.947	2.804	1.562	0.923
	$c_0$		0.057	0.349	1.674	-1.013	0.124	0.016	0.096	-0.404	0.185	0.499	0.506	0.167
	$c_q$		0.064	0.380	1.500	-0.823	0.050	0.002	0.128	-0.504	0.184	0.773	0.321	0.014
Pr	$\zeta$		14.13	7.785	3.372	3.898	1.966	1.330	16.042	7.043	3.015	1.755	1.124	0.772
	$c_0$		0.057	0.350	1.679	-1.019	0.119	0.015	0.098	-0.243	0.446	0.464	0.234	0.048
	$c_q$		0.072	0.363	1.783	-1.089	-0.020	-0.160	0.132	-0.315	0.590	0.589	-0.470	0.565
Nd	$\zeta$		14.54	8.070	3.508	4.055	2.051	1.392	16.376	7.257	3.090	1.804	1.159	0.788
	$c_0$		0.057	0.352	1.685	-1.027	0.115	0.015	0.097	-0.238	0.436	0.455	0.247	0.059
	$c_q$		0.068	0.358	1.779	-1.092	0.003	-0.211	0.130	-0.308	0.577	0.595	-0.424	0.521
Pm	$\zeta$		14.93	8.342	3.634	4.199	2.127	1.446	16.712	7.468	3.162	1.845	1.189	0.797
	$c_0$		0.058	0.354	1.692	-1.036	0.112	0.014	0.096	-0.232	0.427	0.448	0.260	0.068
	$c_q$		0.067	0.371	1.755	-1.047	-0.055	-0.005	0.134	-0.301	0.564	0.599	-0.382	0.486
Sm	$\zeta$		15.31	8.604	3.751	4.333	2.196	1.493	17.04	7.678	3.232	1.162	0.855	1.229
	$c_0$		0.058	0.356	1.699	-1.045	0.110	0.014	0.095	-0.227	0.418	0.443	0.272	0.076
	$c_q$		0.063	0.370	1.662	-0.978	0.005	-0.205	0.130	-0.293	0.553	0.598	-0.344	0.460
Eu	$\zeta$		15.66	8.856	3.860	4.459	2.259	1.534	17.381	7.888	3.297	1.924	1.223	0.800
	$c_0$		0.059	0.359	1.705	-1.054	0.109	0.014	0.093	-0.221	0.410	0.441	0.282	0.081
	$c_q$		0.062	0.389	1.606	-0.905	-0.055	-0.012	0.062	0.389	1.606	-0.905	-0.055	-0.012
Gd	$\zeta$		16.01	9.100	3.964	4.578	2.316	1.571	17.713	8.097	3.360	1.956	1.234	0.798
	$c_0$		0.059	0.361	1.711	-1.063	0.107	0.014	0.091	-0.216	0.402	0.439	0.292	0.087
	$c_q$		0.062	0.379	1.626	-0.950	0.012	-0.194	0.122	-0.285	0.516	0.642	-0.312	0.404
Tb	$\zeta$		16.34	9.335	4.061	4.689	2.369	1.603	18.042	8.306	3.421	1.984	1.241	0.794
	$c_0$		0.060	0.364	1.717	-1.071	0.106	0.014	0.090	-0.210	0.395	0.438	0.301	0.092
	$c_q$		0.062	0.382	1.621	-0.951	0.017	-0.211	0.129	-0.269	0.525	0.588	-0.273	0.445
Dy	$\zeta$		16.66	9.561	4.153	4.795	2.417	1.632	18.365	8.517	3.479	2.008	1.246	0.789
	$c_0$		0.061	0.367	1.722	-1.079	0.105	0.014	0.088	-0.205	0.388	0.438	0.309	0.096
	$c_q$		0.062	0.390	1.581	-0.912	0.010	-0.198	0.133	-0.256	0.525	0.559	-0.249	0.474
Ho	$\zeta$		16.96	9.779	4.239	4.895	2.461	1.657	18.675	8.731	3.532	2.024	1.244	0.780
	$c_0$		0.063	0.371	1.726	-1.087	0.104	0.014	0.086	-0.199	0.383	0.440	0.315	0.099
	$c_q$		0.063	0.383	1.611	-0.959	0.035	-0.271	0.119	-0.255	0.514	0.576	-0.244	0.460
Er	$\zeta$		17.25	9.989	4.321	4.989	2.501	1.679	18.982	8.945	3.584	2.041	1.243	0.772
	$c_0$		0.064	0.374	1.730	-1.094	0.103	0.014	0.085	-0.194	0.377	0.441	0.322	0.104
	$c_q$		0.063	0.389	1.578	-0.931	0.033	-0.275	0.119	-0.248	0.509	0.570	-0.232	0.469
Tm	$\zeta$		17.53	10.191	4.398	5.077	2.537	1.698	19.284	9.161	3.636	2.057	1.242	0.764
	$c_0$		0.066	0.378	1.733	-1.100	0.102	0.014	0.083	-0.189	0.372	0.442	0.329	0.108
	$c_q$		0.064	0.390	1.578	-0.940	0.041	-0.298	0.123	-0.239	0.502	0.568	-0.224	0.479
Yb	$\zeta$		17.79	10.384	4.470	5.160	2.570	1.714	19.575	9.380	3.684	2.070	1.239	0.754
	$c_0$		0.067	0.381	1.735	-1.106	0.101	0.014	0.081	-0.183	0.366	0.443	0.335	0.113
	$c_q$		0.066	0.404	1.587	-0.933	0.031	-0.072	0.118	-0.235	0.499	0.564	-0.214	0.485
Lu	$\zeta$		18.04	10.569	4.538	5.239	2.599	1.727	19.854	9.603	3.731	2.081	1.234	0.745
	$c_0$		0.070	0.385	1.737	-1.112	0.100	0.014	0.080	-0.178	0.362	0.443	0.341	0.118
	$c_q$		0.068	0.407	1.573	-0.928	0.010	-0.213	0.108	-0.227	0.494	0.555	-0.203	0.500

Without entering all the details of the performed analysis, we point that we fitted separately the PAW-type inner bodies and PS outer tails, confining each component to three STO primitives, with respective  $n$ -parameters given in the first and second halves of the Table 1 header, for f and d shells. After obtaining the STO exponentials, based on dense-grid data from pseudo-potential files, one may keep these fixed, adjusting only the coefficients, in the approach of other quantities rendered on poorer radial definitions. Then, the problem becomes a linear algebra procedure, performing the pseudoinverse defined with the help of known primitives, estimated on the available grid points (multiplying then the vector containing the values of the fitted profiles, to get the coefficients). In this way, we fitted the PW results of the Ln(III) triple positive ions, taken with f and d shells in the fractional population schemes, smearing the total number of active electrons equally over the degenerate orbitals. The f radial profiles for ions were indirectly obtained from density differences and isotropic averaging on the 3D maps of  $\alpha$  and  $\beta$  type, for the  $f^n$  and  $f^{n-1}d^1$  configurations.

**Methodologies for adjusting interelectron parameters in DFT+U techniques.** With the above shell STO primitives it is possible to calculate “on the flight” the Slater-Condon parameters, based on averaged radial functions. This gives rise to an interesting opening. Thus, one may estimate quantities like  $U$  and  $J$  needed in the DFT+U theories,<sup>9,10</sup> quite from first principles, without the need to make them adjustable parameters. We rely on the DFT+U formulation implemented in Quantum Espresso, based on the next formula, with  $l$  either d or f-type, termed in  $\alpha$  and  $\beta$  shell populations:

$$E(l) = (1/2) \left( p_{\alpha}^{[l]} + p_{\beta}^{[l]} \right) \left( p_{\alpha}^{[l]} + p_{\beta}^{[l]} - 1 \right) U_{ll} - (1/2) \left( p_{\alpha}^{[l]} \left( p_{\alpha}^{[l]} - 1 \right) + p_{\beta}^{[l]} \left( p_{\beta}^{[l]} - 1 \right) \right) \cdot J_{ll} \quad (2)$$

The relations with the Slater-Condon parameters, for the d and f shells are

$$U_{dd} = F_0^{dd}, \quad J_{dd} = (7/2)(F_2^{dd} + 9F_4^{dd}) \quad (3)$$

$$U_{ff} = F_0^{ff}, \quad J_{ff} = 10F_2^{ff} + 33F_4^{ff} + 286F_6^{ff} \quad (4)$$

The standard wisdom is to select  $U$  and  $J$  empirically, by reproducing correctly general structural features, such as crystallographic data (geometries) or thermodynamic quantities, in certain benchmarks of simple lattices. However, this approach may be prone to uncontrolled parametric balances, yielding good overall results, while the mutual ratios of active factors may be improper. Here we propose a first principle manner to set a DFT+U calculations, using  $U$  and  $J$  values obtained from the pseudo-potential files. Since is expected that, in compounds, the lanthanides are effectively in ionized status, is reasonable to use the values estimated for ions.

The actual DFT+U schemes allow to apply this correction only on a single shell, d or f. We formulate here the possibility of adding simultaneously two-shell increments, deriving the following form of the inter-shell DFT+U correction, added to the sum of  $E(d)$  and  $E(f)$  terms from eq. (2):

$$E(df) = (p_{\alpha}^d + p_{\beta}^d)(p_{\alpha}^f + p_{\beta}^f)(U_{df} - (1/2)J_{df}) - (1/2)(p_{\alpha}^d - p_{\beta}^d)(p_{\alpha}^f - p_{\beta}^f) \cdot J_{df}, \quad (5)$$

$$\text{where } U_{df} = F_0^{df}, \quad J_{df} = 3G_1^{df} + 12G_3^{df} + 66G_5^{df}. \quad (6)$$

An extensive tabulation of the parameters of the generalized DFT+U is shown in Table 2.

