

SIESTA School 2025

Spin, spin-orbit coupling and magnetism (I)

Ramón Cuadrado

Material Science Institute of Madrid (ICMM-CSIC)

Multiscale Materials Modelling Group

A: Non-Collinear magnetism

- How are treated the magnetic systems in DFT?
- **Collinear** spins
- **Non-collinear** spins
- Brief explanation of the theory behind non-collinearity
- Tips about **collinear** and **non-collinear** SIESTA runs

B: Spin-orbit coupling (SOC) in SIESTA

- Motivation
- Relationship between **magnetic anisotropy energy** (MAE) and **SOC**
- **Pseudopotentials** and **SOC**
- Scalar (SR) and fully relativistic (FR) calculations
- How **SOC** is included in SIESTA
- Some examples
- Running SIESTA including **SOC**

C: Electronic structure analysis and convergence

- Charge and magnetic moments (MMs)
- Projected density of states (PDOS)
- Bands structure
- Convergence tips

A: Non-Collinear magnetism

- How are treated the magnetic systems in DFT?
- ***Collinear*** spins
- ***Non-collinear*** spins
- Brief explanation of the theory behind non-collinearity
- Tips about ***collinear*** and ***non-collinear*** SIESTA runs

• **How are treated the magnetic systems in DFT?**

- In general, DFT formulation is spin independent \longrightarrow The states $\Psi_l(\mathbf{r})$ are degenerated
- The degeneracy of the states has to be broken \longrightarrow $\Psi_l(\mathbf{r}) \begin{cases} \psi_l^\uparrow(\mathbf{r}) \\ \psi_l^\downarrow(\mathbf{r}) \end{cases} \longrightarrow \Psi_l(\mathbf{r}) = \begin{pmatrix} \psi_l^\uparrow(\mathbf{r}) \\ \psi_l^\downarrow(\mathbf{r}) \end{pmatrix}$
Spinors
- The quantization axis is usually the Z axis
- We will have one KS eq. per spin $\sigma = \uparrow, \downarrow$ (independent spins)

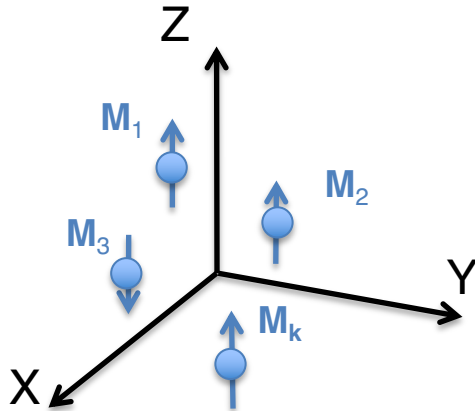
$$\underbrace{\left[\hat{T}_s + \hat{V}_{eI} + \hat{V}_H + \hat{V}_{XC}^\sigma \right]}_{\hat{H}_{TOT}} \psi_i^{KS,\sigma} = \epsilon_i^{KS,\sigma} \psi_i^{KS}$$

$$\hat{H}_{\mu\nu}^{\sigma\sigma} = \langle \phi_\mu, \sigma | \hat{H}_{TOT} | \phi_\nu, \sigma \rangle$$

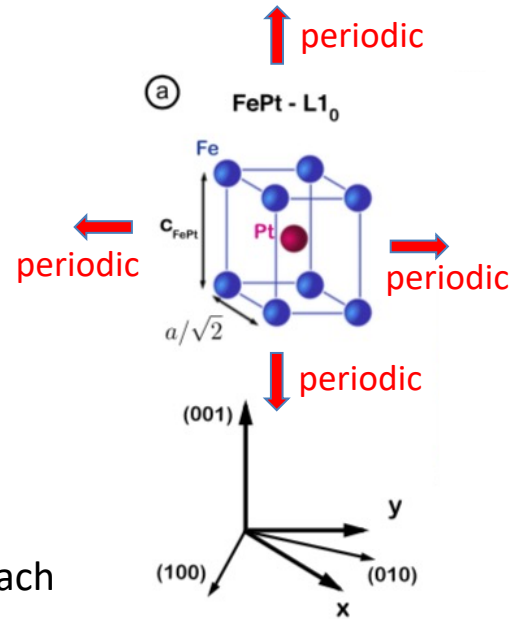
- How is obtained the spin polarization?

SIESTA calculates the spin polarization as the difference between the up and down charge \longrightarrow $MM_Z = Q_Z^\uparrow - Q_Z^\downarrow [\mu_B]$

- Collinear spins:

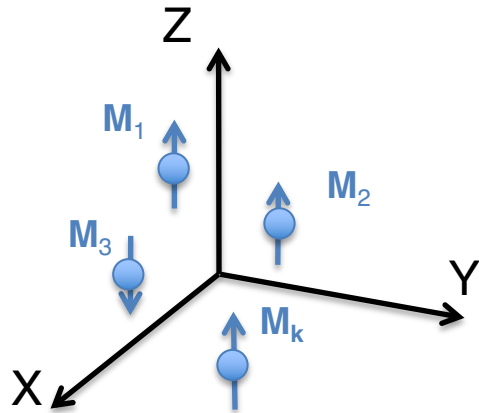


Atomic spins are parallel or antiparallel each other and parallel to the quantization axis



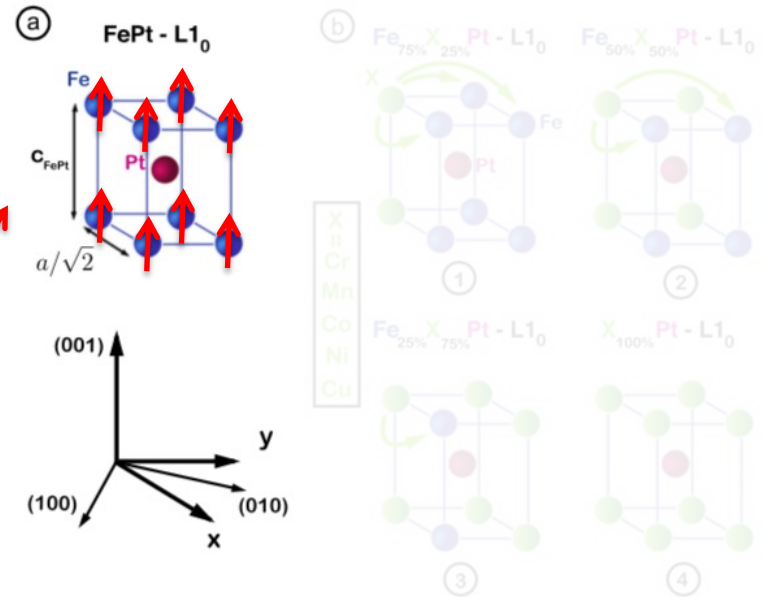
- Ferromagnetic (FM)
- Antiferromagnetic (AFM)
- Ferrimagnetic (FiM)

- Collinear spins:

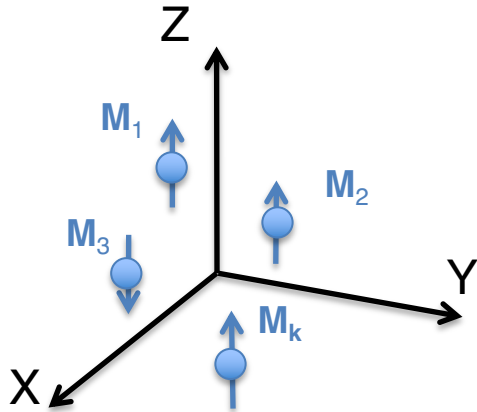


Atomic spins are parallel or antiparallel each other and parallel to the quantization axis

- Ferromagnetic (FM)
- Antiferromagnetic (AFM)
- Ferrimagnetic (FiM)

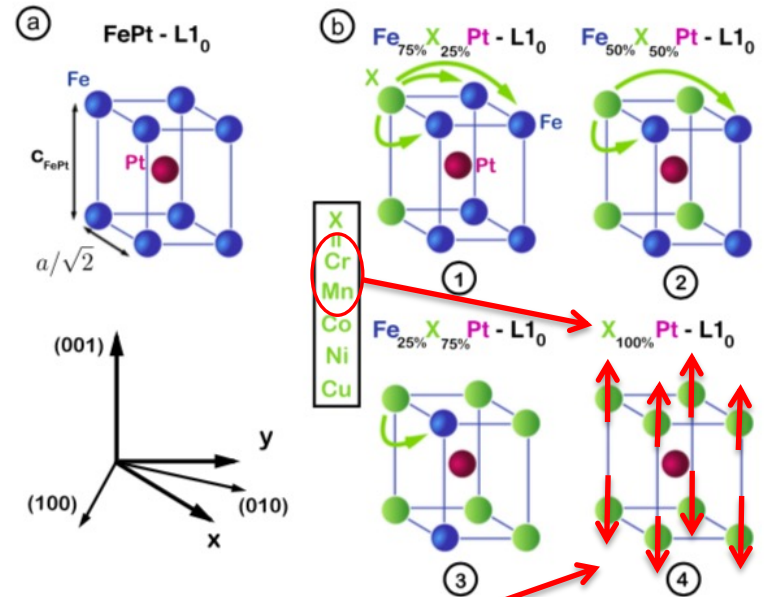


- Collinear spins:

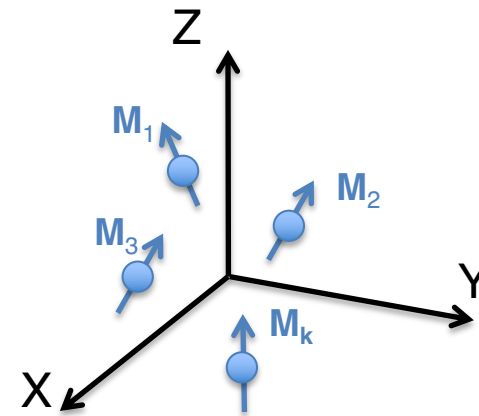
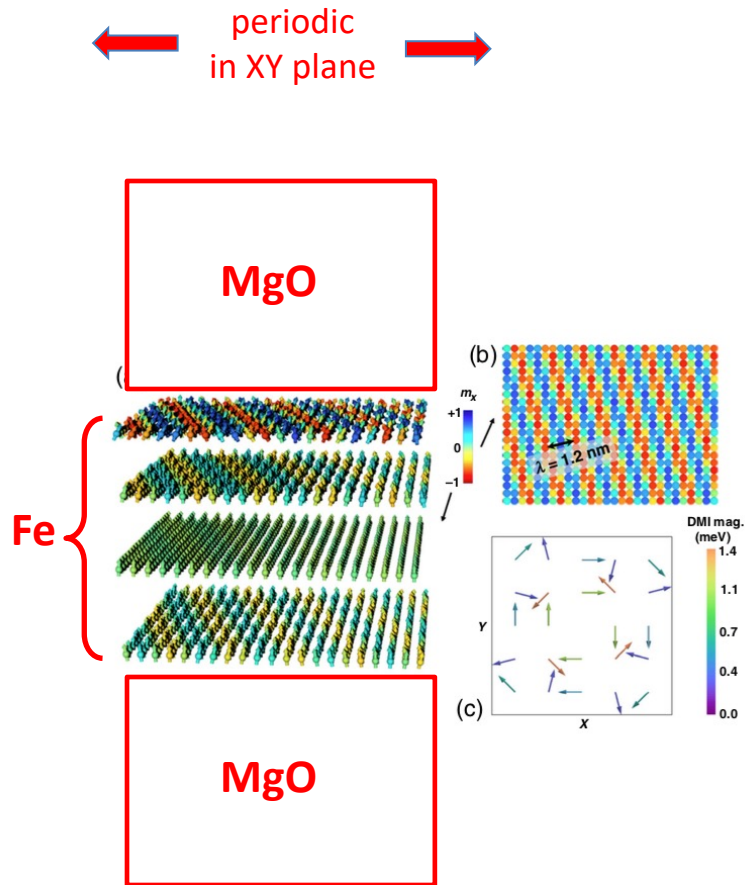


Atomic spins are parallel or antiparallel each other and parallel to the quantization axis

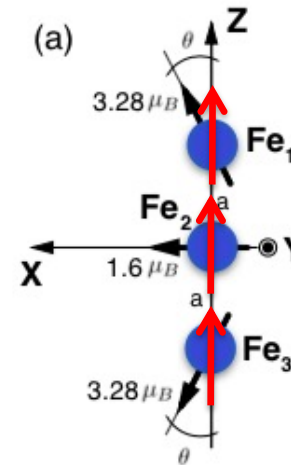
- Ferromagnetic (FM)
- Antiferromagnetic (AFM)
- Ferrimagnetic (FiM)



- Non-collinear spins:

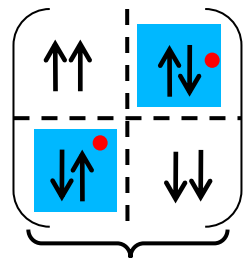


Atomic spins are randomly oriented



- **Non-collinear spins:**

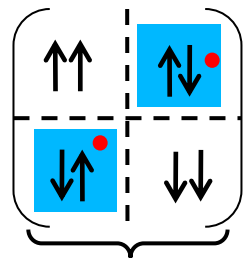
- In the *non-collinear* case the spins are not independent

$$\underbrace{\left[\hat{T}_s + \hat{V}_{eI}^{\sigma\sigma'} + \hat{V}_H + \hat{V}_{XC}^{\sigma\sigma'} \right]}_{\hat{H}_{TOT}} \psi_i^{KS} = \epsilon_i^{KS} \psi_i^{KS} \quad [\sigma, \sigma' = \uparrow, \downarrow]$$


- $\hat{H}_{\mu\nu}^{\sigma\sigma'} = \langle \phi_\mu, \sigma | \hat{H}_{TOT} | \phi_\nu, \sigma' \rangle$

- **Non-collinear spins:**

- In the *non-collinear* case the spins are not independent

$$\underbrace{\left[\hat{T}_s + \hat{V}_{eI}^{\sigma\sigma'} + \hat{V}_H + \hat{V}_{XC}^{\sigma\sigma'} \right]}_{\hat{H}_{TOT}} \psi_i^{KS} = \epsilon_i^{KS} \psi_i^{KS} \quad [\sigma, \sigma' = \uparrow, \downarrow]$$


- $\hat{H}_{\mu\nu}^{\sigma\sigma'} = \langle \phi_{\mu}, \sigma | \hat{H}_{TOT} | \phi_{\nu}, \sigma' \rangle$

- In the *collinear* case:

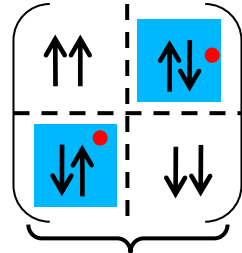
$$M_k^z = Tr \left[\rho(\mathbf{r}) \cdot \sigma_z \right] = \sum_{\mu\nu} (\rho_{\mu\nu}^{\uparrow\uparrow} - \rho_{\mu\nu}^{\downarrow\downarrow}) (W_k)_{\mu\nu}$$

