Notes about *off–site Spin–Orbit* **implementation** (by Dr. Ramón Cuadrado)

The following are some general notes with the aim to help future SIESTA developers about the main parts/subroutines of the code related with the *off–site Spin–Orbit* (OSSO) implementation.

General remarks:

- The most relevant subroutines related with the OSSO implementation have at the end of their names the sufix "`_offsiteSO`". Some of them has been copied from already SIESTA existing ones and modified and re–named accordingly. Note that **not all** the modified subroutines include the `_offsiteSO` suffix. That is due to that the changes therein were not enough to create a new subroutine. In those cases, internally, `if`(...) `then` ... `else` ... `endif` structures will be present if necessary.
- I will **only** write briefly about the subroutines/functions that SIESTA uses to perform an OSSO calculation, i.e., I will exclude any other former code's feautres not directly related with the OSSO.
- As useful help, if one wants to know specifically the related subroutines, functions, etc, where there are OSSO changes, it would be of paramount help the use of the bash *grep* command to check at once what was changed throughout the code. **Key words**: `_offsiteSO`, `OffSite`, `offsite`, `off-site`, `Off-Site`, and in addition it can be found some comment lines starting by "`CC RC`".
- It is important to note that even that the formulation of the on–site SO formalism could be similar in some aspects, in deep there are substantial differnces such as **the calculation of the total energy** or the number/kind of elements involved in the total Hamiltonian that in the present case (OSSO) are all of them complex not being so for the on–site case.
- For further information about the mathematical framework in which the current OSSO method is based on, it can be found in the original work of Dr. Ramón Cuadrado and Dr. Jorge Iribas Cerdá (*R. Cuadrado and J. I. Cerdá, J. Phys.: Condens. Matter 24, 086005, (2012) (DOI:10.1088/0953-8984/24/8/086005)*).

Main OSSO internal flow:

`siesta` driver subroutine calls `siesta_init` to declare the magnetic variables by means of calling `init_spin`. There, the variables that will control what is done or not depending if the OSSO is involved in the calculations will be set up by means of the following "*magnetic*" flags: `spin%none`, `spin%Col`, `spin%NCol`, `spin%SO`, `spin%SO_offsite`, depending on whether within the input file the user wants *no polarisation, spin–polarized, non–collinear, on–site* or *off–site Spin–Orbit* calculation, respectively. After that it will be called `init_energies` where the corresponding OSSO arrays or related variables will be initialised. Next step is to call the subroutine `initatom` for each specie in the claculation where the pseudopotential from the `#.psf` file will be read by calling `read_basis_specs`. Within `ATOM_MAIN` subroutine PPs will be splitted into two radial functions, $V_{l\pm1/2}(r)$ per each l quantum number [See theory for further details – item 5 of *General remarks* above]. The corresponding internal names of $\boxed{V_{l\pm1/2}(r)}$ will be $\boxed{vps\_u/\_d}$, respectivelly. After that, Kleinman–Bylander formalism will be used to build the projectors for the OSSO by means of the subroutines `KBproj_offsiteSO`, `KBgen_offsiteSO`. These radial functions will be stored in `comKB_offsiteSO` in order to use them afterwards. After leaving `ATOM_MAIN` is called `atm_transfer`, `deallocate_old_arrays`, `dump_basis_ascii`, `dump_basis_xml` and `dump_basis_nefcdf`. The first four subroutines support OSSO,