**Table 2.** The DFT+U parameters for 4f, 5d and inter-shell df schemes for neutral and ionized lanthanides.

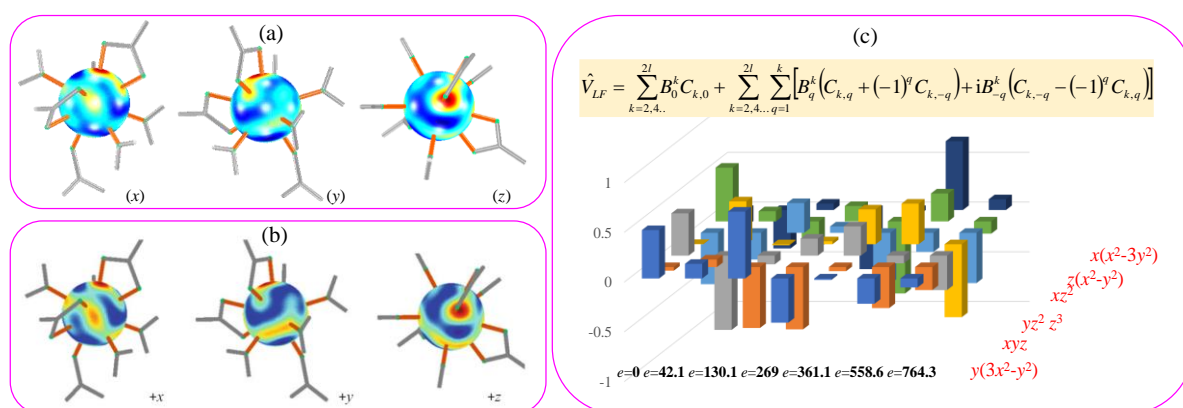
Ln	$q=0$						$q=\pm 3$					
	$U_{ff}$	$U_{dd}$	$U_{fd}$	$J_{ff}$	$J_{dd}$	$J_{fd}$	$U_f$	$U_d$	$U_{fd}$	$J_f$	$J_d$	$J_{fd}$
Ce	22.023	8.247	10.598	0.836	0.492	0.334	24.427	10.487	13.015	0.920	0.676	0.274
Pr	21.936	8.451	10.897	0.855	0.500	0.368	26.290	9.666	13.032	1.016	0.557	0.277
Nd	22.792	8.538	11.089	0.894	0.501	0.362	26.580	9.924	13.220	1.024	0.573	0.268
Pm	22.882	8.606	11.228	0.906	0.501	0.375	29.167	10.095	13.756	1.142	0.580	0.279
Sm	23.057	8.667	11.405	0.921	0.495	0.395	28.375	7.732	10.998	1.094	0.390	0.156
Eu	22.683	8.711	11.505	0.913	0.491	0.421	30.935	19.254	23.494	1.209	1.074	1.680
Gd	23.877	8.724	11.565	0.958	0.494	0.387	30.262	10.649	14.158	1.167	0.616	0.244
Tb	24.467	8.735	11.604	0.983	0.492	0.394	30.867	10.279	14.008	1.188	0.577	0.242
Dy	24.800	8.707	11.607	1.000	0.486	0.399	31.852	9.974	13.934	1.226	0.547	0.240
Ho	24.959	8.733	11.687	1.008	0.484	0.414	31.332	10.092	13.863	1.194	0.556	0.225
Er	24.819	8.722	11.732	1.006	0.477	0.426	31.950	9.985	13.853	1.216	0.544	0.220
Tm	25.013	8.697	11.743	1.016	0.472	0.432	32.146	9.870	13.775	1.218	0.531	0.214
Yb	24.817	8.667	11.764	1.011	0.465	0.441	35.864	9.784	14.264	1.378	0.523	0.256
Lu	26.078	8.124	11.167	1.058	0.428	0.393	35.429	9.556	13.905	1.357	0.504	0.213

The code implementation of a two-shell DFT+U scheme is difficult, beyond our current reach and formulated objectives. On the other hand, this is a valuable finding and collateral new way to go.

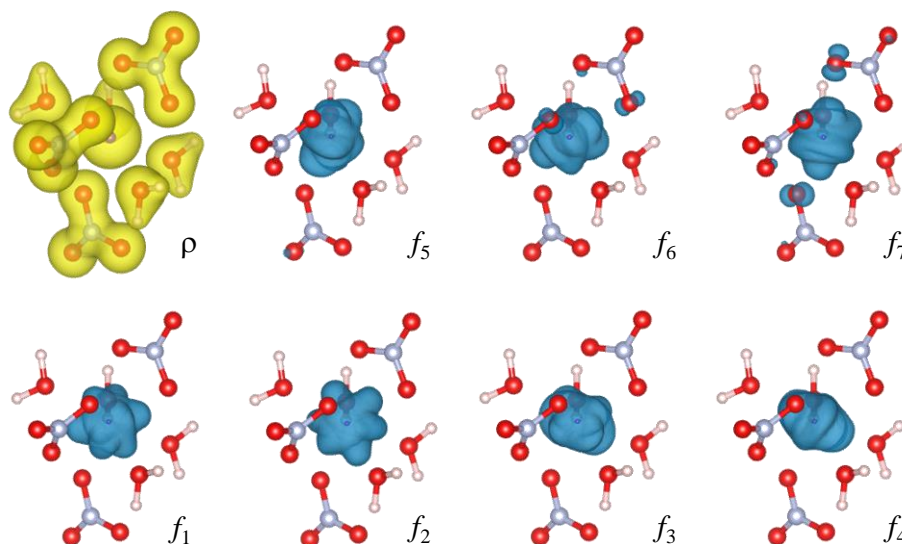
**Ligand Field parameters from PW-DFT versus WFT calculations. Band calculations using VASP, Abinit or related codes.** The basic assumption of the Ligand Field (LF) theories is that one may account selected problems as described in atomic-like basis (d or f shells), even though, realistically, these can be molecular orbitals (keeping, however, the dominant contribution of d or f metal ion AOs). Since the LF problems show a general multi-configurational nature, it is then a challenge to account them by DFT, intrinsically confined to single-configuration groundstates. The technical answer to this issue is taking as reference a population averaged scheme that has the meaning of an average of configurations (AOC) in a WFT frame, which also meets the concept of energy barycenter, tacitly employed in LF schemes (with respect of which the split of metal ion AOs occurs). In this view, one may use in PW codes the less common input controls of band populations, which in the VASP<sup>11</sup> code consist in FERWE

and FERDO keywords, in AbInit<sup>12</sup> by OCC directive and in Quantum Espresso<sup>13</sup> by OCCUPATIONS group, followed by corresponding list of generally fractional values. For the test level of procedures, aiming to extract LF parameters, we selected a simple neutral coordination unit, [Tb(NO<sub>3</sub>)<sub>3</sub>(H<sub>2</sub>O)<sub>4</sub>] isomorphous with a reported erbium system.<sup>14</sup>

The orbital energies are usually believed to represent the LF schemes. However, this is not true. In turn, we propose a series of orbitally controlled DFT calculations, enforcing non-*aufbau* configurations. For a  $f^8 = 7f_\alpha + 1f_\beta$  system, like the Tb(III) complex, one may emulate the LF split of the <sup>7</sup>F term by running the  $\beta$  electron among in the distinct orbitals of f-type. The resulting seven energies empirically correspond to the LF split of the <sup>7</sup>F spectral term of Tb(III) ion, which is also isomorphic with the LF one-electron effective eigenvalues. The energies resulted from the above designed run are, {0.0, 42.1, 130.1, 269.0, 361.1, 558.6, 764.3}, in cm<sup>-1</sup>. Remarkably, the range corresponds well to the energy spectrum expected in a LF problem. We collected the decomposition of plane wave eigenvalues in localized spherical harmonics. The coefficients are not detailed here, the matrix being schematized as histogram in Figure 1c. This matrix (almost unitary) can be used to obtain the Ligand Field Hamiltonian, applying it to the back-transformation of a diagonal matrix containing the computed configuration energies. Furthermore, the resulted LF matrix can be decomposed in the  $B_q^k$  parameters, active in the LF Hamiltonian, which is also compressed in Figure 1.



**Figure 1.** The LF analysis of the [Tb(NO<sub>3</sub>)<sub>3</sub>(H<sub>2</sub>O)<sub>4</sub>] complex unit. (a) the LF color map from PW-DFT treatment; (b) the LF colormap from WFT multiconfigurational calculation. (c) graphical synopsis of the LF master equation and the transformation coefficients (shown as histogram) used to obtain the LF matrix.



**Figure 2.** Density maps from PW-DFT calculations on Tb(NO<sub>3</sub>)<sub>3</sub>(H<sub>2</sub>O)<sub>4</sub>] complex. Upper left corner: the total density. The  $f_1$ - $f_7$  panels correspond to  $\beta$  density contours (at 0.001e/Å<sup>3</sup> isosurfaces) of the f-type orbitals with fractional occupation, in the population-averaged calculation taken as reference for the Ligand Field modeling.

The outlined procedure yielded the following  $B_q^k$  LF parameters: { -41.576, 201.538, 182.554, -366.682, 87.990 } for  $k=2$  and  $q=-2$  to 2 respective series, { -70.911, 72.417, 379.741, 401.616, 337.363, -222.676, -87.808, 122.655, 120.024 } for  $k=4$  and  $q=-4$  to 4, and { -78.638, -291.534, -70.613, 84.791, 146.176, 338.771, 594.390, -276.876, 300.443, -245.698, -182.858, 25.542, 296.757 } for the for  $k=6$  and  $q=-6$  to 6, with all values in  $\text{cm}^{-1}$ . The numerical values of individual parameters are not directly transparent, in meaning. We suggest a picturesque interpretation, with the help of color maps of the Ligand Field potential, as shown in the panel (a) of Figure 1. It is worth noting that the aspect of PW-DFT emulated LF potential map is quite similar to those obtained from multi-configuration WFT calculations, namely the technique appropriate for the LF problem, validating in this way the above outlined DFT-based original procedure.