Orbital overlaps

$$\Psi_l(\mathbf{r}) = \begin{pmatrix} \psi_l^{\uparrow}(\mathbf{r}) \\ \psi_l^{\downarrow}(\mathbf{r}) \end{pmatrix} \xrightarrow{\omega=\uparrow, \downarrow} \rho^{\omega\omega'}(\mathbf{r}) = \sum_i n_i \psi_i^{\omega'*}(\mathbf{r}) \psi_i^{\omega}(\mathbf{r}) = \sum_{\mu\nu} \sum_i n_i [c_{i,\nu}^{\omega'}]^* c_{i,\mu}^{\omega} \phi_{\nu}^*(\mathbf{r}) \phi_{\mu}(\mathbf{r}) = \sum_{\mu\nu} \rho_{\mu\nu}^{\omega\omega'} \phi_{\nu}^*(\mathbf{r}) \phi_{\mu}(\mathbf{r})$$

- **Non-collinear spins:**

- In the **non-collinear** case the spins are not independent

$$\underbrace{\left[\hat{T}_s + \hat{V}_{eI}^{\sigma\sigma'} + \hat{V}_H + \hat{V}_{XC}^{\sigma\sigma'} \right]}_{\hat{H}_{\text{TOT}}} \psi_i^{KS} = \epsilon_i^{KS} \psi_i^{KS} \quad [\sigma, \sigma' = \uparrow, \downarrow]$$


- $\hat{H}_{\mu\nu}^{\sigma\sigma'} = \langle \phi_{\mu}, \sigma | \hat{H}_{\text{TOT}} | \phi_{\nu}, \sigma' \rangle$

- In the **collinear** case:

$$M_k^z = \text{Tr} [\rho(\mathbf{r}) \cdot \sigma_z] = \sum_{\mu\nu} (\rho_{\mu\nu}^{\uparrow\uparrow} - \rho_{\mu\nu}^{\downarrow\downarrow}) (W_k)_{\mu\nu}$$

← Orbital overlaps

- In the **non-collinear** case:

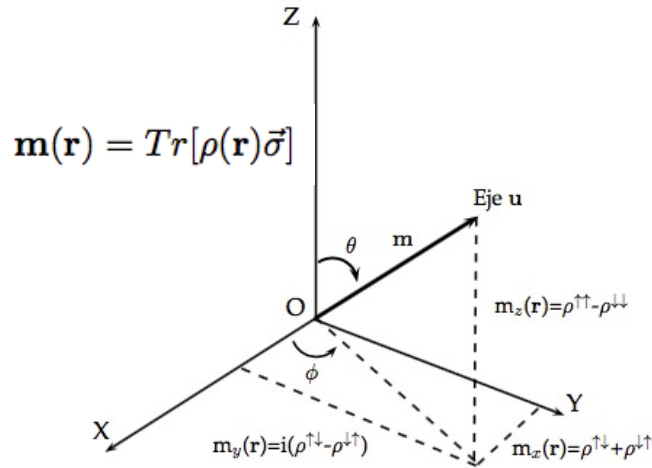
$$M_k^x = \text{Tr} [\rho(\mathbf{r}) \cdot \sigma_x] = \sum_{\mu\nu} (\rho_{\mu\nu}^{\uparrow\downarrow} + \rho_{\mu\nu}^{\downarrow\uparrow}) (W_k)_{\mu\nu}$$

$$M_k^y = \text{Tr} [\rho(\mathbf{r}) \cdot \sigma_y] = i \sum_{\mu\nu} (\rho_{\mu\nu}^{\uparrow\downarrow} - \rho_{\mu\nu}^{\downarrow\uparrow}) (W_k)_{\mu\nu}$$

$$M_k^z = \text{Tr} [\rho(\mathbf{r}) \cdot \sigma_z] = \sum_{\mu\nu} (\rho_{\mu\nu}^{\uparrow\uparrow} - \rho_{\mu\nu}^{\downarrow\downarrow}) (W_k)_{\mu\nu}$$

$$\Psi_l(\mathbf{r}) = \begin{pmatrix} \psi_l^{\uparrow}(\mathbf{r}) \\ \psi_l^{\downarrow}(\mathbf{r}) \end{pmatrix} \xrightarrow{\omega=\uparrow, \downarrow} \rho^{\omega\omega'}(\mathbf{r}) = \sum_i n_i \psi_i^{\omega'*}(\mathbf{r}) \psi_i^{\omega}(\mathbf{r}) = \sum_{\mu\nu} \sum_i n_i [c_{i,\nu}^{\omega'}]^* c_{i,\mu}^{\omega} \phi_{\nu}^*(\mathbf{r}) \phi_{\mu}(\mathbf{r}) = \sum_{\mu\nu} \rho_{\mu\nu}^{\omega\omega'} \phi_{\nu}^*(\mathbf{r}) \phi_{\mu}(\mathbf{r})$$

- Non-collinear spins:



- $|\mathbf{M}_k|^2 = (\rho^{\uparrow\uparrow} - \rho^{\downarrow\downarrow})^2 + 4[\text{Re}\{\rho^{\uparrow\downarrow}\}^2 + \text{Im}\{\rho^{\uparrow\downarrow}\}^2]$
- $\sin\theta_k = \frac{2|\rho^{\uparrow\downarrow}|}{|\mathbf{M}_k|}$; $\cos\theta_k = \frac{\rho^{\uparrow\uparrow} - \rho^{\downarrow\downarrow}}{|\mathbf{M}_k|}$
- $\cos\varphi_k = \frac{\text{Re}\{\rho^{\uparrow\downarrow}\}}{[\text{Re}\{\rho^{\uparrow\downarrow}\}^2 + \text{Im}\{\rho^{\uparrow\downarrow}\}^2]^{1/2}}$;
- $\sin\varphi_k = -\frac{\text{Im}\{\rho^{\uparrow\downarrow}\}}{[\text{Re}\{\rho^{\uparrow\downarrow}\}^2 + \text{Im}\{\rho^{\uparrow\downarrow}\}^2]^{1/2}}$
- $\tan\varphi_k = -\frac{\text{Im}\{\rho^{\uparrow\downarrow}\}}{\text{Re}\{\rho^{\uparrow\downarrow}\}}$; $\tan\theta_k = \frac{2[\text{Re}\{\rho^{\uparrow\downarrow}\}^2 + \text{Im}\{\rho^{\uparrow\downarrow}\}^2]^{1/2}}{\rho^{\uparrow\uparrow} - \rho^{\downarrow\downarrow}}$

- Some tips about *collinear* and *non-collinear* SIESTA runs

6.7 Spin polarization

Spin non-polarized

(string)

deprecates: SpinPolarized, NonCollinearSpin, SpinOrbit

Choose the spin-components in the simulation.

NOTE: This flag has precedence over **SpinOrbit**, **NonCollinearSpin** and **SpinPolarized** while these deprecated flags may still be used.

non-polarized Perform a calculation with spin-degeneracy (only one component).

polarized Perform a calculation with colinear spin (two spin components).

non-collinear Perform a calculation with non-collinear spin (4 spin components), up-down and angles.

Refs: T. Oda et al, PRL, **80**, 3622 (1998); V. M. García-Suárez et al, Eur. Phys. Jour. B **40**, 371 (2004); V. M. García-Suárez et al, Journal of Phys: Cond. Matt **16**, 5453 (2004).

spin-orbit Performs calculations including the spin-orbit coupling. By default the full SO option is set. To perform an on-site SO calculation (see 6.8.1) this option has to be **spin-orbit+onsite**. This requires the pseudopotentials to be relativistic.

See Sect. 6.8 for further specific spin-orbit options.

SIESTA can read a .DM with different spin structure by adapting the information to the currently selected spin multiplicity, averaging or splitting the spin components equally, as needed. This may be used to greatly increase convergence.

Certain options may not be used together with specific parallelization routines.

- Some tips about *collinear* and *non-collinear* SIESTA runs

```
%block DM.InitSpin (None) (block)
```

Define the initial spin density for a spin polarized calculation atom by atom. In the block there is one line per atom to be spin-polarized, containing the atom index (integer, ordinal in the block **AtomicCoordinatesAndAtomicSpecies**) and the desired initial spin-polarization (real, positive for spin up, negative for spin down). A value larger than possible will be reduced to the maximum possible polarization, keeping its sign. Maximum polarization can also be given by introducing the symbol + or - instead of the polarization value. There is no need to include a line for every atom, only for those to be polarized. The atoms not contemplated in the block will be given non-polarized initialization.

For non-collinear spin, the spin direction may be specified for each atom by the polar angle θ and the azimuthal angle ϕ (using the physics ISO convention), given as the last two arguments in degrees. If not specified, $\theta = 0$ is assumed (z -polarized). **Spin** must be set to use non-collinear or spin-orbit for the directions to have effect.

Example:

```
%block DM.InitSpin
  5 -1.  90.  0.  # Atom index, spin, theta, phi (deg)
  3 +   45. -90.
  7 -
%endblock DM.InitSpin
```

In the above example, atom 5 is polarized in the x -direction.

- Some tips about *collinear* and *non-collinear* SIESTA runs

```
%block DM.InitSpin (None) (block)
```

Define the initial spin density for a spin polarized calculation atom by atom. In the block there is one line per atom to be spin-polarized, containing the atom index (integer, ordinal in the block **AtomicCoordinatesAndAtomicSpecies**) and the desired initial spin-polarization (real, positive for spin up, negative for spin down). A value larger than possible will be reduced to the maximum possible polarization, keeping its sign. Maximum polarization can also be given by introducing the symbol + or - instead of the polarization value. There is no need to include a line for every atom, only for those to be polarized. The atoms not contemplated in the block will be given non-polarized initialization.

For non-collinear spin, the spin direction may be specified for each atom by the polar angle θ and the azimuthal angle ϕ (using the physics ISO convention), given as the last two arguments in degrees. If not specified, $\theta = 0$ is assumed (z -polarized). **Spin** must be set to use non-collinear or spin-orbit for the directions to have effect.

Example:

```
%block DM.InitSpin
  5 -1. 90. 0. # Atom index, spin, theta, phi (deg)
  3 + 45. -90.
  7 -
%endblock DM.InitSpin
```

In the above example, atom 5 is polarized in the x -direction.

In general

- Some tips about *collinear* and *non-collinear* SIESTA runs

`%block DM.InitSpin` (None) (block)

Define the initial spin density for a spin polarized calculation atom by atom. In the block there is one line per atom to be spin-polarized, containing the atom index (integer, ordinal in the block **AtomicCoordinatesAndAtomicSpecies**) and the desired initial spin-polarization (real, positive for spin up, negative for spin down). A value larger than possible will be reduced to the maximum possible polarization, keeping its sign. Maximum polarization can also be given by introducing the symbol + or - instead of the polarization value. There is no need to include a line for every atom, only for those to be polarized. The atoms not contemplated in the block will be given non-polarized initialization.

For non-collinear spin, the spin direction may be specified for each atom by the polar angle θ and the azimuthal angle ϕ (using the physics ISO convention), given as the last two arguments in degrees. If not specified, $\theta = 0$ is assumed (z -polarized). **Spin** must be set to use non-collinear or spin-orbit for the directions to have effect.

Example:

```
%block DM.InitSpin
 5 -1. 90. 0. # Atom index, spin, theta, phi (deg)
 3 + 45. -90.
 7 -
%endblock DM.InitSpin
```

In the above example, atom 5 is polarized in the x -direction.

collinear

```
%block DM.InitSpin
 5 -1.
 3 +
 7 -
%endblock DM.InitSpin
```

non-collinear

```
%block DM.InitSpin
 5 -1. 90. 0. # Atom index, spin, theta, phi (deg)
 3 + 45. -90.
 7 -
%endblock DM.InitSpin
```

- Typical SIESTA *non-collinear* output for charge and MMs

mulliken: Atomic and Orbital Populations:

Species: Fe

Atom	Orb	Charge	Spin	Svec		
33	1 4s	-0.19681	0.05819	0.000	0.000	0.058
33	2 4s	0.57426	0.04693	0.000	0.000	-0.047
33	3 3dxy	1.19883	0.47229	0.001	-0.001	0.472
33	4 3dyz	1.16429	0.51568	0.001	-0.004	0.516
33	5 3dz2	1.08874	0.61043	-0.002	-0.002	0.610
33	6 3dxz	1.12054	0.57207	0.004	-0.001	0.572
33	7 3dx2-y2	1.05433	0.63646	0.000	0.000	0.636
33	8 3dxy	0.13853	0.11908	0.000	0.000	0.119
33	9 3dyz	0.13454	0.11858	0.000	-0.001	0.119
33	10 3dz2	0.14139	0.13090	0.000	0.000	0.131
33	11 3dxz	0.13788	0.12690	0.000	0.000	0.127
33	12 3dx2-y2	0.14683	0.13812	0.000	0.000	0.138
33	13 4Ppy	0.18863	0.00747	0.000	0.000	0.007
33	14 4Ppz	0.18812	0.01902	0.000	0.000	0.019
33	15 4Ppx	0.19019	0.01364	0.000	0.000	0.014
33	Total	7.27029	3.49189	0.006	-0.010	3.492

B: Spin-orbit coupling (SOC) in SIESTA

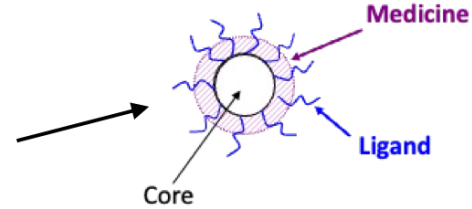
- Motivation
- Relationship between *magnetic anisotropy energy* (MAE) and **SOC**
- *Pseudopotentials* and **SOC**
- Scalar (SR) and fully relativistic (FR) calculations
- How **SOC** is included in SIESTA
- Some examples
- Running SIESTA including **SOC**

- **What is nanotechnology?**

The ability to manipulate materials at the nanometer scale and enhance their functional properties for various types of applications

- **Nanodevices:**

- Medical applications
- Catalytic reactions
- Magnetic recording, MRAM
- Magnetic sensors
- Optoelectronics
- ...



- **Focusing in Magnetic Devices ... specifically on Hard Disk Drives technology**

We want to manufacture the 'ultimate device' that will be simultaneously:

- Inexpensive
- Fast
- Low energy consumption
- High storage density
- Durable

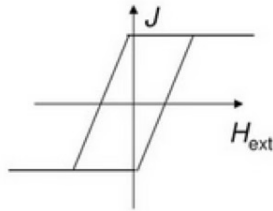
• **Read/write heads:**



State-of-the-art 1-2 Tbit/in² → Goal > 5 Tbit/in²

Writability Trilemma

The **magnetic recording trilemma** embodies the fundamental compromise between **writability**, **thermal stability**, and **readable signal-to-noise ratio**.