however, `netcdf` feature is not available yet fo OSSO. Furthermore, some `dump` subroutines are present in the `radial` module such as `radial_dump_xml_offsiteSO`, `radial_dump_ascii_offsiteSO`, `rad_get_KB`, that have been properly modified to be used if needed. After have been called to `initatom` for each atomic specie, KB projectors are assigned to their corresponding value of the internal `gindex` formalism by calling `register_rfs` and `register_in_rf_pool_KB`. Within the last subroutine, "`_KB`" suffix has been used instead of `_offsiteSO`. This is done within `siesta_init`.
Now that the stuff needed to initialise the calculation when the OSSO is involved has done, it starts the optimization loop that will call `siesta_forces`. There, it is called first `setup_H0` that obtains the hamiltonian terms that will not be updated at each step of the self–consistent (SO) loop and it is calculated only at the begining of the SC loop. In `setup_H0` among others is obtained the OSSO term of the hamiltonian $\mathrm{V}_{so,\mu\nu}$ calling `nlefsm_offsiteSO` that will use the KB projectors previously obtained in `ATOM_MAIN`. `nlefsm_offsiteSO` has similar structure as the former `nlefsm` but adapted to the OSSO [See `nlefsm` for details]. In briefly, what it is calculated therin are the integrals $\langle \phi_\mu \mid v_{kb,i} \rangle$ by means of `matel` calls and after that those scalar terms are combined to construct $\boxed{\mathrm{V}_{SO,\mu\nu}}$ [See Ref. of item 5 of *General remarks* above]. ϕ_μ are the basis orbitals and $v_{kb,i}$ the Kleinman–Bylander projectors. In the process of calculating the SO hamiltonian elements, $V_{SO,\mu\nu}$, it is called `calc_Vj_offsiteSO` where is calculated the V_{SO} and V_{ion} and the corresponding forces using the Clebsch–Gordan coefficients needed to change from the basis $\boxed{|l,m,\sigma\rangle \text{ to } |J,m_J\rangle}$. Leaving `setup_H0`, `siesta_forces` will get in into the main SC loop and it will be called `setup_hamiltonian`, `compute_dm` and the tests subroutines to check the SC. The first one calculates the whole Hamiltonian on each SC step and it will be diagonalize it in `compute_dm` by means of calls to `diagon`, etc.
One important part in the OSSO formalism is the calculation of the XC term that it is accesible when is called `dhscf` subroutine. Now there will be four complex terms in the XC hamiltonian matrix and they have to be calculated proeperly. This is done in `SiestaXC/ggaxc.f` subroutine where non–collinear formulation is taken into account. In general, in `setup_hamiltonian` the OSSO terms will be added in the following way:

$$
\begin{aligned}
H(:,1) &= H(:,1) + real(H0\_offsiteSO(:,1)) \\
H(:,5) &= \qquad\quad imag(H0\_offsiteSO(:,1))
\end{aligned}
$$

$$
\begin{aligned}
H(:,2) &= H(:,2) + real(H0\_offsiteSO(:,2)) \\
H(:,6) &= \qquad\quad imag(H0\_offsiteSO(:,2))
\end{aligned}
$$

$$
\begin{aligned}
H(:,3) &= H(:,3) + real(H0\_offsiteSO(:,3)) \\
H(:,4) &= H(:,4) + imag(H0\_offsiteSO(:,3))
\end{aligned}
$$

$$
\begin{aligned}
H(:,7) &= H(:,7) + real(H0\_offsiteSO(:,4)) \\
H(:,8) &= H(:,8) + imag(H0\_offsiteSO(:,4))
\end{aligned}
$$

Where the numbers from 1 to 8 are the real and imaginary components of the total hamiltonian as follows:

$$
H = \begin{pmatrix} H(:,1) + iH(:,5) & H(:,3) + iH(:,4) \\ H(:,7) + iH(:,8) & H(:,2) + iH(:,6) \end{pmatrix} \Longleftrightarrow \begin{pmatrix} H^{\uparrow\uparrow} & H^{\uparrow\downarrow} \\ H^{\downarrow\uparrow} & H^{\downarrow\downarrow} \end{pmatrix}
$$

After that is called `compute_dm` and for the OSSO the subroutines called on whether one has only to calculate the gamma point or bulk the subroutines called will be `diag3g` or `diag3k` respectively. Inside them an `if`(...) `then` ... `else` ... `endif` structure gides by the correct Spin–orbit formalism to use. The density matrix is obtained following the same sign criteria than before for the total hamiltonian. Note the change in the sign of $H(:,4)$ or $DM(:,4)$ components. This is due to the fact that here all tha terms are set up positive and the proper theory acts to have the correct sign when the total energy is obtained by means of