The WFT potential is represented in the panel (b) of Figure 1. In spite of the resemblance between panels (a) and (b) in Fig.1, the parameters look quite different. Thus, the WFT based  $B_q^k$  lists are: { 32.4, -112.0, 72.0, -140.3, 104.7 } for the  $k=2$  set, { -17.0, 56.4, 401.7, 341.8, 100.2, -177.7, -56.9, 112.1, 155.8 } for  $k=4$  and { -66.1, -190.2, -39.5, 60.6, 125.1, 211.9, 385.1, -211.2, 210.9, -203.9, -138.6, 44.7, 245.7 } for the  $k=6$  (values in  $\text{cm}^{-1}$ ). One notes then that different LF decomposition schemes may be nominally distinct and yet globally similar.

For the sake of completeness, we emulated the orbitals taken as “active” in the PW-DFT treatment (see Figure 2). The snippets are not properly orbitals, but density differences engineered to yield the pattern of the  $\beta$  electron in f-type MOs. Therefore, the sign of function is not obtained. Noticing contours with six or eight lobes, suggesting that the accommodation of the  $\beta$  electron occurs in f-type MOs, one validates the adopted procedure.

### A. 1.2 Elaborating new Ligand Field models

Whether for the Ligand Field treatment on the d or f shells there are consecrated codes, like respectively CAMMAG<sup>15</sup> and BONNMAG,<sup>16</sup> the df double-shell treatment is yet an advanced deal of customized handling inside of few groups, including ours, here making steps toward the release of a stable procedure in such an analysis. Certain technicalities are explained in the second part of this section and in the attached Appendix. The generalization of one-electron Ligand Field part to a two-shell frame is not that difficult, conceptually. In the spirit of Angular Overlap Model,<sup>3,17</sup> one must take the known master formula, employing projection factors,  $D_{i\lambda}^{[l]}$  and,  $D_{j\lambda}^{[l']}$ , coming from different quantum numbers,  $l$  and  $l'$ , as well as a new set of inter-shell parameters,  $e_{[l,l']\lambda}^L$ , aside those ascribed inside each shell type ( $e_{[l,l]\lambda}^L$  and  $e_{[l',l']\lambda}^L$ ):

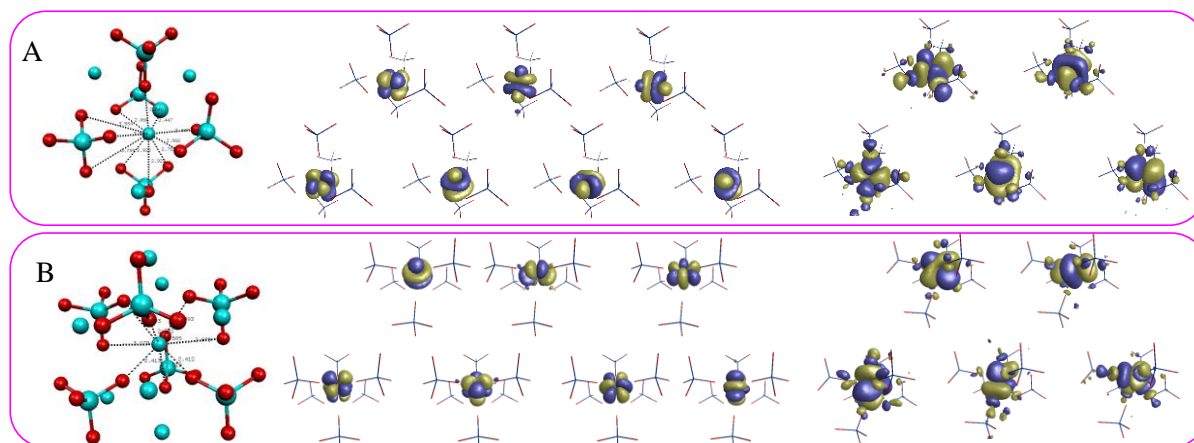
$$\langle [l]_i | V_{\text{AOM}} | [l']_j \rangle = \sum_L \sum_{\lambda=\sigma,\pi_L,\pi_{||}} D_{i\lambda}^{[l]}(\theta_L, \phi_L, \psi_L) D_{j\lambda}^{[l']}(\theta_L, \phi_L, \psi_L) e_{[l,l']\lambda}^L \quad (7)$$

In the technique of Wybourne parameterization<sup>18</sup>, i.e. the formula briefed in panel (c) of figure (1), the generalization is also straightforward, accepting a new branch of inter-shell  $B_q^k$  parameters running on odd  $k$  indices,  $k=1,3$  and 5. However, insofar, because the f and d shells are mixing mutually very little, one may drop the inter-shell explicit parameterization for one-electron LF Hamiltonian. Then, the main technical difficulty in the LF phenomenology of df spectra is the account of inter-shell two-electron part, worked algebraically as suggested in the end section and implemented as detailed in the appendices.

In the following we will illustrate the treatment of a realistic system, modeling Eu(II) ions doped in  $\text{Ba}_2\text{SO}_4$  a lattice. First, the crystal is optimized within a PW-DFT calculation, then representative clusters are excerpted and submitted to WFT ab initio treatments, yielding spectral terms, subsequently fitted with a customized LF model. Starting from the  $\text{Ba}_2\text{SiO}_4$  lattice, the doping of the Eu(II) was simulated taking rectangular cells containing one Eu(II) and seven Ba(II) sites. This would correspond to rather high concentration, 12.5%Eu, but, at the same time, sufficient for neglecting the inter-lanthanide coupling. The band calculations (with geometry optimization including cell dimensions and ion positions) were done by VASP, in unrestricted frame (spin polarized option), taking care that the magnetic moment at lanthanide sites corresponds to the  $f^7$  configuration.

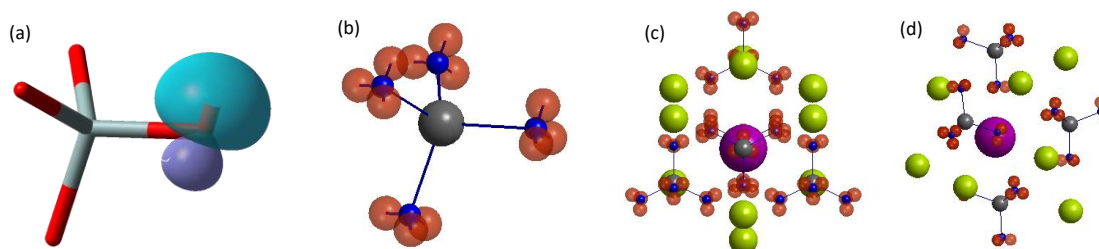
There are two sites, which can be suggestively described as follows: **A-** with Eu(II) surrounded by five  $\text{SiO}_4^{4-}$  units, approximately placed as a trigonal bipyramid; **B-** with Eu(II) coordinated by six  $\text{SiO}_4^{4-}$  units, approximately placed as an octahedron. The  $\text{Eu}_A$  site is surrounded by six Ba ions

(belonging to the type B of environment), approximately making a trigonal prism with the mean axis aligned to those of the quasi-trigonal axis of the bipyramid-alike placement of the anions. The cationic environment of the  $\text{Eu}_B$  site consists in six Ba ions of A-type, with vertices approximately above the edges of the octahedron sketched by anionic bodies. Besides, almost at the same radius, must inscribe two more cations, belonging to the B-type site, roughly capping the octahedron of the A-type cations.



**Figure 3.** Synopsis of the modelling of Eu(II) doped in  $\text{Ba}_2\text{SO}_4$  lattice. A and B panels, the two crystallographic sites. Left side: optimized geometries; Middle part: WFT computed f-type orbitals; Right side: the d-type MOs.

In these structures, the  $\text{SiO}_4^{2-}$  units act as monodentate ligand ( $\text{Eu-OSiO}_3$ ), approximately along a Si-O bond), as bidentate chelates ( $\text{EuO}_2\text{SiO}_2$ ), or tridentate ( $\text{EuO}_3\text{SiO}$ ). Certain situations can be regarded as intermediate between bidentate and tridentate. With the obtained local environments, considered as molecular clusters, we go to further modelling, namely the WFT-type approach by CASSCF (Complete Active Space Self Consistent Field) calculations. For site A, we considered the  $[\text{EuBa}_6(\text{SiO}_4)_5]^{4+}$  fragment, the positive charge not impinging essentially on the results. For the B type, we took the  $[\text{EuCs}_2\text{Ba}_6(\text{SiO}_4)_6]^{4+}$  unit, where we considered Cs ions instead of barium sites, in order to keep the total charge of the cluster the same as in the case A.



**Figure 4.** Synopsis of the superposition LF model. (a) a lone pair (LP) on silicate moiety; (b) the schematized assembly of lone pairs on the  $\text{SiO}_4^{4-}$  unit, to be parameterized as point-charge; (c) and (d) different views for the system of ligators for the  $\text{Eu}[\text{Ba}_6(\text{SiO}_4)_5]$  cluster representing the the A-type site. Each ball represents a ligator (blue-oxygen site, red-lone pairs) or anti-ligatator (gray- Si-centers, green the Ba ions).

We propose assembling the silicate unit from point-charge components bearing explicit distance dependence with respect of the coordination site. The ligators placed at the position of oxygen atoms (first type of ligator), at the barycenters of each of the three lone pairs around an oxygen center (LP=the second type of ligator). The anti-ligators (bearing positive charge) are the silicon atom and barium ions. We use the term “ligator” to distinguish the dichotomized components from the notion of “ligand”, as a defined chemical entity. The contributions from different ligators are taken additive:

$$\hat{V}_{LF} = \sum_L^{\text{Ligators}} \hat{V}_{LF}(L) = \sum_{k=0}^{2l} \sum_{q=-k}^k a_k(L) \frac{4\pi}{2k+1} Y_{k,q}(\theta_L, \phi_L) \sum_L^{\text{Ligators}} Y_{k,q}^*(\theta_L, \phi_L) \quad (8)$$

The Figure 4 shows the components of the designed modeling, confined to the site A. The ligators are introduced as simple  $a_2/R^3 + a_4/R^5 + a_6/R^7$  dependences in the f shell and  $b_2/R^3 + b_4/R^5$  for the d shell. In first instance, one may neglect the fd nondiagonal block. The fitted parameters for the four proposed ligators (O center, LP lone pair, Si and Ba atoms) are given in Table 3.