Writeability:

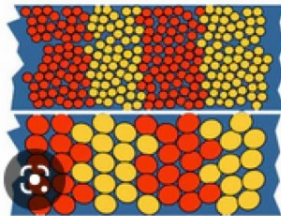
$$H_c = 2K / J_s$$

$$H_c < 1.0 - 1.5 \text{ T}$$

Signal to Noise:

$$SNR \sim \log_{10}(N)$$

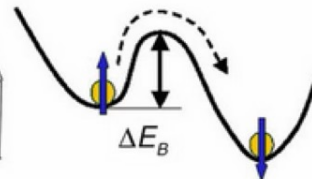
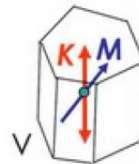
small grain V



Thermal stability:

$$\Delta E_B = KV$$

high $KV \sim 70 k_b T_{300}$

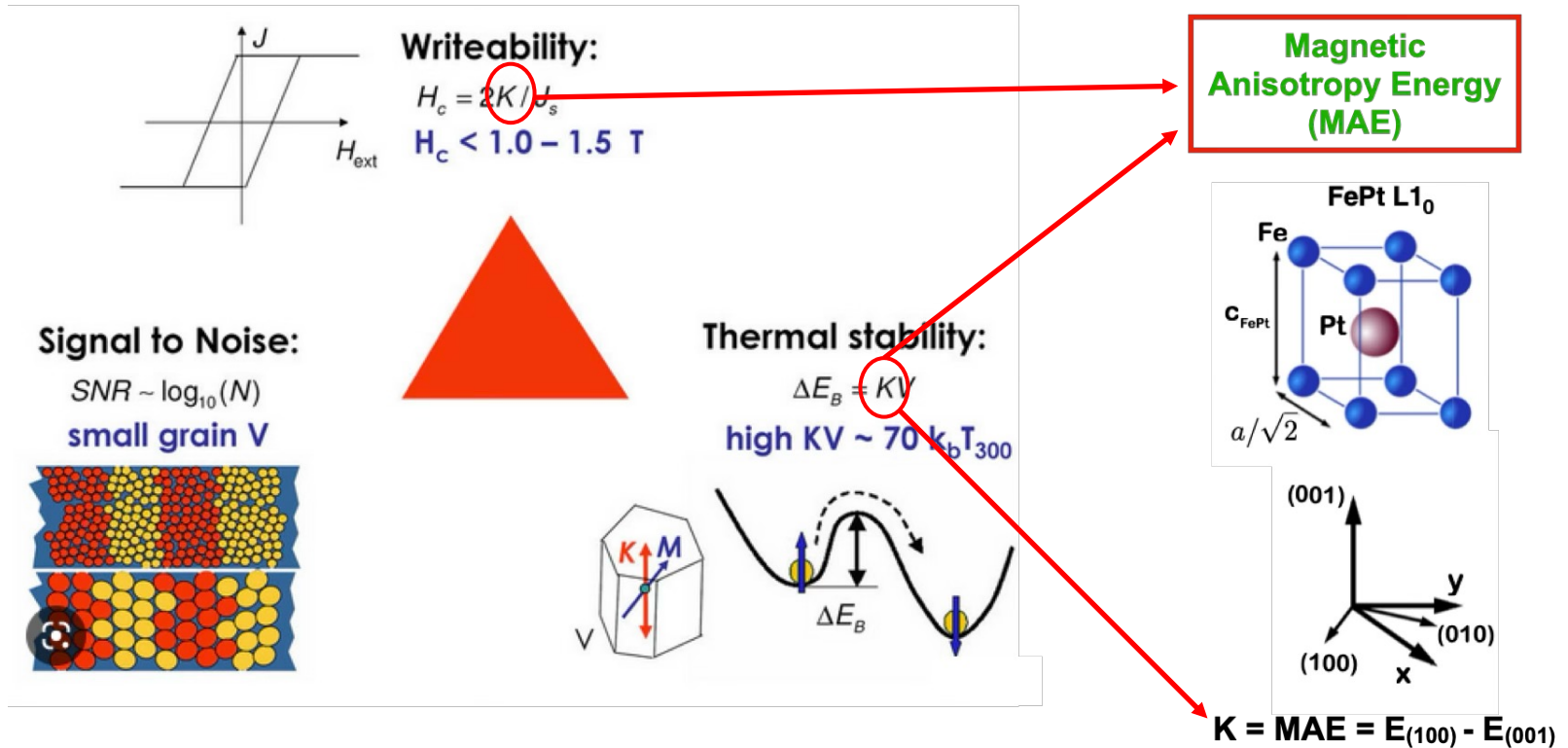


• Read/write heads:



State-of-the-art 1-2 Tbit/in² → Goal > 5 Tbit/in²

Writability Trilemma



- Which is the relationship between MAE and SOC?

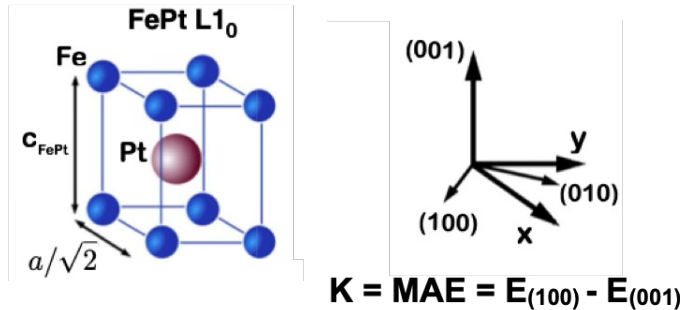
Magnetic anisotropy energy, K



Direct consequence of the SOC



It is a relativistic effect



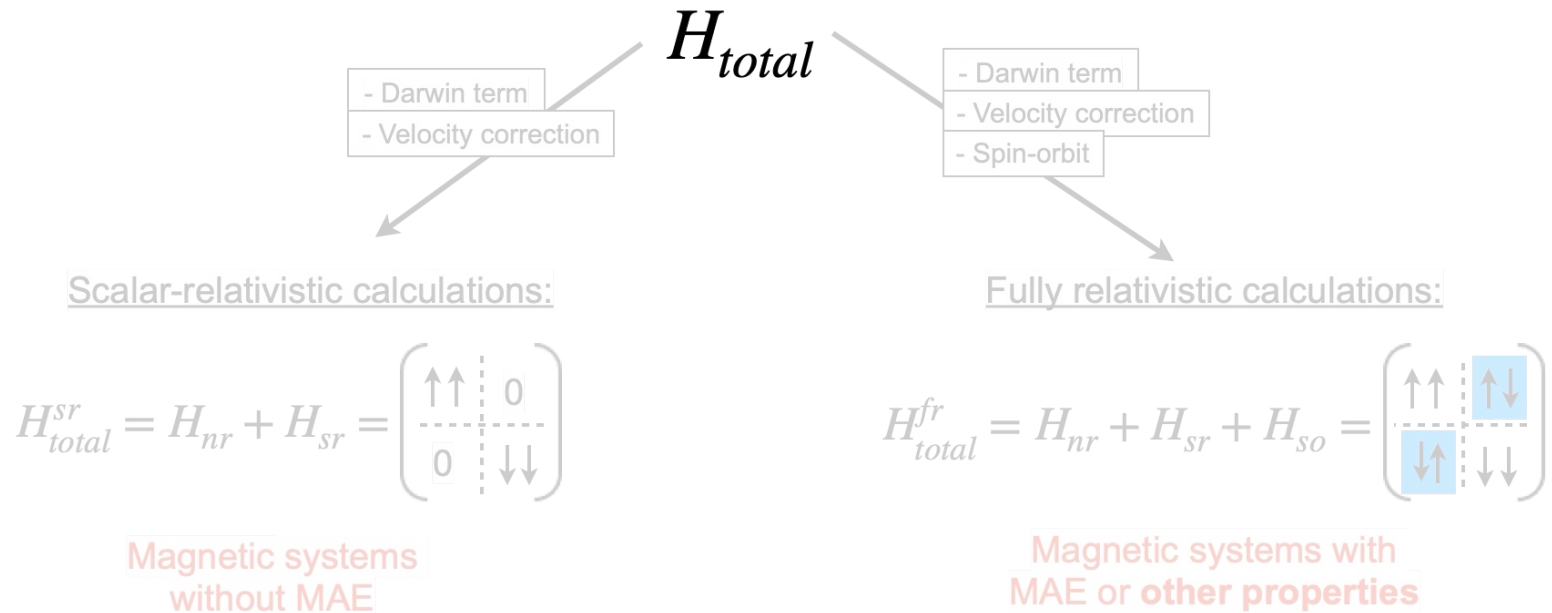
We have to include the Relativistic Effects in the DFT codes



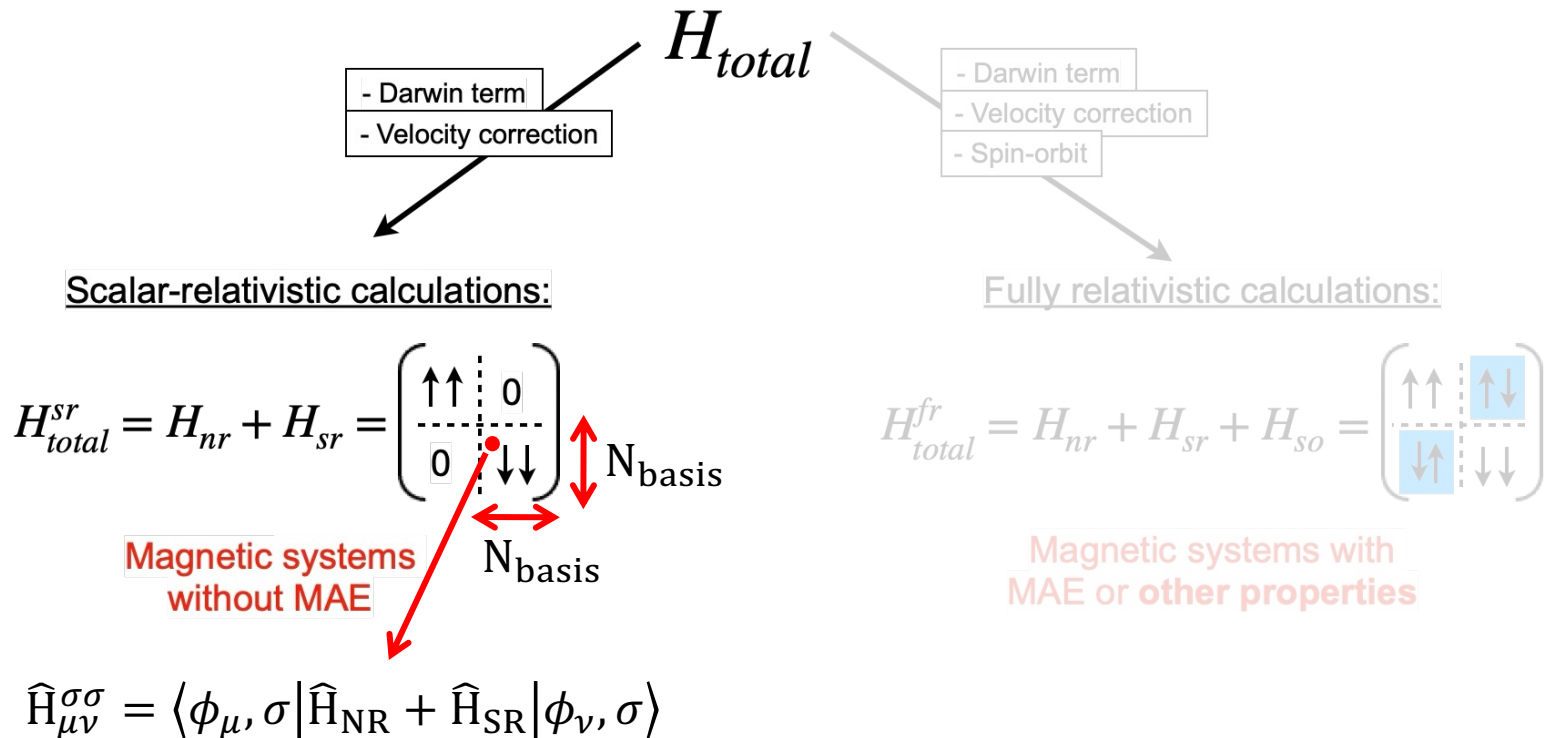
- Difficulties of SOC implementation:

- Complex mathematical formalism to be adapted to the DFT code
- Need high computational resources due to the increase of the sizes of the matrices
- Parallel programming
- Magnetic anisotropy energy is of order of meV, so it is needed high accuracy in the calculations