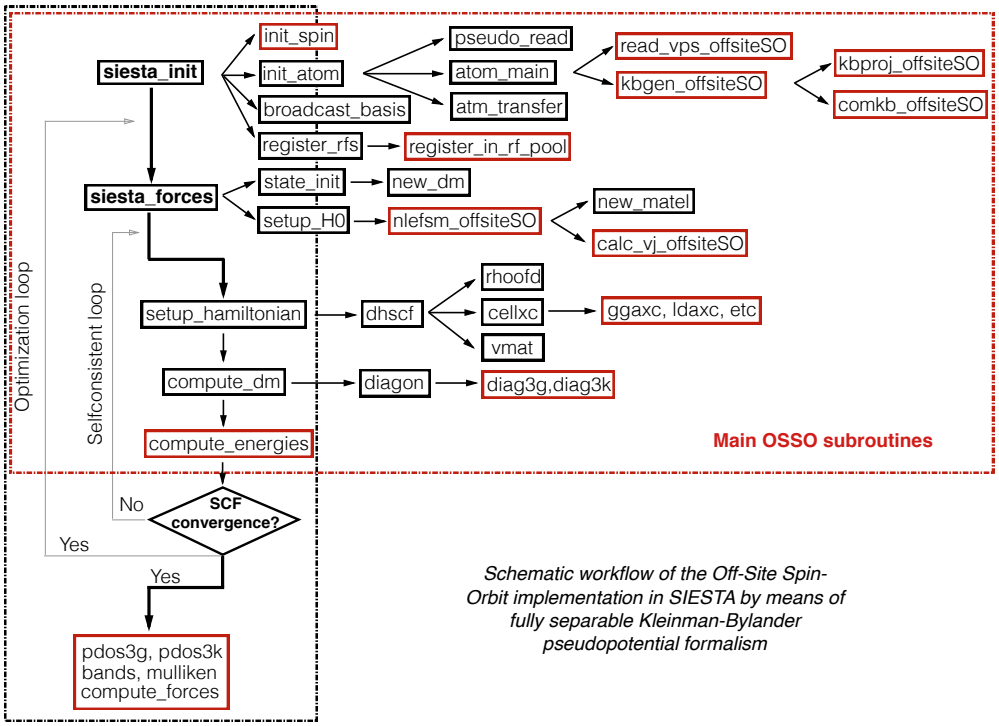

Figura 1: Schematic workflow of the process when the OSSO is included in the SIESTA calculation where only the most relevant subroutines have been shown. The main path of any selfconsistent calculation (inside the dashed black line square on the left) depicts the principal subroutine calls, i. e., initialization plus SC and final analysis. Some of the subroutines/modules outside the box were implemented previously and hence take part of the former SIESTA version, however in order to clarify the OSSO flow they are close to the ones that have been modified and/or created. Red boxes depict some of the new subroutines and also some of the old ones involved directly in the OSSO.

multiplication, for example. What it has to happen is that H and DM has to be hermitian, so that $\boxed{H^{\sigma\sigma'}_{\mu\nu} = (H^{\sigma'\sigma}_{\nu\mu})^*}$ So interms of the H componets, for instance, $H(:,7) = H(:,3)$ and $H(:,8) = -H(:,4)$ and within the diagonal blocks: $\boxed{H_{\mu\nu}(1) = H_{\nu\mu}(1)}$ and $\boxed{H_{\mu\nu}(5) = -H_{\nu\mu}(5)}$. When the SC is achieved `compute_energies` is called to obtain band structure energy as well as Harris term. In the final step `final_H_f_stress` subroutine is called and there `nlefsm_offsiteSO` is called again to update the OSSO terms with the SC density matrix.