**Table 3.** Ligator-based superposition LF model fitting the WFT computed spectral terms on site A

	$R_0$	$a_2$	$a_4$	$a_6$	$b_2$	$b_4$
O	2.5	4715.8	970.8	185.9	24416.0	28564.0
LP	2.5	-1673.3	-245.5	-53.2	-21469.0	-6706.3
Si	3.25	-4206.2	181.1	-94.5	-84822.0	155.4
Ba	3.6	-5511.6	266.0	-35.4	-150790.0	2566.1

The overall result of the fit is good, emulating the following eigenvalues of the fd Ligand Field matrix: {0, 76.5, 95.6, 231.9, 270.5, 354.5, 377.5, 20200, 20943, 23609, 25301, 28838}, comparable with the WFT computation: {0, 70.8, 75.9, 194.6, 269.7, 365.4, 371.5, 20215, 20607, 23198, 25003, 28864} (all values in  $\text{cm}^{-1}$ ). However, it is somewhat strange that we obtained negative parameters for the LP ligators. Possibly the system is over-parameterized (e.g. the O ligators were sufficient, without appended explicit lone pairs), but we shall check later the issue. Although realistic and complex, the results are not completely satisfactory, remaining to dig into technicalities. An improvement of the fit can be, in principle, attained if consider the  $tk$  power radial of dependence  $a_k(R_0/R)^k$  as matter of optimization, instead of fixed  $tk=\{3,5,7\}$  integers for respective  $k=2,4,6$  terms.

**Elaborating and testing fortran, matlab-octave or Mathematica programs for handling data and multi-shell Ligand Field (LF) models.** Our work is heavily based on writing our own codes in different environments, ranging from classical fortran varieties (gfortran or f90), up to advanced Mathematica symbolic algebra. For instance, although we skip certain details, a powerful piece of Mathematica program is illustrated below, this accomplishing what is needed to generate the f-d type two-electron integrals in terms of Slater-Condon parameters:

```

Do[{monopos[i, j_] = Max[i, j] + ((Max[i, j] - 1)/2) + Min[i, j], b[pos[i, j_]]{k, l_] = monopos[monopos[i, j], monopos[k, l]]}],
{L, m, t, f_] = SphericalHarmonicY[1, m, t, f]; Vc[L, m, t, f_] = (-1)^m*SphericalHarmonicY[1, -m, t, f]
sh={2,3};shn={"d","f"};ns=Length[sh]
Do[Do[If[{iorb[io] == 1 || iorb[jo] == 1 || iorb[lo] == 1} && {iorb[io] == 2 || iorb[jo] == 2 || iorb[ko] == 2 || iorb[lo] == 2},
{is = iorb[io], mi = morb[io], js = iorb[jo], mj = morb[jo], ks = iorb[ko], mk = morb[ko], ls = iorb[lo], ml = morb[lo],
biel[io, jo][ko, lo] = Expand[Sum[If[monopos[is, js] >= monopos[ks, ls], nMF[Min[is, js], Max[is, js]][Min[ks, ls], Max[ks, ls]][1],
nMF[Min[ks, ls], Max[ks, ls]][Min[is, js], Max[is, js]][1]]*(4*(Pi/(2*1 + 1)))*If[-mi + mj - m == 0 && -mk + ml + m == 0,
Simplify[Integrate[Integrate[Vc[sh[is]], mi, 0, 1, 0, 1]*Vc[sh[js]], mj, 0, 1, 0, 1]*Vc[1, m, 0, 1, 0, 1], {0, 1, 0, 2*Pi}]*Sin[0], {0, 1, 0, Pi}]]*
Simplify[Integrate[Integrate[Vc[sh[ks]], mk, 0, 2, 0, 2]*Vc[sh[ls]], ml, 0, 2, 0, 2]*Vc[1, m, 0, 2, 0, 2], {0, 2, 0, 2*Pi}]*Sin[0], {0, 2, 0, Pi}]], 0],
{1, Min[Abs[sh[is]] - sh[js]], Abs[sh[ks]] - sh[ls]], Max[sh[is] + sh[js], sh[ks] + sh[ls]]], {m, -1, 1}], pver = 1, If[biel[io, jo][ko, lo] == 0, pver = 0],
Null, If[pver == 1, Print[" ", shn[is], " ", mi, "; ", shn[ks], " ", mk, "; ", shn[js], " ", mj, "; ", shn[ls], " ", ml, "; ", biel[io, jo][ko, lo]]], {io, 1, no},
{jo, 1, no}], {ko, 1, no}, {lo, 1, no}

```

Properly handled, this output is integrated in other codes. In this way, we use high-level programming in writing other pieces of codes. The output of the previous computer algebra coding was practically generating the numerical subroutines illustrated in the fortran90 code discussed in the following.

Very frequently we use the convenient frame of Matlab or Octave, benefiting from built-in procedures (e.g. eigenvalues, minimization, graphical facilities). In Appendix 1 we illustrate a part of a matlab-octave subroutine performing the calculations of the  $R^k(n_{ala}, n_{blb}, n_{ala}, n_{blb})$  quantities, discussed around equation (1), needed to compute the Slater-Condon parameters from STO components (e.g those from Table 1) and subsequently estimate the quantities from Table 2.

In Appendix 2 we illustrate the core of a Fortran 90 program performing the calculation of Ligand Field (LF) spectral terms, that covers all the d, f and fd possible configurations. This implements the new developments of phenomenological two-shell LF modelling. The LF one-electron matrix (for one shell, d or f, or extended two-shell format) is introduced not as a numeric block, but by announcing the number of free parameters and, after reading them, providing the matrices of coefficients for each LF parameter. If want to simply introduce LF entirely numeric, then put 1.0 as sole parameter and provide the brute matrix (the lower diagonal part).

In this form, the input file is set.inp. The first line reads one of the LFtype=dd, LFtype=ff or LFtype=fd directives. If none of these options is retrieved, the program stops. Then, the number of electrons is taken, followed by the minimal and maximal spin projections, ne, msmin, msmax, this option deciding the set of Slater determinants basis. For instance, 2 1.0 1.0 selects only the high spin ( $S_z=1$ ) configurations with two electrons. If want all, must put 2 -1.0 1.0 (i.e.  $S_z=-1,0,1$ ).

In the present implementation, the code generates practically the full CAS-type (Complete Active Space) stack of configurations (merging the sets with defined range of  $S_z$  projections). This means that, for the fd case, one may get very many configurations. In practical problems, starting from a  $f^n$  groundstate, only  $f^{(n-1)d1}$  situations are to be accounted. Limitations in configuration interaction



basis can be introduced in further editions. In the present extended form of  $(fd)^n$  CAS-type, must add the F0dd, F2dd and F4dd two-electron parameters of the upper orbitals, while these will be not needed in the  $f^{(n-1)d1}$  situation (since no inter-electron effects inside the 5d shell are met). The complete list of the two-electron parameters for the fd case is now: F0ff, F2ff, F4ff, F6ff, F0fd, F2fd, F4fd, F6fd, G1fd, G3fd G5fd, F0dd, F2dd, F4dd. Note that the F0ff, F0fd, F0dd parametric sequence and the (hd-hf) one electron gap between shells are not independent. The user should be aware about this issue.

This code is a general configuration interaction (CI) program, generating with Slater rules, the matrix elements of the total Hamiltonian. The particularization to the d, f or fd problems is handled with three “Case” options reading one of the ddbiel, ffbiel or fdbiel routines, with corresponding coefficients of the bi-electronic parameters. Because the lists of fixed coefficients introducing the integrals in the CI are quite lengthy, their printing is omitted in the given Appendix 2. If, instead of using predefined d, f and fd cases, reading the two-electron parameters and their structure-coefficients from input line, one may use this program for more general configuration interactions problems.

### Conclusions (*Executive Summary*)

The project delineates this year phase by the title “**From models to materials. Mutations and extensions in DFT & LF (Density Functional Theory and Ligand Field) modelling strategies.**” incorporating a complex spectrum of topics and actions. Must point that it plays on different specialization areas, whose borders are rarely meet together. Thus, we are mostly specialized in molecular quantum chemical calculations, done with Gaussian or Slater type orbitals (GTOs and STOs), but we audaciously stepped in the domain of Plane Wave (PW) bases, mastered by solid-state physicists. We approached the domain with critical intentions, claiming that Density Functional Theory (DFT), the only class of available methods in PW frame, is not well suited for lanthanide spectroscopy topics. With such strong claims, we had to take a precautionary sweep of the methodological environment, with great cost in time and brain power. We did a thorough assessment of the pseudo-potential (PS) techniques used in PW-DFT, translating these building blocks into STO-based functions and estimating integrals specific to Wave-Function Theory (WFT) and Ligand Field (LF) paradigms. These objectives are subsumed to the following activities: *<Non-standard plane-wave (PW) calculations by fractional populations of f-type and s. Ligand Field parameters from PW-DFT versus WFT calculations. Band calculations using VASP, Abinit or related codes (Quantum Espresso)>*

To cover the alleged methodological lack, we worked continuously outside routine. Thus, aside using various codes of molecular and solid-state calculations, we did a large amount of programming in different environments: Mathematica, Matlab-Octave and Fortran. Given the non-trivial deal of coding, this outcome counts as a firm deliverable, although not destined to immediate dissemination, partly by legitimate reasons of keeping certain technological secrets. In this year, we wrote and tested many original codes in the mentioned programming languages, amounting a total of thousands of lines. These developments are summarized in the annual report in the section *<A. 1.2 Elaborating new Ligand Field models/ Elaborating and testing fortran, matlab-octave or Mathematica programs for handling data and multi-shell Ligand Field (LF) models>*. Given the deep foundations built in order to initialize de project, we did not aim research papers in this year. However, during the past months, we achieved about 80% from the realization of a paper devoted to the assessment of PW-DFT/PS methodologies in DFT approach of lanthanide compounds and approximately 20% from an applied series of calculations on the LF rationalization of optical properties of lanthanide-based nitrates and silicates.