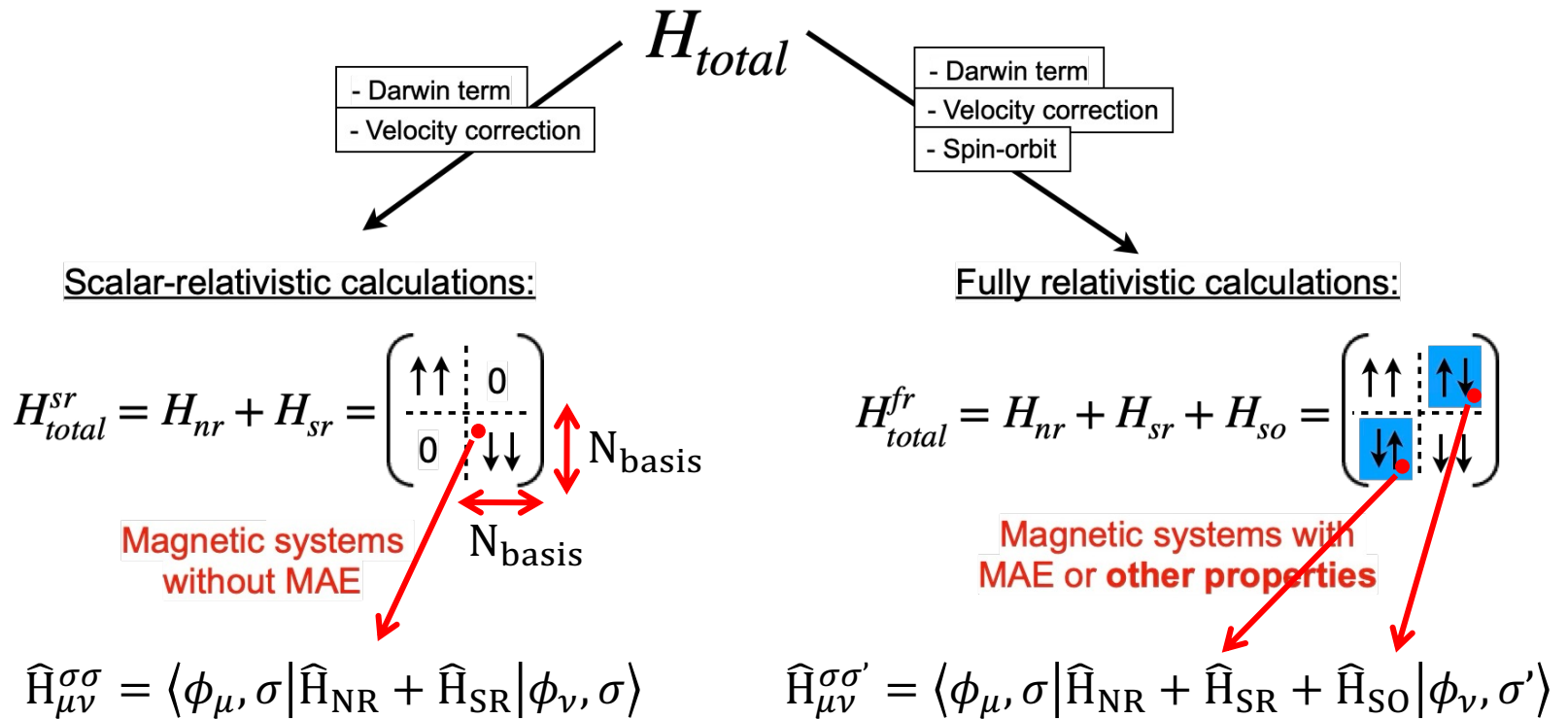
- **Two types of calculations:**



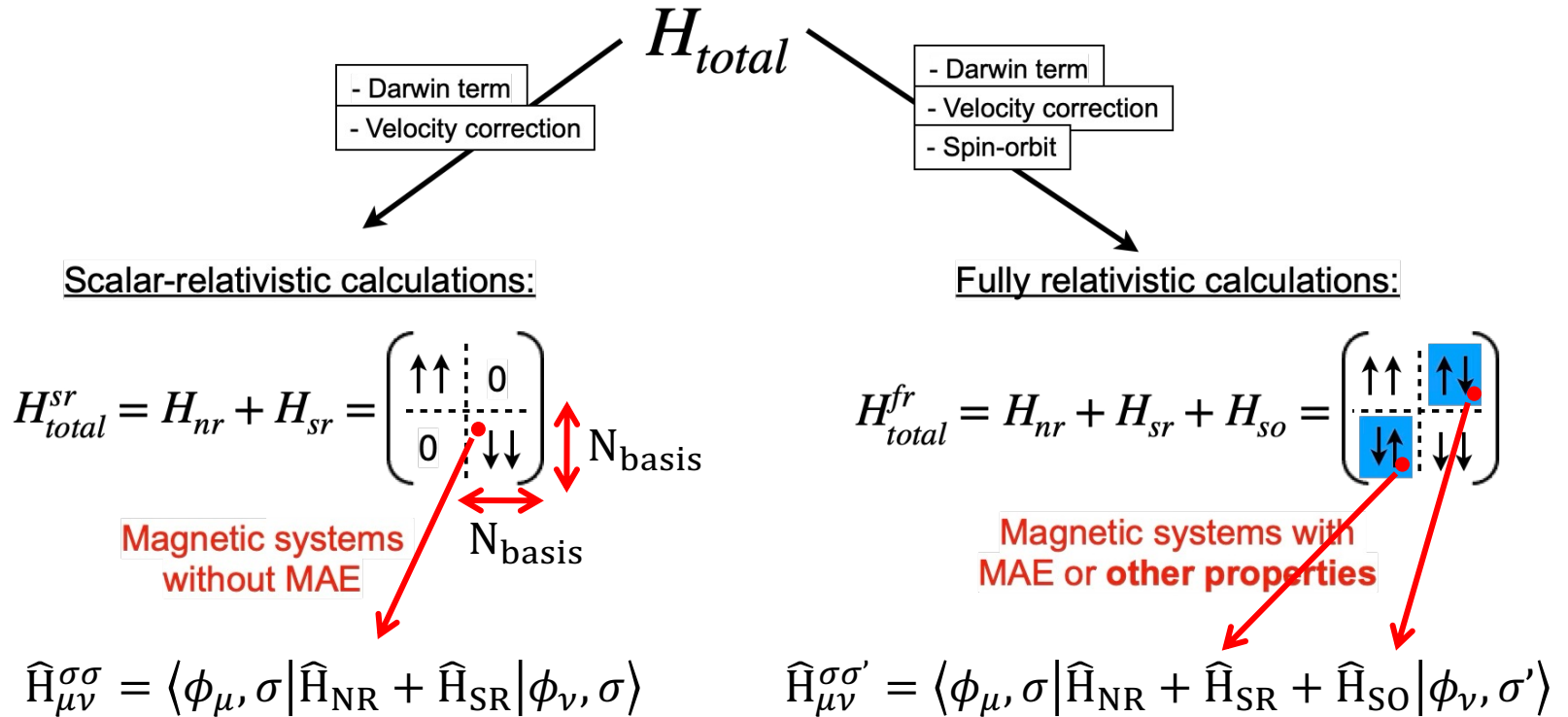
• **Two types of calculations:**



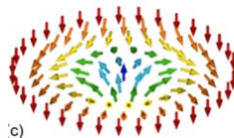
- Two types of calculations:



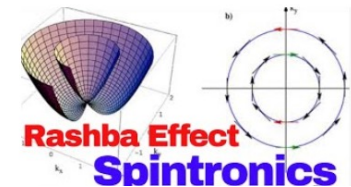
- Two types of calculations:



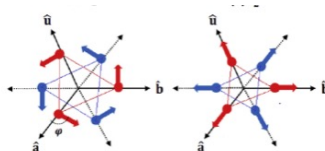
- Skymionic Systems



- Rashba Effect

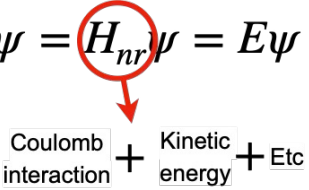


- Non-collinear magnetism



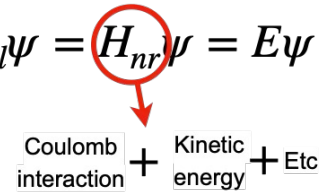
- **How will we include SOC in a Density Functional Theory code?**

→ SOC is a relativistic effect, however, DFT itself is a non-relativistic theory $\longleftrightarrow H_{total}\psi = H_{nr}\psi = E\psi$


Coulomb interaction + Kinetic energy + Etc

- **How will we include SOC in a Density Functional Theory code?**

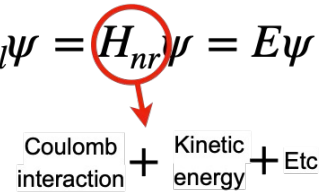
- SOC is a relativistic effect, however, DFT itself is a non-relativistic theory $\longleftrightarrow H_{total}\psi = H_{nr}\psi = E\psi$
- If we want to include Relativistic Effects $\longleftrightarrow H_{total} = H_{nr} + \text{Relativistic effects}$



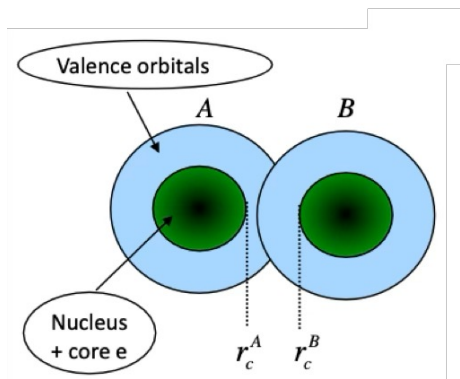
• **How will we include SOC in a Density Functional Theory code?**

→ SOC is a relativistic effect, however, DFT itself is a non-relativistic theory $\leftrightarrow H_{total}\psi = H_{nr}\psi = E\psi$

→ If we want to include Relativistic Effects $\leftrightarrow H_{total} = H_{nr} + \text{Relativistic effects}$



→ Using Atomic Pseudopotentials (PPs)



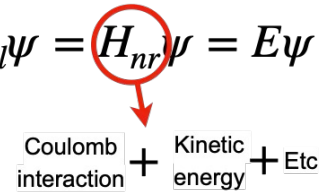
— The valence electrons are the main responsible of chemical bonding

— So one can build a radial function (PPs) that joins the effect of **nucleus + core-electrons**

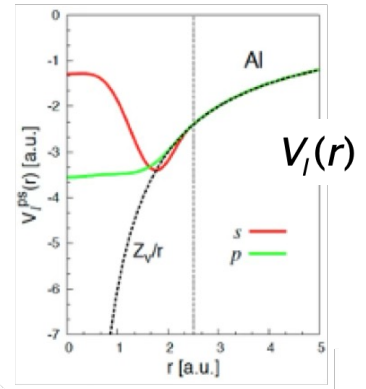
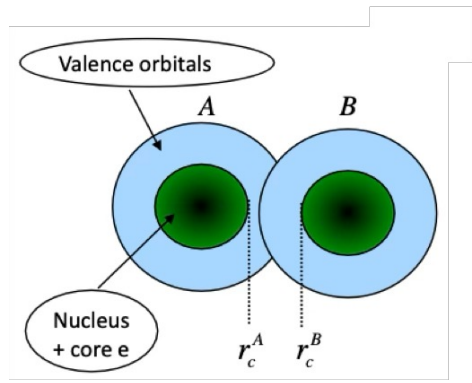
• **How will we include SOC in a Density Functional Theory code?**

→ SOC is a relativistic effect, however, DFT itself is a non-relativistic theory $\leftrightarrow H_{total}\psi = H_{nr}\psi = E\psi$

→ If we want to include Relativistic Effects $\leftrightarrow H_{total} = H_{nr} + \text{Relativistic effects}$



→ Using Atomic Pseudopotentials (PPs)



- The valence electrons are the main responsible of chemical bonding
- So one can built a radial function (PPs) that joins the effect of **nucleus + core-electrons**

Schrödinger (non relativistic)

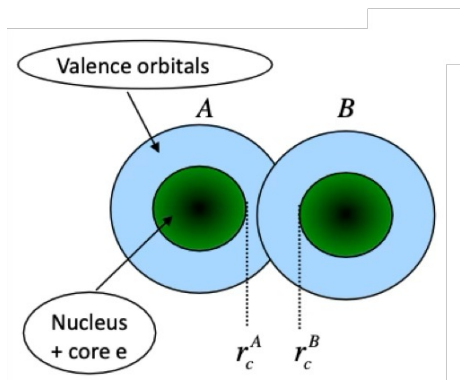
• How will we include SOC in a Density Functional Theory code?

→ SOC is a relativistic effect, however, DFT itself is a non-relativistic theory $\leftrightarrow H_{total}\psi = H_{nr}\psi = E\psi$

→ If we want to include Relativistic Effects $\leftrightarrow H_{total} = H_{nr} + \text{Relativistic effects}$

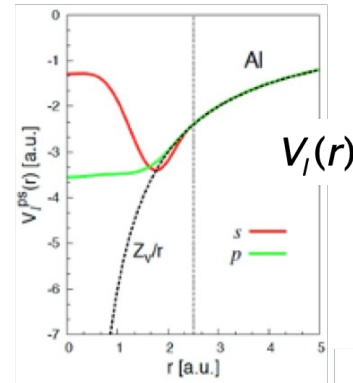
Coulomb interaction + Kinetic energy + Etc

→ Using Atomic Pseudopotentials (PPs)



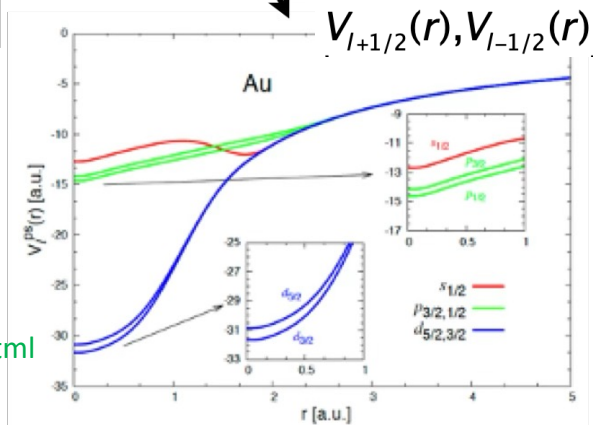
— The valence electrons are the main responsible of chemical bonding

— So one can build a radial function (PPs) that joins the effect of **nucleus + core-electrons**



Schrödinger (non relativistic)

Dirac (relativistic)

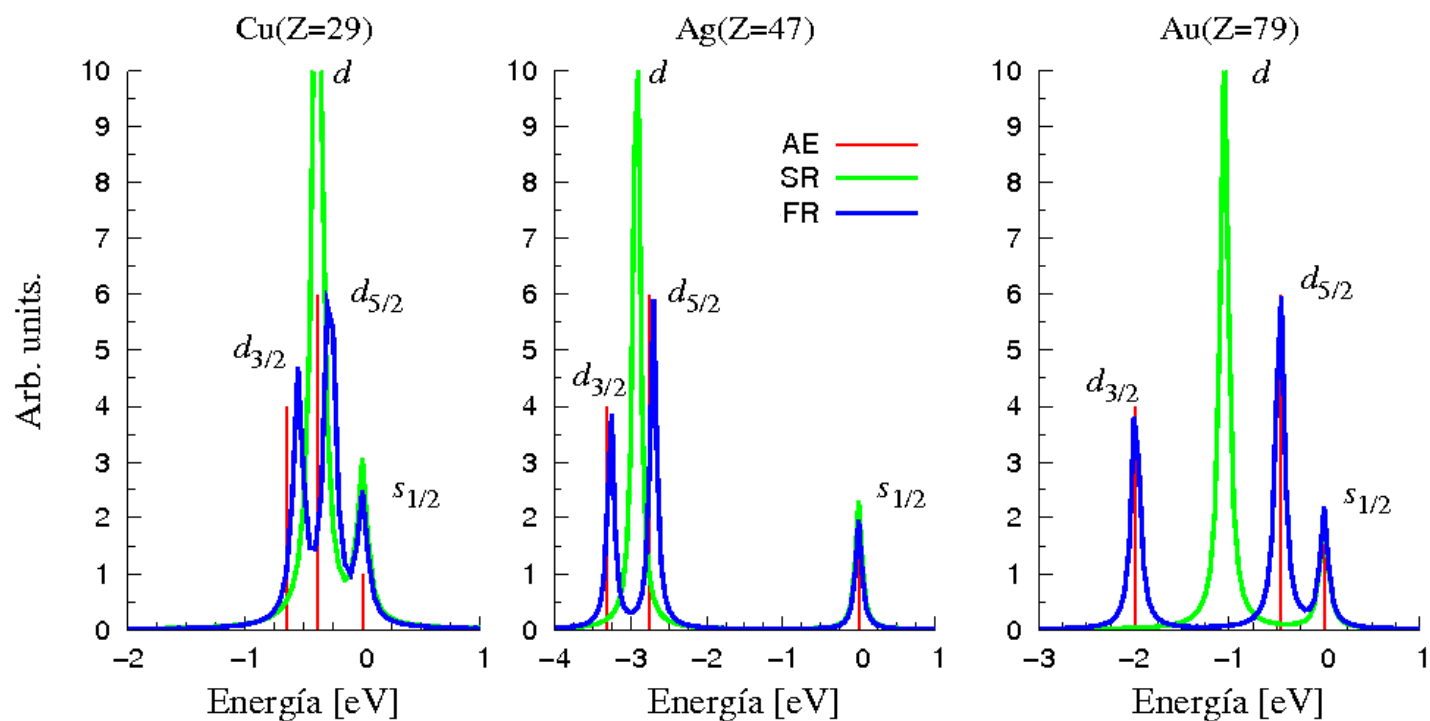


★ **ATOM PROGRAM:** <https://departments.icmab.es/leem/siesta/Pseudopotentials/index.html>