### Deliverables and Dissemination

- ◆ The large amount and the difficulty of codes developed along the project’s methodological goals (written in Mathematica, Matlab-Octave and Fortran) counts as a firm delivered outcome, the achievement validation being proven in the extended research report.
- ◆ Dissemination at conferences: 1 keynote (F. Cimpoesu) at ICAMANA2022, November 22-24th, 2022, Louvain-la-Neuve, Belgium; 1 oral presentation (F. Cimpoesu) at IBWAP 2022, Constanța, 2 poster presentations (A. Toader and C. Buta) at IBWAP 2022, Constanța.
- ◆ Dissemination by publication, not scheduled in this phase, mentioning, however, several articles in progress (one in 80% degree of finalization, another about 20% and two others in preliminary setting).





**Appendix 1.** Part of matlab-octave subroutine computing ingredients for evaluating Slater-Condon generalized parameters. The printing is truncated to the  $k=6$  maximal order

```

function rkg=rgksto1(bas1,bas2)

ncomp1=length(bas1(:,1));
ncomp2=length(bas2(:,1));
lshell1=bas1(1,2);
lshell2=bas2(1,2);
lmin=abs(lshell1-lshell2);
lmax=(lshell1+lshell2);
kont=0;
for k=lmin:2:lmax
kont=kont+1;

for ka1=1:ncomp1
for kb1=1:ncomp2
for ka2=1:ncomp1
for kb2=1:ncomp2
na1=bas1(ka1,1);la1=bas1(ka1,2);za1=bas1(ka1,3);
nb1=bas2(kb1,1);lb1=bas2(kb1,2);zb1=bas2(kb1,3);
na2=bas1(ka2,1);la2=bas1(ka2,2);za2=bas1(ka2,3);
nb2=bas2(kb2,1);lb2=bas2(kb2,2);zb2=bas2(kb2,3);

f1=sqrt((2*za1)^(2*na1+1)/factorial(2*na1))*sqrt
((2*zb1)^(2*nb1+1)/factorial(2*nb1));
f2=sqrt((2*za2)^(2*na2+1)/factorial(2*na2))*sqrt
((2*zb2)^(2*nb2+1)/factorial(2*nb2));

if na1+nb1>na2+nb2
in1=na1+nb1; in2=na2+nb2;
a1=za1+zb1; a2=za2+zb2;
nln2k=strcat(int2str(in1),'_',int2str(in2),'_',i
nt2str(k));
else
in2=na1+nb1; in1=na2+nb2;
a2=za1+zb1; a1=za2+zb2;
nln2k=strcat(int2str(in1),'_',int2str(in2),'_',i
nt2str(k));
end

switch nln2k
case '2 2 0'
Rk=-
(2*(a1^2+3*a1*a2+a2^2))/(a1^2*a2^2*(a1+a2)^3);
case '2 2 1'
Rk=-6/(a1*a2*(a1+a2)^3);
case '3 2 0'
Rk=-
(2*(3*a1^3+12*a1^2*a2+8*a1*a2^2+2*a2^3))/(a1^3*a
2^2*(a1+a2)^4);
case '3 2 1'
Rk=-
(6*(4*a1+a2))/(a1^2*a2*(a1+a2)^4);
case '3 3 0'
Rk=-
(12*(a1^4+5*a1^3*a2+10*a1^2*a2^2+5*a1*a2^3+...
a2^4))/(a1^3*a2^3*(a1+a2)^5);
case '3 3 1'
Rk=-
(24*(a1^2+5*a1*a2+a2^2))/(a1^2*a2^2*(a1+a2)^5);
case '3 3 2'
Rk=-120/(a1*a2*(a1+a2)^5);
case '4 2 0'
Rk=-
(12*(2*a1^4+10*a1^3*a2+10*a1^2*a2^2+5*a1*a2^3+a2
^4))/(a1^4*a2^2*(a1+a2)^5);
case '4 2 1'
Rk=-
(12*(10*a1^2+5*a1*a2+a2^2))/(a1^3*a2*(a1+a2)^5);
case '4 3 0'
Rk=-
(12*(4*a1^5+24*a1^4*a2+60*a1^3*a2^2+...
45*a1^2*a2^3+18*a1*a2^4+3*a2^5))/(a1^4*a2^3*(a1+
a2)^6);
case '4 3 1'
Rk=-
(24*(5*a1^3+30*a1^2*a2+12*a1*a2^2+2*a2^3))/(a1^3
*a2^2*(a1+a2)^6);
case '4 3 2'
Rk=-
(120*(6*a1+a2))/(a1^2*a2*(a1+a2)^6);
case '4 4 0'
Rk=-
(144*(a1^6+7*a1^5*a2+21*a1^4*a2^2+35*a1^3*a2^3+.
.
21*a1^2*a2^4+7*a1*a2^5+a2^6))/(a1^4*a2^4*(a1+a2)
^7);
case '4 4 1'
Rk=-
(240*(a1^4+7*a1^3*a2+21*a1^2*a2^2+7*a1*a2^3+...
a2^4))/(a1^3*a2^3*(a1+a2)^7);
case '4 4 2'
Rk=-
(720*(a1^2+7*a1*a2+a2^2))/(a1^2*a2^2*(a1+a2)^7);
case '4 4 3'
case '6 4 1'

Rk=5040/(a1*a2*(a1+a2)^7);
case '5 2 0'
Rk=-
(24*(5*a1^5+30*a1^4*a2+40*a1^3*a2^2+...
30*a1^2*a2^3+12*a1*a2^4+2*a2^5))/(a1^5*a2^2*(a1
+a2)^6);
case '5 2 1'
Rk=-
(36*(20*a1^3+15*a1^2*a2+6*a1*a2^2+a2^3))/(a1^4*
a2*(a1+a2)^6);
case '5 3 0'
Rk=-
(48*(5*a1^6+35*a1^5*a2+105*a1^4*a2^2+...
105*a1^3*a2^3+63*a1^2*a2^4+21*a1*a2^5+3*a2^6))/
(a1^5*a2^3*(a1+a2)^7);
case '5 3 1'
Rk=-
(144*(5*a1^4+35*a1^3*a2+21*a1^2*a2^2+7*a1*a2^3+
a2^4))/(a1^4*a2^2*(a1+a2)^7);
case '5 3 2'
Rk=-
(240*(21*a1^2+7*a1*a2+a2^2))/(a1^3*a2*(a1+a2)^7
);
case '5 4 0'
Rk=-
(144*(5*a1^7+40*a1^6*a2+140*a1^5*a2^2+...
280*a1^4*a2^3+224*a1^3*a2^4+112*a1^2*a2^5+32*a1
*a2^6+4*a2^7))/(a1^5*a2^4*(a1+a2)^8);
case '5 4 1'
Rk=-
(720*(2*a1^5+16*a1^4*a2+56*a1^3*a2^2+...
28*a1^2*a2^3+8*a1*a2^4+a2^5))/(a1^4*a2^3*(a1+a2
)^8);
case '5 4 2'
Rk=-
(720*(7*a1^3+56*a1^2*a2+16*a1*a2^2+2*a2^3))/(a1
^3*a2^2*(a1+a2)^8);
case '5 4 3'
Rk=-
(5040*(8*a1+a2))/(a1^2*a2*(a1+a2)^8);
case '5 5 0'
Rk=-
(2880*(a1^8+9*a1^7*a2+36*a1^6*a2^2+84*a1^5*a2^3
+...
126*a1^4*a2^4+84*a1^3*a2^5+36*a1^2*a2^6+9*a1*a2
^7+a2^8))/(a1^5*a2^5*(a1+a2)^9);
case '5 5 1'
Rk=-
(4320*(a1^6+9*a1^5*a2+36*a1^4*a2^2+84*a1^3*a2^3
+...
36*a1^2*a2^4+9*a1*a2^5+a2^6))/(a1^4*a2^4*(a1+a2
)^9);
case '5 5 2'
Rk=-
(10080*(a1^4+9*a1^3*a2+36*a1^2*a2^2+9*a1*a2^3+a
2^4))/(a1^3*a2^3*(a1+a2)^9);
case '5 5 3'
Rk=-
(40320*(a1^2+9*a1*a2+a2^2))/(a1^2*a2^2*(a1+a2)^
9);
case '5 5 4'
Rk=-362880/(a1*a2*(a1+a2)^9);
case '6 2 0'
Rk=-
(240*(3*a1^6+21*a1^5*a2+35*a1^4*a2^2+...
35*a1^3*a2^3+21*a1^2*a2^4+7*a1*a2^5+a2^6))/(a1^
6*a2^2*(a1+a2)^7);
case '6 2 1'
Rk=-
(144*(35*a1^4+35*a1^3*a2+21*a1^2*a2^2+7*a1*a2^3
+a2^4))/(a1^5*a2*(a1+a2)^7);
case '6 3 0'
Rk=-
(720*(2*a1^7+16*a1^6*a2+56*a1^5*a2^2+...
70*a1^4*a2^3+56*a1^3*a2^4+28*a1^2*a2^5+8*a1*a2^
6+a2^7))/(a1^6*a2^3*(a1+a2)^8);
case '6 3 1'
Rk=-
(144*(35*a1^5+280*a1^4*a2+224*a1^3*a2^2+...
112*a1^2*a2^3+32*a1*a2^4+4*a2^5))/(a1^5*a2^2*(a
1+a2)^8);
case '6 3 2'
Rk=-
(720*(56*a1^3+28*a1^2*a2+8*a1*a2^2+a2^3))/(a1^4
*a2*(a1+a2)^8);
case '6 4 0'
Rk=-
(1440*(3*a1^8+27*a1^7*a2+108*a1^6*a2^2+...
252*a1^5*a2^3+252*a1^4*a2^4+168*a1^3*a2^5+72*a1
^2*a2^6+...
18*a1*a2^7+2*a2^8))/(a1^6*a2^4*(a1+a2)^9);
case '6 6 0'

```

<pre> Rk= (1440*(7*a1^6+63*a1^5*a2+252*a1^4*a2^2+... 168*a1^3*a2^3+72*a1^2*a2^4+18*a1*a2^5+2*a2^6))/( a1^5*a2^3*(a1+a2)^9); case '6 4 2' Rk=- (1440*(28*a1^4+252*a1^3*a2+108*a1^2*a2^2+... 27*a1*a2^3+3*a2^4))/(a1^4*a2^2*(a1+a2)^9); case '6 4 3' Rk=- (10080*(36*a1^2+9*a1*a2+a2^2))/(a1^3*a2*(a1+a2)^ 9); case '6 5 0' Rk=- (2880*(6*a1^9+60*a1^8*a2+270*a1^7*a2^2+... 720*a1^6*a2^3+1260*a1^5*a2^4+1050*a1^4*a2^5+600* a1^3*a2^6+... 225*a1^2*a2^7+50*a1*a2^8+5*a2^9))/(a1^6*a2^5*(a1 a2)^10); case '6 5 1' Rk=- (4320*(7*a1^7+70*a1^6*a2+315*a1^5*a2^2+... 840*a1^4*a2^3+480*a1^3*a2^4+180*a1^2*a2^5+40*a1* a2^6+... 4*a2^7))/(a1^5*a2^4*(a1+a2)^10); case '6 5 2' Rk=- (10080*(8*a1^5+80*a1^4*a2+360*a1^3*a2^2+... 135*a1^2*a2^3+30*a1*a2^4+3*a2^5))/(a1^4*a2^3*(a1 a2)^10); case '6 5 3' Rk=- (40320*(9*a1^3+90*a1^2*a2+20*a1*a2^2+2*a2^3))/(a 1^3*a2^2*(a1+a2)^10); case '6 5 4' Rk=- (362880*(10*a1+a2))/(a1^2*a2*(a1+a2)^10); </pre>	<pre> Rk= (86400*(a1^10+11*a1^9*a2+55*a1^8*a2^2+... 165*a1^7*a2^3+330*a1^6*a2^4+462*a1^5*a2^5+330*a 1^4*a2^6+... 165*a1^3*a2^7+55*a1^2*a2^8+11*a1*a2^9+a2^10))/( a1^6*a2^6*(a1+a2)^11); case '6 6 1' Rk=- (120960*(a1^8+11*a1^7*a2+55*a1^6*a2^2+... 165*a1^5*a2^3+330*a1^4*a2^4+165*a1^3*a2^5+55*a1 ^2*a2^6+... 11*a1*a2^7+a2^8))/(a1^5*a2^5*(a1+a2)^11); case '6 6 2' Rk=- (241920*(a1^6+11*a1^5*a2+55*a1^4*a2^2+... 165*a1^3*a2^3+55*a1^2*a2^4+11*a1*a2^5+a2^6))/(a 1^4*a2^4*(a1+a2)^11); case '6 6 3' Rk=- (725760*(a1^4+11*a1^3*a2+55*a1^2*a2^2+... 11*a1*a2^3+a2^4))/(a1^3*a2^3*(a1+a2)^11); case '6 6 4' Rk=- (362880*(a1^2+11*a1*a2+a2^2))/(a1^2*a2^2*(a1+a 2)^11); case '6 6 5' Rk=-39916800/(a1*a2*(a1+a2)^11); otherwise Rk=0; end  rkg(ka1, kb1, ka2, kb2, kont)=f1*f2*Rk;  end end end end  end % k </pre>
--	--

**Appendix 2.** Fortran 90 code performing general Ligand Field (LF) Configuration Interaction (CI) calculation of spectral terms for single shell (f or d) and double shell fd cases. The Input consist in LF one-electron matrices and lists of corresponding Slater-Condon parameters.

<pre> PROGRAM CI_LF  ! Temporary Input Format: ! Read from set.inp ! First line must contain one of the ! following statements ! LFtype=dd, LFtype=ff or LFtype=fd ! Second line gives the number of electrons ! and the domain of spin projections ! ne, mspin, msmax ! Third line gives the number of Ligand Field !(LF) (aka one electron parameters) ! nLFpar ! Fourth line gives the row of LF parameters ! Fifth line gives the row of two electron ! (Slater-Condon) parameters !(according to the sets from the first line) ! The next lines are looping for nLFpar times ! to read the coefficients of ! each LF parameter, ! considering the the low-diagonal ! format of the LF matrix  Implicit none Integer :: i,j,k,l,m Integer :: k1,l1,k2,l2,k11,k12 Integer :: monopos,bipos,spintype Integer :: ne,dg,ncfgs,ncfga,ncfgb, status Real :: mspin,msmax,ms Integer :: ks,lfstat Integer,Dimension(0:30) :: Fact Integer,Allocatable :: comba(:,,:),combb(:,,:), configs(:,,:) Integer,Allocatable :: nms(:),nalpna(:),nbeta(:) Character(2) :: LFtype Character(80) :: line Integer :: nLFpar, ntwoel, ntwoelint Real rho Real,Allocatable :: LFpar(:),twoelpar(:),Wparam(:),Hrow(:),E(:), H(:,,:),C(:,,:) Real,Allocatable :: VLF(:,,:),LFdef(:,,:),twoelcf(:,,:,:), twoelints(:,,:,:)  Select Case (LFtype) Case('dd') Write(*,*) "START dd" </pre>	<pre> Fact(0) = 1 Do m=1,30 Fact(m)=m*Fact(m-1) End do  Open(1,file='set.inp') read(1,'(a80)') line read(1,*) ne, mspin, msmax write(*,*) ne, mspin, msmax call tocap(line) lfstat=0 LFtype=' '  if (lfstat == 0) then call searchk(line,'LFTYPE=DD',lfstat) if (lfstat==1) then LFtype='dd' endif endif write(*,*) lfstat if (lfstat == 0) then call searchk(line,'LFTYPE=FF',lfstat) if (lfstat==1) then LFtype='ff' endif endif  if (lfstat == 0) then call searchk(line,'LFTYPE=FD',lfstat) if (lfstat==1) then LFtype='fd' endif endif  if (lfstat == 0) then call searchk(line,'LFTYPE=DF',lfstat) if (lfstat==1) then LFtype='fd' endif endif  write(*,*) lfstat write(*,*) line ncfga=Fact(dg)/(Fact(nalpna(ks))*Fact (dg-nalpna(ks))) ncfgb=Fact(dg)/(Fact(nbeta(ks))*Fact (dg-nbeta(ks))) </pre>
---	--



```

dg=5; ntwoel=3;
Write(*,*) "START ff"
Case('ff')
dg=7; ntwoel=4;
Case('fd')
Write(*,*) "START fd"
dg=12; ntwoel=13;
Case(' ')
Write(*,*) "STOP"
Stop
End Select

Read(1,*) nLFpar
Write(*,*) nLFpar
Allocate (LFpar(nLFpar))
Allocate (LFdef(dg*(dg+1)/2,nLFpar))
Allocate (VLF(dg,dg))
Allocate (twoelpar(ntwoel))
Allocate (twoelcf(dg,dg,dg,dg,ntwoel))
Allocate (twoelints(dg,dg,dg,dg))

Read(1,*) (LFpar(i),i=1,nLFpar)
Write(*,*) (LFpar(i),i=1,nLFpar)
Read(1,*) (twoelpar(i),i=1,ntwoel)
Write(*,*) (twoelpar(i),i=1,ntwoel)

do m=1,nLFpar
do i=1,dg
read(1,*) (LFdef(i*(i-1)/2+j,m),j=1,i)
write(*,*) (LFdef(i*(i-1)/2+j,m),j=1,i)
end do !i
end do !m

Close(1)

write(*,*)
VLF=0
do i=1,dg
do j=1,i
do m=1,nLFpar
VLF(i,j)=VLF(i,j)+LFdef(i*(i-1)/2+j,m)*LFpar(m)
end do !m
VLF(j,i)=VLF(i,j)
end do !j
end do !i

!do i=1,dg
!write(*,*) (VLF(i,j),j=1,dg)
!end do !i

Select Case (LFTYPE)
Case('dd')
Write(*,*) "START dd"
call ddbiel(twoelcf)
Case('ff')
Write(*,*) "START ff"
call ffbiel(twoelcf)
Case('fd')
Write(*,*) "START fd"
call fdbiel(twoelcf)
Case(' ')
Write(*,*) "STOP"
Stop
End Select

twoelints=0
do m=1,ntwoel
twoelints=twoelints+twoelcf(:, :, :, m)*twoelpar(m)
end do

Allocate (nalpha(int(msmax-msmin+1)))
Allocate ( nbeta(int(msmax-msmin+1)))

ncfgs=0
do ks=1,int(msmax-msmin+1)
ms=msmin+(ks-1)

nalpha(ks)= int((2*ms + ne)/2.)
nbeta(ks)= int((-ms + ne/2.))
ncfga=Fact(dg)/(Fact(nalpa(ks))*Fact
(dg-nalpa(ks)))
ncfgb=Fact(dg)/(Fact( nbeta(ks))*Fact
(dg- nbeta(ks)))
ncfgs=ncfgs+ncfga*ncfgb
end do ! ks
write(*,*) "no.of. configurations ",ncfgs

Allocate (configs(ncfgs, ne),STAT=Status)
configs=0

k=0
do ks=1,int(msmax-msmin+1)

Subroutine combinari(ns, ne, ncfg, comb)
Integer :: ns, ne, ncfg
Integer :: nm, i, j, VM, cfgindex, status