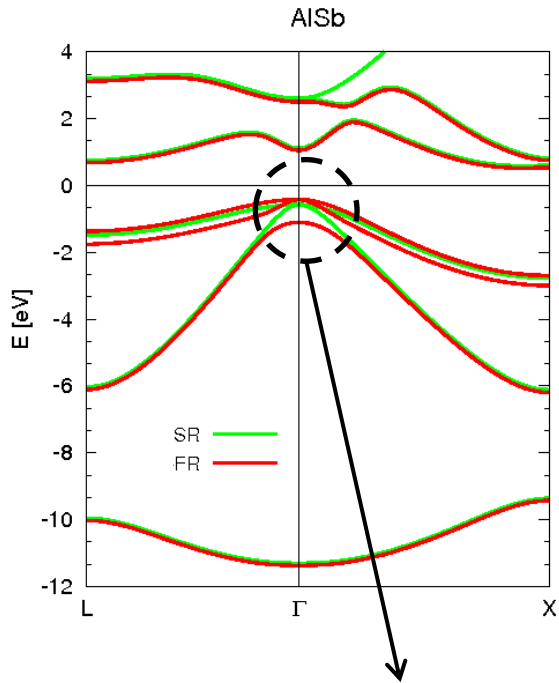
★ **PSML format:** <https://www.pseudo-dojo.org/>

- Spin-Orbit splittings in atoms

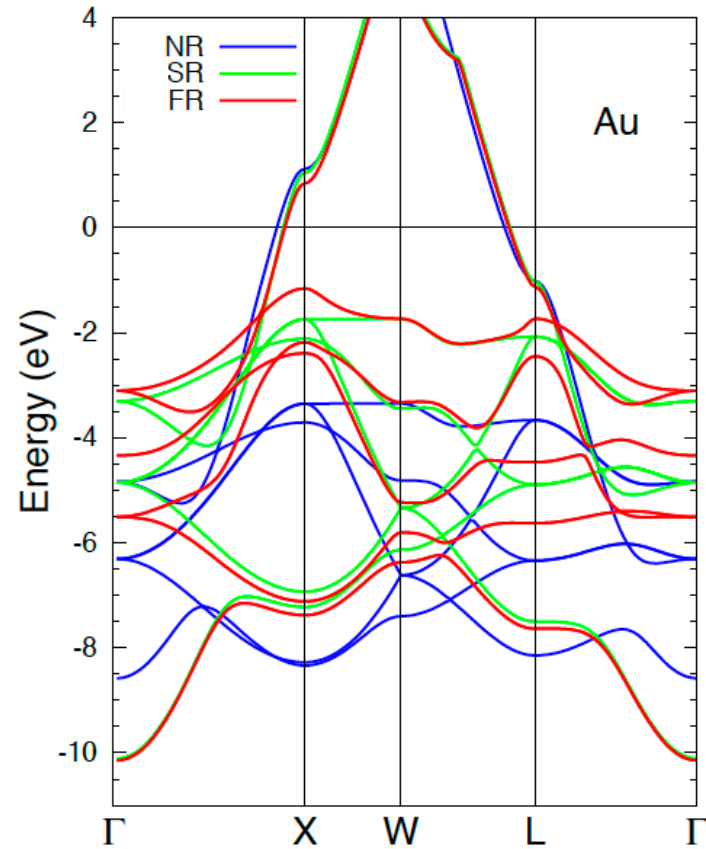
d-splitting increases when Z increases



• Bands structure



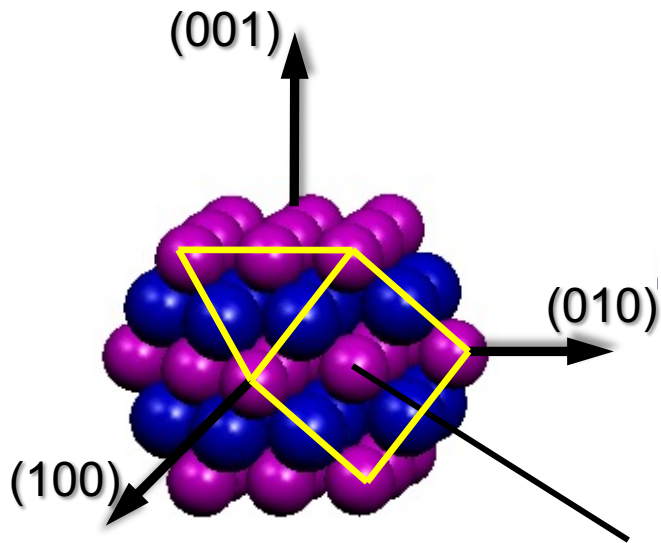
Semi.	a [Å]		Δ_0 [eV]	
AlSb	6.14	6.12	0.75	0.68
AlAs	5.66	5.67	0.28	0.28
GaAs	5.65	5.61	0.34	0.33
GaSb	6.09	6.02	0.82	0.73
Si	5.43	5.40	0.04	0.05
Ge	5.65	5.61	0.29	0.29



- SR Triple degenerate states
- FR FR calculations broke the degeneracy shifting the lower band

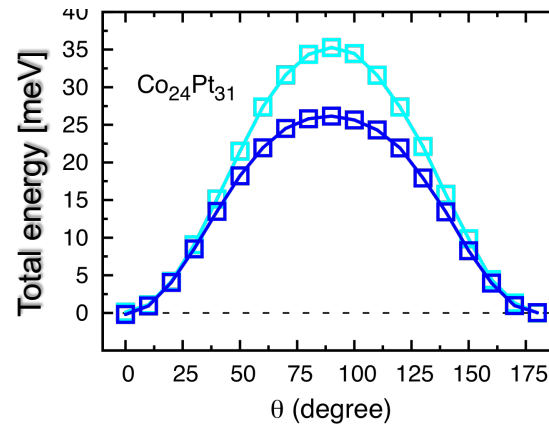
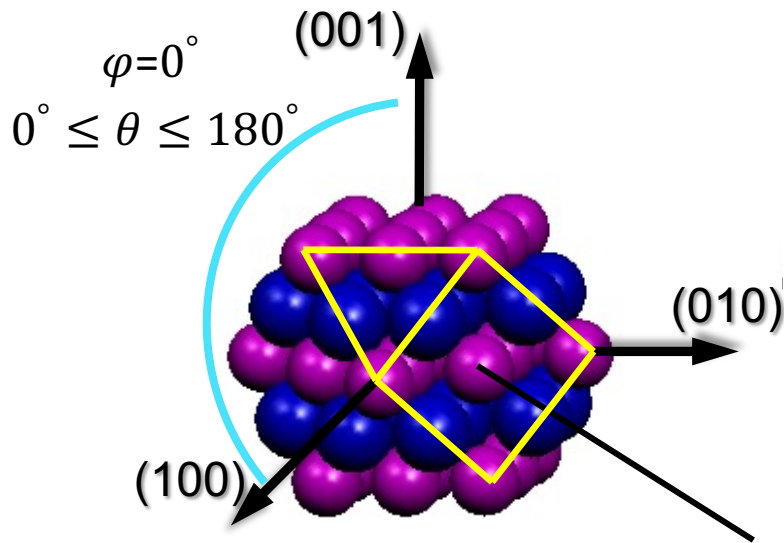
- Total energy differences between \neq orientations of the atomic spins of $\text{Co}_{24}\text{Pt}_{31}\text{-L1}_0$ NP

Total SCF energy vs. magnetizations angles



- Total energy differences between \neq orientations of the atomic spins of $\text{Co}_{24}\text{Pt}_{31}\text{-L1}_0$ NP

Total SCF energy vs. magnetizations angles



```

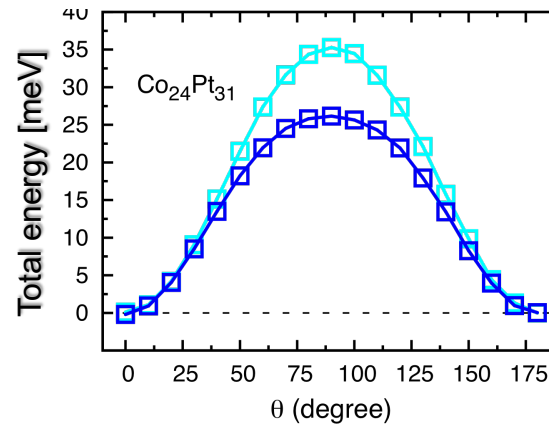
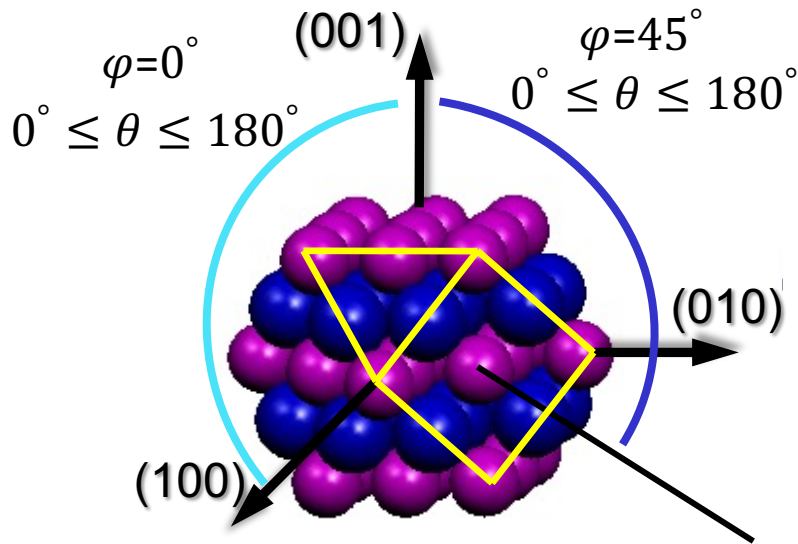
%block DM.InitSpin
  1 + 0. 0.
  2 + 0. 0.
%endblock DM.InitSpin
  
```

← Fixing phi to 0°

↑ Modifying theta for each SCF calculation

- Total energy differences between \neq orientations of the atomic spins of $\text{Co}_{24}\text{Pt}_{31}\text{-L1}_0$ NP

Total SCF energy vs. magnetizations angles



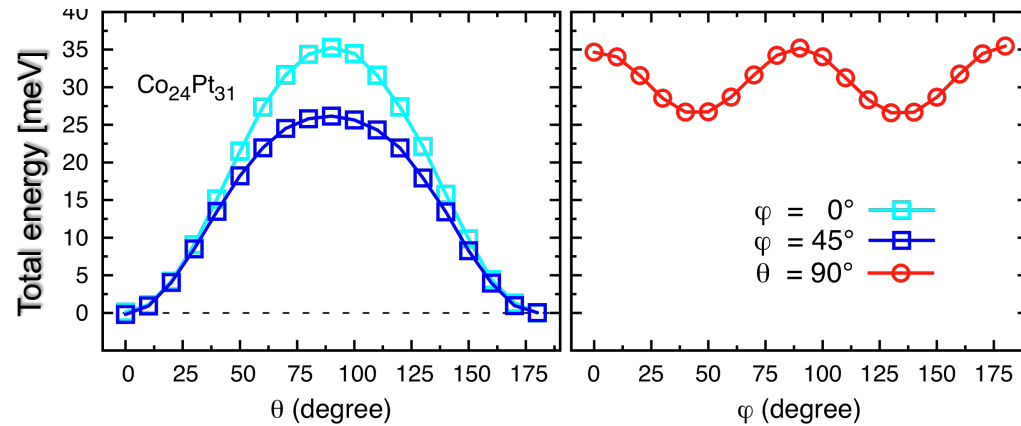
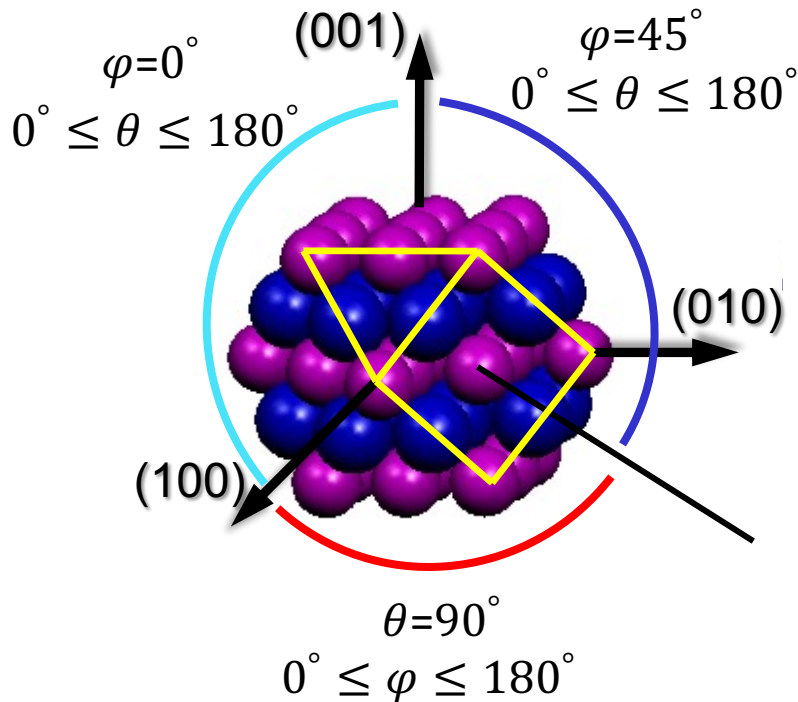
```
%block DM.InitSpin
  1 + 0. 45.
  2 + 0. 45.
%endblock DM.InitSpin
```