```

```

! write(*,*) ks, ms, nalpa(ks),nbeta(ks),
ncfga,ncfgb

Allocate (comba(ncfga,nalpa(ks)),STAT=Status)
Call combinari(dg, nalpa(ks),ncfga, comba)

Allocate (combb(ncfgb,nbeta(ks)),STAT=Status)
Call combinari(dg, nbeta(ks),ncfgb, combb)

do i=1,ncfga
do j=1,ncfgb
k=k+1
configs(k,1:nalpa(ks))=comba(i,:)
configs(k,nalpa(ks)+1:nalpa(ks)+nbeta(ks))
=dg+combb(j,:)
end do !j
end do !i

Deallocate (comba)
Deallocate (combb)

end do ! ks

Do i=1,ncfgs
write(*,fmt='(50I3)') (configs(i,j),j=1,ne)
end do

Allocate (Hrow(ncfgs*(ncfgs+1)/2))
call SlaterRules(Hrow,dg, ncfgs, ne, configs,
nLFpar, ntwoel, vlf, twoelints)

!do i=1,ncfgs
!write(*,*) (Hrow(i*(i-1)/2+j),j=1,i)
!end do !i

Allocate (E(ncfgs))
Allocate (H(ncfgs,ncfgs))
Allocate (C(ncfgs,ncfgs))

do i=1,ncfgs
do j=1,i
H(i,j)=Hrow(i*(i-1)/2+j)
H(j,i)=H(i,j)
enddo
enddo

rho=0.0000001
call eign(ncfgs,rho,H,C,E)
write(*,*) "E"
write(*,fmt='(10F15.2)') E
E=E-minval(E)
write(*,*) "E.rel"
write(*,fmt='(10F15.2)') E

End Program CISlaterRules

Function monopos(k, l)
Integer :: k,l ! input
Integer :: monopos ! output

If (k <= l) Then
monopos=1*(l-1)/2+k
Else
monopos=k*(k-1)/2+1
End If
End Function monopos

Function bipos(k1, l1, k2, l2)
Integer :: k1,l1,k2,l2,k11,k12 ! input
Integer :: monopos,bipos ! output

k11=monopos(k1,l1)
k12=monopos(k2,l2)
bipos=monopos(k11,k12)

End Function bipos

Function phasesign(X)
phasesign=X/2-Int(X/2)
End Function

Function spintype(dg, i)
Integer :: dg, i, spintype
If (i <= dg) Then
spintype = 1
Else
spintype = -1
End If
End Function spintype

Integer :: combs(ncfgs, ne)
Real :: monoel(dg,dg), biel(dg,dg,dg,dg),
Hrow(ncfgs*(ncfgs+1)/2)

```



```

Integer, Dimension(0:ne):: V
Integer, Dimension(ncfg,ne) :: comb
V=0
comb=0
mm = 0
cfgindex = 0
i = 1
j=0
24 V=0
VM = ns + 1 - i - ne
mm = 0
GoTo 21
22 V(j)=V(j)+1
21 Continue
Call addcomb(mm, cfgindex, ne, i, ncfg,comb, V)

mm = 0
Do j = ne, 2, -1
If (V(j) < VM) GoTo 22
mm = 1
V(j) = V(j - 1)
End do !j
mm = 0
i = i + 1
! Deallocate(V, STAT=Status)
V=0
If (VM == 0) Return
GoTo 24
End Subroutine combinari

Subroutine addcomb(mm, cfgindex, ne, i, ncfg,
comb, V)
Integer :: l, mm, cfgindex, ne, i, ncfg
Integer, Dimension(0:ne) :: V
Integer, Dimension(ncfg,ne) :: comb
If (mm == 1) return
cfgindex = cfgindex + 1
Do l = 1, ne
comb(cfgindex, l) = i - 1 + l + V(l)
end do
End Subroutine addcomb

Subroutine tocap(line)
integer i, k, ncars
character*(*) line
character(120) line1
ncars=len(line)
do i=1,ncars
j = ichar(line(i:i))
if (j.ge.97.and.j.le.122) line(i:i) = char(j-
32)
end do
k=0
do i=1,ncars
if( line(i:i).ne.' ') then
k=k+1
line1(k:k)=line(i:i)
endif
end do
line(1:k)=line1(1:k)
do i=k+1,ncars
line(i:i)=' '
end do
return
End Subroutine tocap

subroutine searchk(line,name,istat)
character*(*) line,name
integer istat,lenl, lenn
character*80 caps
lenline=len(line)
lennam=len(name)
istat=0
do i=1,len(line)-len(name)
if (line(i:i+len(name)).eq.name) then
istat=1
exit
endif
end do

End Subroutine search

Subroutine SlaterRules(Hrow,dg, ncfgs, ne, combs,
mmono, nbi, monoel, biel)
Integer :: i,j,k,l,m
Integer :: k1,l1,k2,l2,k11,k12
Integer :: monopos,bipos,spintype
Integer :: ne,dg,ncfgs, ncfga,ncfgb, status
Integer :: icfg, jcfg, sc, cnt, R, cphase
Integer :: Sindex(4), Rindex(4), ord(4)
Integer :: cfgi(2*dg), cfgj(2*dg)

Rindex(1) = Sindex(1); Rindex(2) = Abs(Sindex(2))
End If
ord(1) = 0

```

```

write(*,*) "enter slatru1"
!do i=1,dg
!write(*,*) (monoel(i,j),j=1,dg)
!end do !i

!Do i=1,ncfgs
!write(*,fmt='(50I3)') (combs(i,j),j=1,ne)
!end do

Hrow=0
do icfg = 1, ncfgs
do jcfg = 1, icfg
sc = 0; cnt = 0;
Sindex=0; Rindex=0; ord=0
cfgi=0; cfgj=0;

do l = 1, ne
cfgi(combs(icfg, l)) = 1
cfgj(combs(jcfg, l)) = 1
end do ! l

do l = 1, 2*dg
If (cfgi(l) /= cfgj(l)) Then
R = 1
If ((cfgi(l) - cfgj(l))<0) R = -1
cnt = cnt + 1
If (cnt > 4) Exit
Sindex(cnt) = R
EndIf
end do ! l

Select Case(cnt)
Case(0) ! diagonal CI element; First case
of Slater rules
! one electron part
do k = 1, 2*dg
If (cfgi(k) /= 0) Then
! "+ <"; orb$(K); "|f|"; orb$(K); ">";
sc = sc + 1
If (k > dg) Then
k1 = k - dg;
Else
k1 = k;
End If
!write(*,*) k1, monoel(k1, k1)
Hrow(icfg*(icfg-1)/2+jcfg) =
Hrow(icfg*(icfg-1)/2+jcfg) + monoel(k1, k1)

End If
end do ! k

! two-electron part
do k = 1, 2*dg
do m = k + 1, 2*dg

If (cfgi(k) /= 0 .and. cfgi(m) /= 0) Then
sc=sc+1
If (k > dg) Then
k1 = k - dg;
Else
k1 = k;
End If

If (m > dg) Then
m1 = m - dg;
Else
m1 = m;
End If

! Coulomb integral
Hrow(icfg*(icfg-1)/2+jcfg) =
Hrow(icfg*(icfg-1)/2+jcfg) +
biel(k1,m1,k1,m1)
If (spintype(dg, k) == spintype(dg, m))
Then
sc = sc + 1
! Exchange integral
Hrow(icfg*(icfg-1)/2+jcfg) =
Hrow(icfg*(icfg-1)/2+jcfg) -
biel(k1,m1,m1,k1)

End If
End If
end do !m
end do !k
Case(2) ! bra and ket differ by a pair of
orbitals ; Second case of Slater rules
sc = 0
If (Sindex(1) < 0) Then
Rindex(1) = Sindex(2); Rindex(2) =
Abs(Sindex(1))
Else
ord(1) = 0
Do
ord(1) = ord(1) + 1

```



```

Do
ord(1) = ord(1) + 1
If (combs(icfg, ord(1)) == Abs(Rindex(1)))
Exit
End Do
ord(2) = 0
Do
ord(2) = ord(2) + 1
If (combs(jcfg, ord(2)) == Abs(Rindex(2)))
Exit
End Do
If (Mod(Abs(ord(1) - ord(2)),2)==0) Then
cphase = 1
Else
cphase = -1
End If
If (Rindex(1) > dg) Then
k1 = Rindex(1) - dg
Else
k1 = Rindex(1)
End If
If (Rindex(2) > dg) Then
m1 = Rindex(2) - dg
Else
m1 = Rindex(2)
End If
If (spintype(dg, Rindex(1)) == spintype(dg,
Rindex(2))) Then
sc = sc + 1
! one electron part
! sign* "<"; orb$(Rindex(1)); "|f|";
orb$(Rindex(2)); ">";
Hrow(icfg*(icfg-1)/2+jcfg) = Hrow(icfg*(icfg-
1)/2+jcfg) + monoel(k1, m1)*cphase
End If
Do k = 1, 2*dg
If (k /= Rindex(1) .and. k /= Rindex(2)) Then
If (cfgi(k) /= 0 .and. cfgj(k) /= 0) Then
If (k > dg) Then
k2 = k - dg
Else
k2 = k
End If
If (spintype(dg, Rindex(1)) ==
spintype(dg, Rindex(2))) Then
sc = sc + 1
! sign* "<"; orb$(Rindex(1)); ",,"; orb$(K); "|g|";
orb$(Rindex(2)); ",,"; orb$(K); ">";
Hrow(icfg*(icfg-1)/2+jcfg) = Hrow(icfg*(icfg-
1)/2+jcfg) + biel(k1, k2, m1, k2)*cphase
End If
If (spintype(dg, Rindex(1)) == spintype(dg,
K)) Then
If (spintype(dg, Rindex(2)) == spintype(dg,
K)) Then
sc = sc + 1
! -sign* "<"; orb$(Rindex(1)); ",,"; orb$(K);
"|g|"; orb$(K); ",,"; orb$(Rindex(2)); ">";
Hrow(icfg*(icfg-1)/2+jcfg) = Hrow(icfg*(icfg-
1)/2+jcfg) - biel(k1, k2, m1, k2)*cphase
End If
End If
End If
End If
End do !k
Case(4)
! bra and ket differ by two orbital pairs ;
Case III in Slater rules
sc = 0
Do k = 1, 4
Do m = k+1, 4
If (Sindex(k) > 0 .and. Sindex(m) > 0) Then
Rindex(1) = Abs(min(Sindex(k), Sindex(m)));
Rindex(2) = Abs(max(Sindex(k), Sindex(m)))
Else
If (Sindex(k) < 0 .and. Sindex(m) < 0) Then
Rindex(3) = Abs(max(Sindex(k), Sindex(m)));
Rindex(4) = Abs(min(Sindex(k), Sindex(m)))
End If
End If
End do !m
End do !k