Fixing φ to 45°

Modifying θ for each SCF calculation

- Total energy differences between \neq orientations of the atomic spins of $\text{Co}_{24}\text{Pt}_{31}\text{-L1}_0$ NP

Total SCF energy vs. magnetizations angles



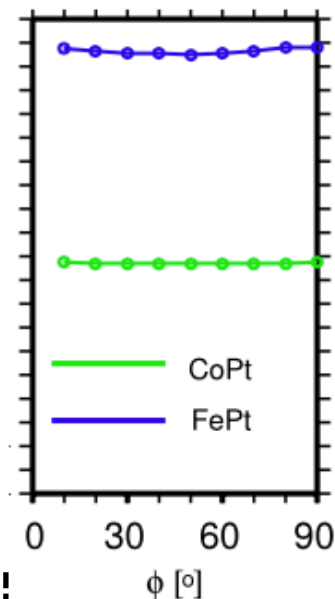
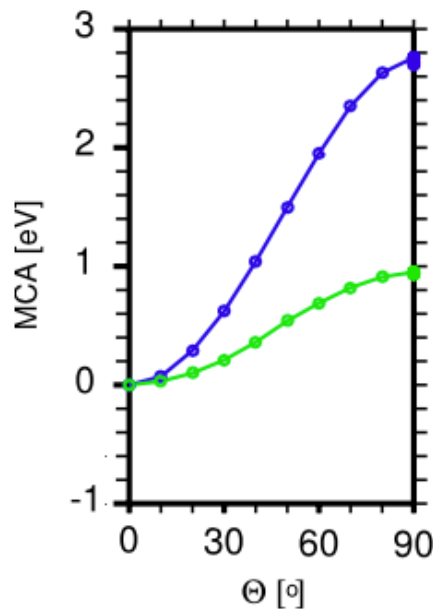
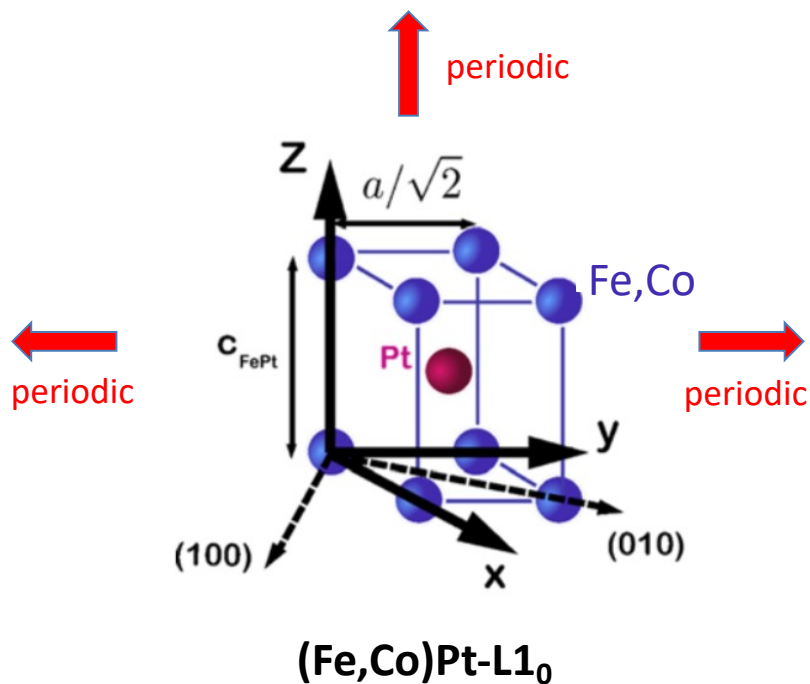
```

%block DM_InitSpin
  1 + 90. 0.
  2 + 90. 0.
%endblock DM_InitSpin
  
```

Modifying phi for each SCF step

Fixing theta to 90°

• Magnetic Anisotropy Energy of bulk (Fe,Co)Pt-L1₀ alloy



```
%block DM.InitSpin
1 +1. 0. 0.
2 +1. 0. 0.
%endblock DM.InitSpin
```

Modifying theta for each SCF calculation

```
%block DM.InitSpin
1 +1. 0. 0.
2 +1. 0. 0.
%endblock DM.InitSpin
```

Modifying phi for each SCF calculation

- Spin-orbit coupling in the SIESTA manual:

SIESTA 5.4.1

September 16, 2025

6.8 Spin-Orbit coupling

SIESTA includes the option of including in the total Hamiltonian n correction terms (Scalar-Relativistic calculations), but also the **Spin** option on how to enable spin-orbit coupling.

The SO functionality has been implemented by Dr. Ramón Cuadrado and Dr. Jorge I. Cerdá based on their initial work (R. Cuadrado and J. I. Cerdá “Fully relativistic pseudopotential formalism under an atomic orbital basis: spin-orbit splittings and magnetic anisotropies”, *J. Phys.: Condens. Matter* **24**, 086005 (2012); “In-plane/out-of-plane disorder influence on the magnetic anisotropy of $\text{Fe}_{1-y}\text{Mn}_y\text{Pt-L1}(0)$ bulk alloy”, R. Cuadrado, Kai Liu, Timothy J. Klemmer and R. W. Chantrell, *Applied Physics Letters*, **108**, 123102 (2016)).

The inclusion of the SO term in the Hamiltonian (and in the Density Matrix) causes an increase in the number of non-zero elements in their off-diagonal parts, i.e., for some (μ, ν) pair of basis orbitals, $\mathbf{H}_{\mu\nu}^{\sigma\sigma'}$ ($\mathbf{DM}_{\mu\nu}^{\sigma\sigma'}$) [$\sigma, \sigma' = \uparrow, \downarrow$] will be $\neq 0$. This is mainly due to the fact that the $\mathbf{L} \cdot \mathbf{S}$ operator will promote the mixing between different spin-up/down components. In addition, these $\mathbf{H}_{\mu\nu}^{\sigma\sigma'}$ (and $\mathbf{DM}_{\mu\nu}^{\sigma\sigma'}$) elements will be complex, in contrast with typical polarized/non-polarized calculations where these matrices are purely real. Since the spin-up and spin-down manifolds are essentially mixed, the solver has to deal with matrices whose dimensions are twice as large as for the collinear (unmixed) spin problem. Due to this, we advise to take special attention to the memory needed to perform a spin-orbit calculation.

Apart from the study of effects of the spin-orbit interaction in the band structure, a feature enabled by a SO formalism is the computation of the Magnetic Anisotropy Energy (MAE): it can be obtained as the difference in the total selfconsistent energy in two different spin orientations, usually along the easy axis and the hard axis. In SIESTA it is possible to perform calculations for different magnetization orientations using the block **DM.InitSpin** in the `fdf` file. In doing so one will be able to include the initial orientation angles of the magnetization for each atom, as well as an initial value of their net magnetic moments. See also the recent review^[7].

- Some tips about *SOC* SIESTA runs

6.7 Spin polarization

Spin non-polarized

(string)

deprecates: **SpinPolarized**, **NonCollinearSpin**, **SpinOrbit**

Choose the spin-components in the simulation.

NOTE: This flag has precedence over **SpinOrbit**, **NonCollinearSpin** and **SpinPolarized** while these deprecated flags may still be used.

non-polarized Perform a calculation with spin-degeneracy (only one component).

polarized Perform a calculation with colinear spin (two spin components).

non-colinear Perform a calculation with non-colinear spin (4 spin components), up-down and angles.

Refs: T. Oda et al, PRL, **80**, 3622 (1998); V. M. García-Suárez et al, Eur. Phys. Jour. B **40**, 371 (2004); V. M. García-Suárez et al, Journal of Phys: Cond. Matt **16**, 5453 (2004).

spin-orbit Performs calculations including the spin-orbit coupling. By default the full SO option is set. To perform an on-site SO calculation (see 6.8.1) this option has to be **spin-orbit+onsite**. This requires the pseudopotentials to be relativistic.

See Sect. 6.8 for further specific spin-orbit options.

SIESTA can read a .DM with different spin structure by adapting the information to the currently selected spin multiplicity, averaging or splitting the spin components equally, as needed. This may be used to greatly increase convergence.

Certain options may not be used together with specific parallelization routines.

- Some tips about *SOC SIESTA* runs

```
%block DM.InitSpin (None) (block)
```

Define the initial spin density for a spin polarized calculation atom by atom. In the block there is one line per atom to be spin-polarized, containing the atom index (integer, ordinal in the block **AtomicCoordinatesAndAtomicSpecies**) and the desired initial spin-polarization (real, positive for spin up, negative for spin down). A value larger than possible will be reduced to the maximum possible polarization, keeping its sign. Maximum polarization can also be given by introducing the symbol + or - instead of the polarization value. There is no need to include a line for every atom, only for those to be polarized. The atoms not contemplated in the block will be given non-polarized initialization.

For non-collinear spin, the spin direction may be specified for each atom by the polar angle θ and the azimuthal angle ϕ (using the physics ISO convention), given as the last two arguments in degrees. If not specified, $\theta = 0$ is assumed (z -polarized). **Spin** must be set to use non-collinear or spin-orbit for the directions to have effect.

Example:

```
%block DM.InitSpin
  5 -1. 90. 0. # Atom index, spin, theta, phi (deg)
  3 + 45. -90.
  7 -
%endblock DM.InitSpin
```

In the above example, atom 5 is polarized in the x -direction.

- Some tips about *SOC SIESTA* runs

```
%block DM.InitSpin (None) (block)
```

Define the initial spin density for a spin polarized calculation atom by atom. In the block there is one line per atom to be spin-polarized, containing the atom index (integer, ordinal in the block **AtomicCoordinatesAndAtomicSpecies**) and the desired initial spin-polarization (real, positive for spin up, negative for spin down). A value larger than possible will be reduced to the maximum possible polarization, keeping its sign. Maximum polarization can also be given by introducing the symbol + or - instead of the polarization value. There is no need to include a line for every atom, only for those to be polarized. The atoms not contemplated in the block will be given non-polarized initialization.

For non-collinear spin, the spin direction may be specified for each atom by the polar angle θ and the azimuthal angle ϕ (using the physics ISO convention), given as the last two arguments in degrees. If not specified, $\theta = 0$ is assumed (z -polarized) Spin must be set to use non-collinear or spin-orbit for the directions to have effect.

Example:

```
%block DM.InitSpin
  5 -1.  90.  0.  # Atom index, spin, theta, phi (deg)
  3 +   45. -90.
  7 -
%endblock DM.InitSpin
```

In the above example, atom 5 is polarized in the x -direction.

- Some tips about SOC SIESTA runs

- Total Energy calculations including E^{SO} :

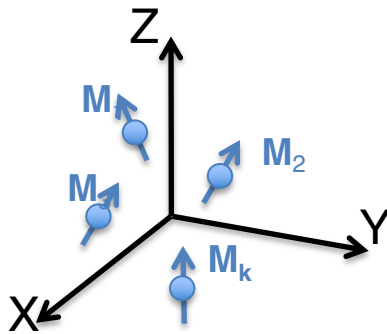
$$E^{SO} = \text{Tr}(\hat{\rho}\hat{V}^{SO}) = \sum_{\mu\nu\sigma\sigma'} \rho_{\mu\nu}^{\sigma\sigma'} V_{\nu\mu}^{SO,\sigma'\sigma}$$

- SOC contribution to the Forces:
(you can perform optimizations with SOC)

$$F_I^{SO} = - \sum_{\sigma\sigma'} \frac{\partial E^{SO,\sigma\sigma'}}{\partial \mathbf{R}_I}$$

- Bands structure.

- Mulliken or other population analysis:



$$\begin{cases} M_k^x = \text{Tr}[\rho(\mathbf{r}) \cdot \sigma_x] \\ M_k^y = \text{Tr}[\rho(\mathbf{r}) \cdot \sigma_y] \\ M_k^z = \text{Tr}[\rho(\mathbf{r}) \cdot \sigma_z] \end{cases}$$

```

siesta: Program's energy decomposition (eV):
siesta: Ebs      = -186.088668
siesta: Eions   = 2552.419586
siesta: Ena     = 75.153146
siesta: Ekin    = 4650.897453
siesta: Enl    = -3816.826902
siesta: Eso(offS) = -0.045988
siesta: Etdau   = 0.000000
siesta: DEna   = -21.150000
siesta: DUscf  = 7.867448
siesta: DUext  = 0.000000
siesta: Exc    = -1614.095394
siesta: eta*DQ = 0.000000
siesta: Emadel = 0.000000
siesta: Emeta  = 0.000000
siesta: Emolmec = 0.000000
siesta: Ekinion = 0.000000
siesta: Eharris = -3270.509003
siesta: Etot   = -3270.509003
siesta: FreeEng = -3270.520677
    
```

- Projected density of states.