```

```

If (combs(icfg, ord(1)) == Abs(Rindex(1)))
Exit
End Do
ord(2) = 0
Do
ord(2) = ord(2) + 1
If (combs(icfg, ord(2)) == Abs(Rindex(2)))
Exit
ord(3) = 0
Do
ord(3) = ord(3) + 1
If (combs(jcfg, ord(3)) == Abs(Rindex(3)))
Exit
End Do
ord(4) = 0
Do
ord(4) = ord(4) + 1
If (combs(jcfg, ord(4)) == Abs(Rindex(4)))
Exit
End Do
k = Abs(ord(1) - ord(3)) + Abs(ord(2) -
ord(4))
If (mod(k,2) == 0) Then
cphase = 1
Else
cphase = -1
End If
If (Rindex(1) > dg) Then
k1 = Rindex(1) - dg
Else
k1 = Rindex(1)
End If
If (Rindex(2) > dg) Then
k2 = Rindex(2) - dg
Else
k2 = Rindex(2)
End If
If (Rindex(3) > dg) Then
m1 = Rindex(3) - dg
Else
m1 = Rindex(3)
End If
If (Rindex(4) > dg) Then
m2 = Rindex(4) - dg
Else
m2 = Rindex(4)
End If
If (spintype(dg, Rindex(1)) == spintype(dg,
Rindex(3))) Then
If (spintype(dg, Rindex(2)) == spintype(dg,
Rindex(4))) Then
sc = sc + 1
! sign* "<"; orb$(Rindex(1)); ",,";
orb$(Rindex(2)); "|g|"; orb$(Rindex(3)); ",,";
orb$(Rindex(4)); ">";
Hrow(icfg*(icfg-1)/2+jcfg) =
Hrow(icfg*(icfg-1)/2+jcfg) +
biel(k1, k2, m1, m2)*cphase
End If
End If
If (spintype(dg, Rindex(1)) == spintype(dg,
Rindex(4))) Then
If (spintype(dg, Rindex(2)) == spintype(dg,
Rindex(3))) Then
sc = sc + 1
! -sign* "<"; orb$(Rindex(1)); ",,";
orb$(Rindex(2)); "|g|"; orb$(Rindex(4)); ",,";
orb$(Rindex(3)); ">";
Hrow(icfg*(icfg-1)/2+jcfg) =
Hrow(icfg*(icfg-1)/2+jcfg) -
biel(k1, k2, m2, m1)*cphase
End If
End If
End Select
end do !jcfg
end do !icfg
End Subroutine SlaterRules

```



## References

- <sup>1</sup> Putz, M. V.; Cimpoesu, F.; Ferbinteanu, M. *Structural Chemistry, Principles, Methods, and Case Studies*. Springer: Cham, **2018**. (<http://doi.org/10.1007/978-3-319-55875-2>)
- <sup>2</sup> Toader, A. M.; Buta, M. C.; Maftei, D.; Putz, M. V.; Cimpoesu, F. *Int. J. Mol. Sci.*, **2019**;
- <sup>3</sup> Suta, M.; Cimpoesu, F.; Urland, W. *Coord. Chem. Rev.* **2021**, *441*, 213981;
- <sup>4</sup> (a) Paulovic, J.; Cimpoesu, F.; Ferbinteanu, M.; Hirao, K. *J. Am. Chem. Soc.* **2004**, *126*, 3321-3331; (b) Ferbinteanu, M.; Kajiwara, T.; Choi, K-Y.; Nojiri, H.; Nakamoto, A.; Kojima, N.; Cimpoesu, F.; Fujimura, Y.; Takaishi, S.; Yamashita, M. *J. Am. Chem. Soc.*, **2006**, *128*, 9008-9009; (c) Cimpoesu, F.; Dahan, F.; Ladera, S.; Ferbinteanu, M.; Costes, J.-P. *Inorg. Chem.*, **2012**, *51*(21), 11279-11293.
- <sup>5</sup> (a) Ramanantoanina, H.; Cimpoesu, F.; Gottel, C.; Sahnoun, M.; Herden, B.; Suta, M.; Wickleder, C.; Urland, W.; Daul, C. *Inorg. Chem.*, **2015**, *54*(17), 8319-8326; (b) Ramanantoanina, H.; Urland, W.; Herden, B.; Cimpoesu, F.; Daul, C. *Phys. Chem. Chem. Phys.*, **2015**, *17*(14), 9116-9125; (c) García-Fuente, A.; Cimpoesu, F.; Ramanantoanina, H.; Herden, B.; Daul, C.; Suta, M.; Wickleder, C.; Urland, W. *Chem. Phys. Lett.*, **2015**, *622*, 120-123; (d) Ramanantoanina, H.; Sahnoun, M.; Barbiero, A.; Ferbinteanu, M.; Cimpoesu, F. *Phys. Chem. Chem. Phys.*, **2015**, *17*(28), 18547-18557; (e) Ramanantoanina, H.; Urland, W.; García-Fuente, A.; Cimpoesu, F.; Daul, C. *Phys. Chem. Chem. Phys.*, **2014**, *16*, 14625-14634; (f) Ramanantoanina, H.; Urland, W.; Cimpoesu, F.; Daul, C. *Phys. Chem. Chem. Phys.*, **2014**, *16*, 12282-12290; (g) Ramanantoanina, H.; Urland, W.; García-Fuente, A.; Cimpoesu, F.; Daul, C. *Chem. Phys. Lett.*, **2013**, *588*, 260-266; (h) Ramanantoanina, H.; Urland, W.; Cimpoesu, F.; Daul, C. *Phys. Chem. Chem. Phys.*, **2013**, *15*, 13902-13910.
- <sup>6</sup> García-Fuente, A.; Baur, F.; Cimpoesu, F.; Vega, A.; Jüstel, T.; Urland, W. *Chem. Eur. J.* **2018**, *24*, 16276-16281.
- <sup>7</sup> Topsakal, M.; Wentzcovitch, R.M. *Comp. Mat. Sci.* **2014**, *95*, 263-270
- <sup>8</sup> Cowan, R.D. *The Theory of Atomic Structure and Spectra*. University of California Press: Berkeley, CA, USA, **1981**.
- <sup>9</sup> Lichtenstein, A. I.; Anisimov, V. I.; Zaanen, J. *Phys. Rev. B* **1995**, *52*, R5467.
- <sup>10</sup> Anisimov, V. I.; Aryasetiawan, F.; Lichtenstein, A. I. *J. Phys.: Cond. Matter.* **1997**, *9*(4), 767.
- <sup>11</sup> (a) Kresse, G.; Hafner, J. *J. Phys.: Condens. Matt.* **1994**, *6*, 8245; (b) Kresse, G.; Joubert, D. *Phys. Rev.* **1999**, *59*, 1758.
- <sup>12</sup> Amadon, B.; Anglade, P.-M.; Beuken, J.-M.; Bottin, F.; Boulanger, P.; Bruneval, F.; Caliste, D.; Caracas, R.; Cote, M.; Deutsch, T.; Genovese, L.; Ghosez, Ph.; Giantomassi, M.; Goedecker, S.; Hamann, D.R.; Hermet, P.; Jollet, F.; Jomard, G.; Leroux, S.; Mancini, M.; Mazevet, S.; Oliveira, M.J.T.; Onida, G.; Pouillon, Y.; Rangel, T.; Rignanese, G.-M.; Sangalli, D.; Shaltaf, R.; Torrent, M.; Verstraete, M.J.; Zerah, G.; Zwanziger, J.W. *Comp. Phys. Comm.* **2009**, *180*(12):2582.
- <sup>13</sup> Giannozzi, P.; Andreussi, O.; Brumme, T.; Bunau, O.; Nardelli, M.B.; Calandra, M.; Car, R.; Cavazzoni, C.; Ceresoli, D.; Cococcioni, M.; Colonna, N. *J. Phys.: Cond. Matt.* **2017**, *29*(46), 465901.
- <sup>14</sup> Ferbinteanu, M.; Kajiwara, T.; Cimpoesu, F.; Katagari, K.; Yamashita, M. *Polyhedron* **2007**, *26*:2069-2073.
- <sup>15</sup> Cruse, D. A.; Davies, J. E.; Gerloch, M.; Harding, J. H.; Mackey D. J.; McMeeking R. F. *CAMMAG*, Vol. Cambridge, **1983**.
- <sup>16</sup> Bronova, A.; Bredow, T.; Glaum, R.; Riley, M.J.; Urland, W. *BonnMag: Computer program for ligand-field analysis of  $f^n$  systems within the angular overlap model.* **2018**, 176-186.
- <sup>17</sup> Schönherr, T.; Atanasov, M.; Adamsky, H. *Angular Overlap Model* in Lever, A. B. P.; McCleverty, J. A.; Meyer, T. J. (eds.) *Comprehensive coordination chemistry II*, vol. 2. Elsevier, Oxford, **2003**, 443-455.
- <sup>18</sup> (a) Wybourne, B. G. *Spectroscopic Properties of Rare Earths*. Wiley Interscience, New York, **1965**; (b) Newman, D. J.; Ng, B. K. C. *Crystal Field Handbook*. Cambridge University Press: Cambridge, **2000**.