- DFT+U+SOC

- D3 dispersion correction+SOC

Species: Fe

Atom	Orb	Charge	Spin	Svec		
25	1 4s	-0.18131	0.01610	0.000	0.000	0.016
25	2 4s	0.61076	0.03126	0.000	-0.001	0.031
25	3 3dxy	1.13211	0.54063	-0.003	-0.001	0.541
25	4 3dyz	1.11510	0.59972	0.001	0.006	0.600
25	5 3dz2	1.21548	0.49411	-0.003	-0.005	0.494
25	6 3dxz	1.16038	0.39366	-0.003	0.000	0.394
25	7 3dx2-y2	1.23344	0.48719	0.000	-0.002	0.487
25	8 3dxy	0.11818	0.09901	0.000	0.000	0.099
25	9 3dyz	0.12831	0.11494	0.000	0.000	0.115
25	10 3dz2	0.12127	0.10262	-0.001	0.000	0.103
25	11 3dxz	0.08961	0.07738	0.000	0.000	0.077
25	12 3dx2-y2	0.11431	0.09747	0.000	0.000	0.097
25	13 4ppy	0.16868	0.02443	0.000	0.000	0.024
25	14 4ppz	0.17519	0.01665	0.000	0.000	0.017
25	15 4ppx	0.12020	0.02829	0.000	0.000	0.028
25	Total	7.32171	3.12337	-0.009	-0.002	3.123

C: Electronic structure analysis and convergence

- Charge and magnetic moments (MMs)
- Projected density of states (PDOS)
- Bands structure
- Convergence tips

OUTPUTS

- Output spin resolved charges (**collinear spins**)

mulliken: Atomic and Orbital Populations:

mulliken: Spin UP

Species: Fe

Atom	Qatom	Qorb	4s	4s	3dxy	3dyz	3dz2	3dxz	3dx2-y2	3dxy
			3dyz	3dz2	3dxz	3dx2-y2	4Ppy	4Ppz	4Ppx	
33	5.380	-0.083	0.277	0.829	0.834	0.843	0.840	0.838	0.137	
		0.134	0.144	0.141	0.151	0.096	0.101	0.100		
34	5.380	-0.084	0.277	0.830	0.834	0.843	0.840	0.838	0.136	
		0.134	0.144	0.141	0.150	0.095	0.102	0.100		
35	5.279	0.085	0.182	0.821	0.816	0.812	0.789	0.830	0.138	
		0.144	0.146	0.127	0.145	0.079	0.078	0.088		

mulliken: Spin DOWN

Species: Fe

Atom	Qatom	Qorb	4s	4s	3dxy	3dyz	3dz2	3dxz	3dx2-y2	3dxy
			3dyz	3dz2	3dxz	3dx2-y2	4Ppy	4Ppz	4Ppx	
33	1.907	-0.142	0.327	0.369	0.330	0.242	0.280	0.211	0.006	
		0.005	0.005	0.002	0.004	0.092	0.087	0.089		
34	1.910	-0.142	0.327	0.374	0.332	0.240	0.280	0.210	0.007	
		0.005	0.005	0.002	0.004	0.091	0.087	0.089		
35	2.078	0.108	0.156	0.435	0.355	0.247	0.314	0.196	0.017	
		0.019	0.003	0.011	-0.001	0.073	0.072	0.074		

- Output spin resolved charges (**collinear spins**)

mulliken: Atomic and Orbital Populations:

mulliken: Spin UP

Species: Fe

Atom	Qatom	Qorb	4s	4s	3dxy	3dyz	3dz2	3dxz	3dx2-y2	3dxy
			3dyz	3dz2	3dxz	3dx2-y2	4Ppy	4Ppz	4Ppx	
33	5.380	-0.083	0.277	0.277	0.829	0.834	0.843	0.840	0.838	0.137
		0.134	0.144	0.141	0.151	0.096	0.101	0.100		
34	5.380	-0.084	0.277	0.277	0.830	0.834	0.843	0.840	0.838	0.136
		0.134	0.144	0.141	0.150	0.095	0.102	0.100		
35	5.279	0.085	0.182	0.182	0.821	0.816	0.812	0.789	0.830	0.138
		0.144	0.146	0.127	0.145	0.079	0.078	0.088		

mulliken: Spin DOWN

Species: Fe

Atom	Qatom	Qorb	4s	4s	3dxy	3dyz	3dz2	3dxz	3dx2-y2	3dxy
			3dyz	3dz2	3dxz	3dx2-y2	4Ppy	4Ppz	4Ppx	
33	1.907	-0.142	0.327	0.327	0.369	0.330	0.242	0.280	0.211	0.006
		0.005	0.005	0.002	0.004	0.092	0.087	0.089		
34	1.910	-0.142	0.327	0.327	0.374	0.332	0.240	0.280	0.210	0.007
		0.005	0.005	0.002	0.004	0.091	0.087	0.089		
35	2.078	0.108	0.156	0.156	0.435	0.355	0.247	0.314	0.196	0.017
		0.019	0.003	0.011	-0.001	0.073	0.072	0.074		

- At the end of the Selfconsistency

$$MM_{Z/at} = Q_Z^{\uparrow} - Q_Z^{\downarrow} [\mu_B/at]$$



Mulliken Atomic Populations:

Atom #	charge [q]	valence [e]	Sz [e]	Species
1	-0.131555	6.131555	0.000181	0
2	-0.296255	6.296255	0.000367	0
3	-0.307736	6.307736	-0.000035	0
4	-0.315978	6.315978	-0.000386	0
5	0.624973	3.375027	-0.000782	Ti
6	0.568902	3.431098	-0.000215	Ti
7	-0.153160	4.153160	0.000151	C

- Output charges and the components of the spin moments (**non-collinear spins**)

mulliken: Atomic and Orbital Populations:

Species: Fe

Atom	Orb	Charge	Spin	Svec		
33	1 4s	-0.19681	0.05819	0.000	0.000	0.058
33	2 4s	0.57426	0.04693	0.000	0.000	-0.047
33	3 3dxy	1.19883	0.47229	0.001	-0.001	0.472
33	4 3dyz	1.16429	0.51568	0.001	-0.004	0.516
33	5 3dz2	1.08874	0.61043	-0.002	-0.002	0.610
33	6 3dxz	1.12054	0.57207	0.004	-0.001	0.572
33	7 3dx2-y2	1.05433	0.63646	0.000	0.000	0.636
33	8 3dxy	0.13853	0.11908	0.000	0.000	0.119
33	9 3dyz	0.13454	0.11858	0.000	-0.001	0.119
33	10 3dz2	0.14139	0.13090	0.000	0.000	0.131
33	11 3dxz	0.13788	0.12690	0.000	0.000	0.127
33	12 3dx2-y2	0.14683	0.13812	0.000	0.000	0.138
33	13 4Ppy	0.18863	0.00747	0.000	0.000	0.007
33	14 4Ppz	0.18812	0.01902	0.000	0.000	0.019
33	15 4Ppx	0.19019	0.01364	0.000	0.000	0.014
33	Total	7.27029	3.49189	0.006	-0.010	3.492

$$M_k^x = Tr[\rho(\mathbf{r}) \cdot \sigma_x]$$

$$M_k^y = Tr[\rho(\mathbf{r}) \cdot \sigma_y]$$

$$M_k^z = Tr[\rho(\mathbf{r}) \cdot \sigma_z]$$

- On-the-fly spin moments:

```

      iscf      Eharris(eV)          E_KS(eV)      FreeEng(eV)    dDmax    Ef(eV) dHmax(eV)
scf:   1  -33630.670822    -33630.670878    -33630.670878  0.000778 -5.051140  0.020317
      spin moment: S , {S} = 27.97126 -0.01855  0.01795 -27.97125
timer: Routine,Calls,Time,% = IterSCF      1    111.020  65.56
scf:   2  -33630.631719    -33630.658045    -33630.658045  0.007744 -5.051169  0.416894
      spin moment: S , {S} = 27.97105 -0.02076  0.02091 -27.97104
scf:   3  -33630.671552    -33630.670909    -33630.670909  0.007393 -5.051265  0.005304
      spin moment: S , {S} = 27.97125 -0.01865  0.01809 -27.97124
scf:   4  -33630.670909    -33630.670910    -33630.670910  0.000036 -5.051307  0.004707
      spin moment: S , {S} = 27.97124 -0.01873  0.01819 -27.97123

```

SpinInSCF = True ← By default

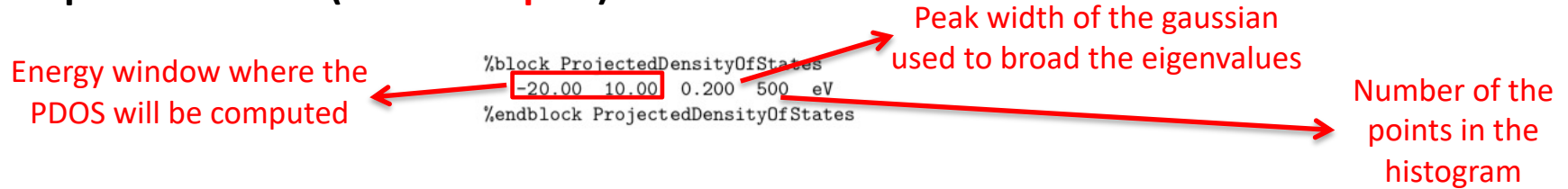
Note: Also for Spin Polarized case!

Additional feature when SO is included:

- Possibility to calculate the Orbital Moments by means of the flag:

WriteOrbMom = True

- Output PDOS files (**collinear spins**)



SystemLabel.DOS (Total DOS)

Energy(eV)	DOS Spin up (eV ⁻¹)	DOS Spin down (eV ⁻¹)
-19.99998	0.35533	0.35459
-19.93986	0.36960	0.36852
-19.87974	0.42308	0.42201
-19.81962	0.50413	0.50344
-19.75950	0.58919	0.58909
-19.69938	0.65232	0.65284
⋮	⋮	⋮

$$Q_Z^\uparrow = \int_{-\infty}^{E_F} DOS_Z^\uparrow(\epsilon) d\epsilon$$

$$Q_Z^\downarrow = \int_{-\infty}^{E_F} DOS_Z^\downarrow(\epsilon) d\epsilon,$$

$$M_Z/at = Q_Z^\uparrow - Q_Z^\downarrow \quad [\mu_B/at]$$

SystemLabel.PDOS (orbital)

```

<pdos>
<nspin>2</nspin>
<norbitals> 242</norbitals>
<energy_values units="eV">
-19.99998
-19.93986
-19.87974
-19.81962
⋮
</energy_values>
<orbital
index="
atom_index="
species="Fe"
position=" 0.000000 -2.963092 32.732488"
n="
l="
m="
z="
>
<data>
0.00023 0.00020
0.00020 0.00017
0.00019 0.00016
0.00020 0.00017
0.00020 0.00017
⋮
  
```

- Output PDOS files (**non-collinear spins**)

Same as SystemLabel.DOS and SystemLabel.PDOS but now there are five columns

Energy(eV) DOS Spin up (eV⁻¹) DOS Spin down (eV⁻¹) Re {DOS} (eV⁻¹) Im{DOS} (eV⁻¹)

$$Q_Z^\uparrow = \int_{-\infty}^{E_F} DOS_Z^\uparrow(\epsilon) d\epsilon$$

$$Q_Z^\downarrow = \int_{-\infty}^{E_F} DOS_Z^\downarrow(\epsilon) d\epsilon,$$



$$M_Z/at = Q_Z^\uparrow - Q_Z^\downarrow \quad [\mu_B/at]$$

$$M_X = Re \int_{-\infty}^{E_F} DOS^{\uparrow\downarrow}(\epsilon) d\epsilon$$

$$M_Y = Im \int_{-\infty}^{E_F} DOS^{\uparrow\downarrow}(\epsilon) d\epsilon$$

• PDOS utility

`/home/your_user_name/your_SIESTA_main_directory/Util/Contrib/APostnikov`

* `fmpdos.f`  Compile it: `make`

Run `./fmpdos` and some questions have to be answered:

- 1) Input file name (PDOS):
- 2) Output file name :
- 3) Extract data for atom index (enter atom NUMBER, or 0 to select all),
or for all atoms of given species (enter its chemical LABEL):
- 4) Extract data for n= ... (0 for all n):

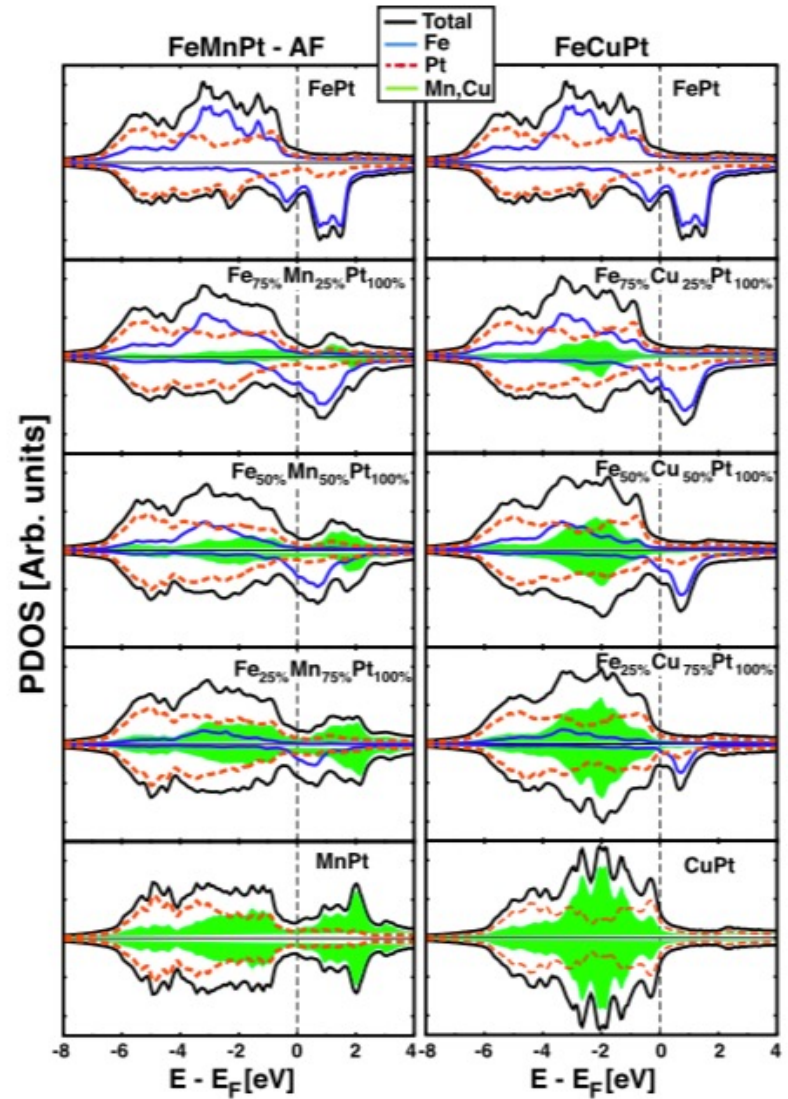
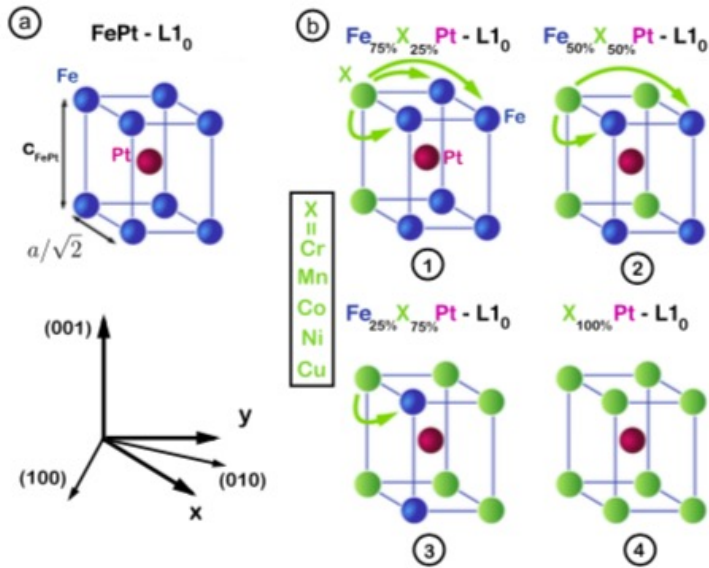
Note 1: You can build your own program to extract the PDOS

Note 2: There is another utility in `Util/pdosxml` directory

Note 3: Use the utility present in the SIESTA version that you are using.

```
# partial DOS for atom index      1
#
# Add data for atom_index = 1, n,l,m,z= 4 0 0 1
# Add data for atom_index = 1, n,l,m,z= 4 0 0 2
# Add data for atom_index = 1, n,l,m,z= 3 2 -2 1
# Add data for atom_index = 1, n,l,m,z= 3 2 -1 1
# Add data for atom_index = 1, n,l,m,z= 3 2 0 1
# Add data for atom_index = 1, n,l,m,z= 3 2 1 1
# Add data for atom_index = 1, n,l,m,z= 3 2 2 1
# Add data for atom_index = 1, n,l,m,z= 3 2 -2 2
# Add data for atom_index = 1, n,l,m,z= 3 2 -1 2
# Add data for atom_index = 1, n,l,m,z= 3 2 0 2
# Add data for atom_index = 1, n,l,m,z= 3 2 1 2
# Add data for atom_index = 1, n,l,m,z= 3 2 2 2
# Add data for atom_index = 1, n,l,m,z= 4 1 -1 1
# Add data for atom_index = 1, n,l,m,z= 4 1 0 1
# Add data for atom_index = 1, n,l,m,z= 4 1 1 1
#
# Energy          spin 1          spin2
#
-19.9999800      0.01034000      0.01018000
-19.9398600      0.01032000      0.00996000
-19.8797400      0.00976000      0.00928000
-19.8196200      0.00887000      0.00832000
-19.7595000      0.00788000      0.00744000
```

• Example:



- **Output Bands files:**

In **spin-polarized DFT without SOC** the Hamiltonian conserves S_z , so bands can be labeled as spin-up or spin-down, but **once SOC is included** S_z is no longer a good quantum number and the bands become mixed-spin states; in that case one can only obtain spin projections or the full spin texture of each band, not pure up/down labels.

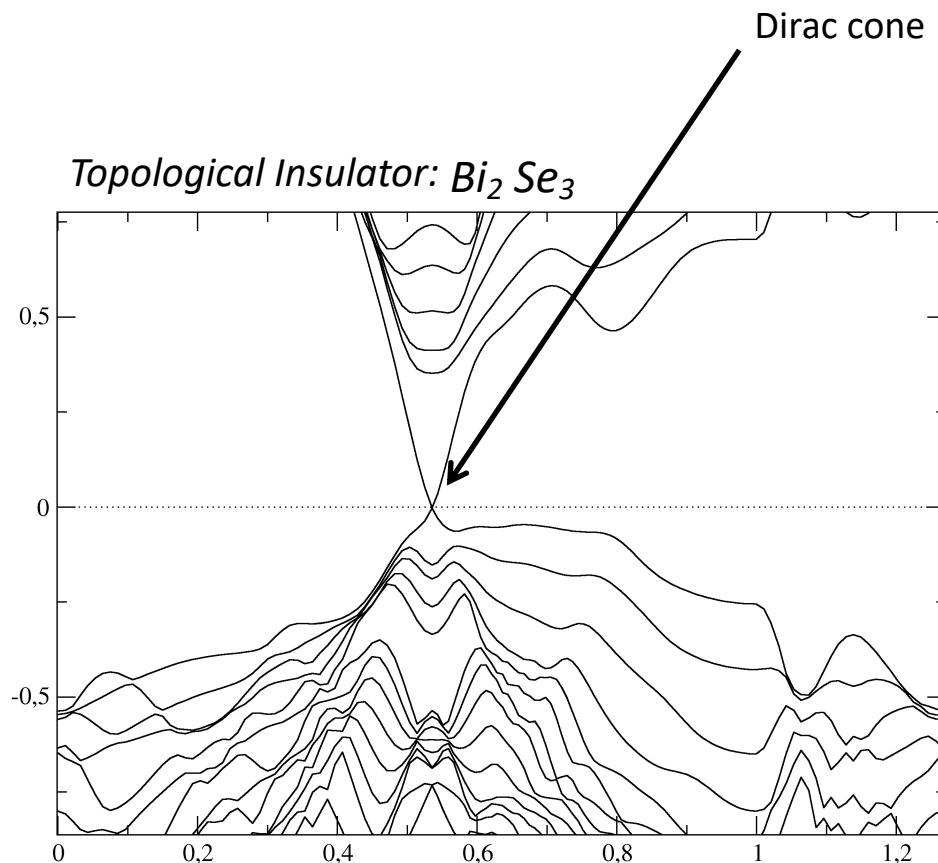
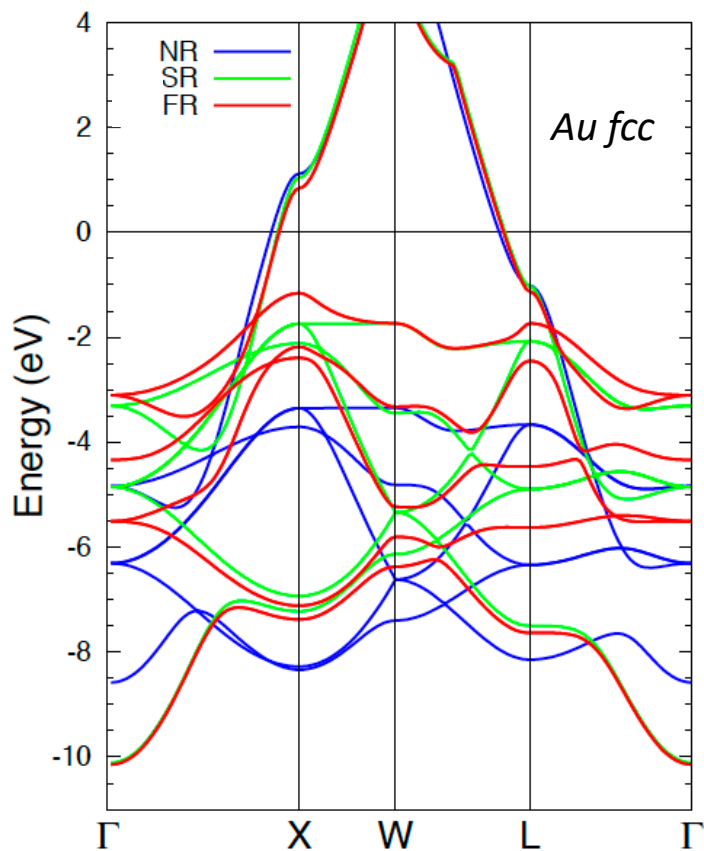
6.15.1 Format of the .bands file

```
FermiEnergy (all energies in eV) \\
kmin, kmax (along the k-lines path, i.e. range of k in the band plot) \\
Emin, Emax (range of all eigenvalues) \\
NumberOfBands, NumberOfSpins (1 or 2), NumberOfkPoints \\
k1, ((ek(iband,ispin,1),iband=1,NumberOfBands),ispin=1,NumberOfSpins) \\
k2, ek \\
. \\
. \\
. \\
klast, ek \\
NumberOfkLines \\
kAtBegOfLine1, kPointLabel \\
kAtEndOfLine1, kPointLabel \\
. \\
. \\
. \\
kAtEndOfLastLine, kPointLabel \\
```

The **gnubands** postprocessing utility program (found in the Util/**Bands** directory) reads the **SystemLabel.bands** for plotting. See the **BandLines** data descriptor above for more information.

```
4.34702858188511
0.0000000000000000E+000 1.26741316649005
-51.3339816969934 54.6001776376910
      780      1      136
0.000000 -51.3306 -51.3305 -51.3305 -51.3305 -51.3289 -51.3288 -51.3288 -51.3288 -51.3036 -51.3036
-51.3035 -51.3034 -51.3019 -51.3018 -51.3018 -51.3017 -51.2910 -51.2909 -51.2905 -51.2905
-51.2644 -51.2643 -51.2638 -51.2637 -51.2508 -51.2508 -51.2506 -51.2506 -51.2500 -51.2500
-51.2481 -51.2481 -51.2480 -51.2480 -51.2475 -51.2475 -51.2471 -51.2471 -51.2460 -51.2460
-51.2447 -51.2446 -51.2438 -51.2438 -51.2167 -51.2167 -51.2165 -51.2165 -51.2158 -51.2158
-51.2146 -51.2146 -51.2144 -51.2144 -51.2137 -51.2137 -51.2133 -51.2133 -51.2124 -51.2123
      :      :      :      :
      4
0.000000 'K'
0.535672 'Gamma'
0.999577 'M'
1.267413 'K'
```

- Examples:



- **Convergence:** *Not only in magnetism*

DFT calculations is not only pushing the button

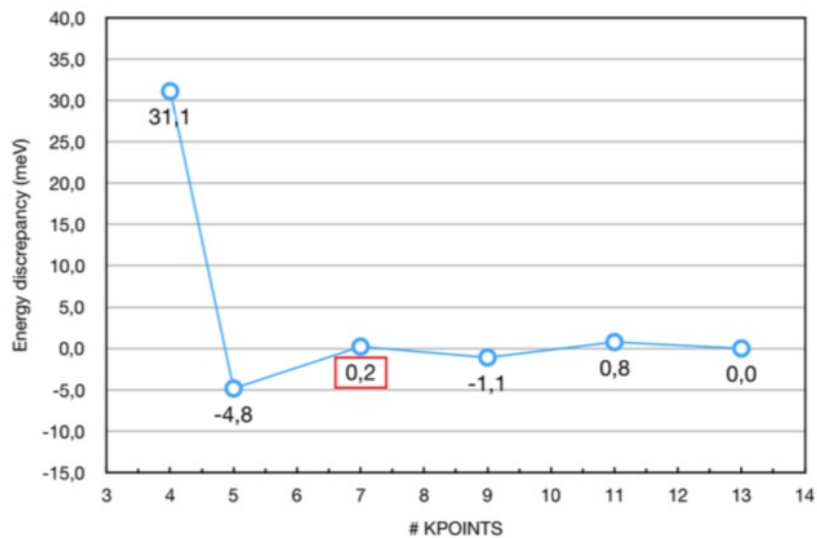
- * Have a good pseudopotential (perform some calculations for a known system)
- * Have a good basis set (same as for PPs): Simplex method, SZ, DZP, TZP, etc
- * Converge the mesh cutoff (space grid for integrals in real space)
- * Converge the number of k-points (periodically systems)
- * Converge the UC when only the gamma point is used (molecules, clusters)
- * Converge the electronic temperature kT
- * Geometry optimization
- * Converge any property you want to obtain (MAE, MMs, Bands, etc)

- **Convergence:** *Not only in magnetism*

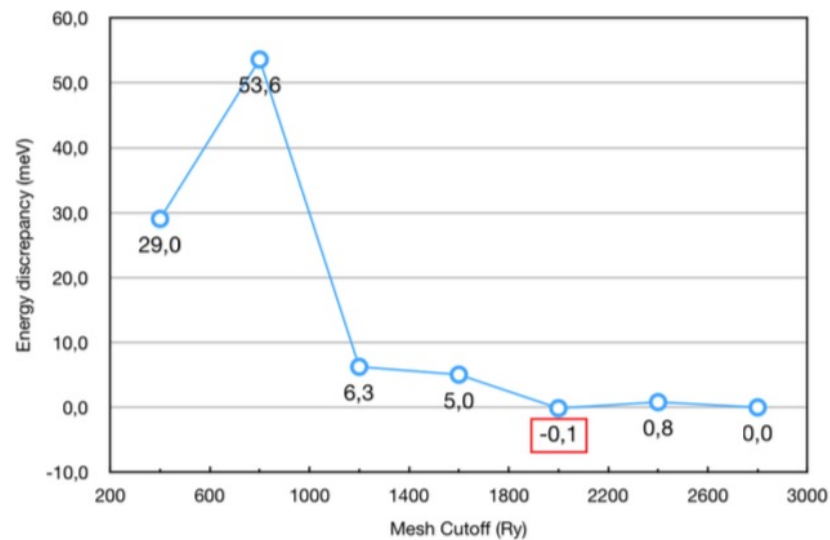
DFT calculations is not only pushing the button

- * Have a good pseudopotential (perform some calculations for a known system)
- * Have a good basis set (same as for PPs): Simplex method, SZ, DZP, TZP, etc
- * **Converge the mesh cutoff (space grid for integrals in real space)**
- * **Converge the number of k-points (periodically systems)**
- * Converge the UC when only the gamma point is used (molecules, clusters)
- * Converge the electronic temperature kT
- * Geometry optimization
- * **Converge any property you want to obtain (MAE, MMs, Bands, etc)**

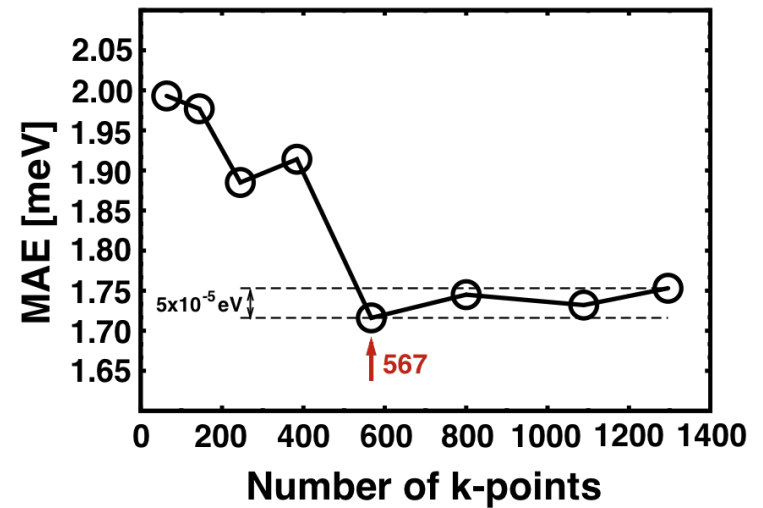
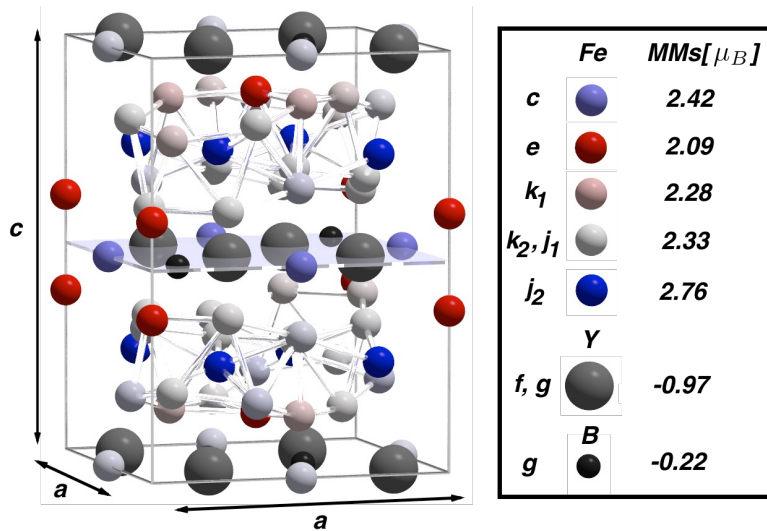
- General mesh cutoff and k-points convergence: bulk Fe_3O_4



$K_x \times K_y \times K_z$



- Convergence of MAE in the bulk YFeB alloy (68 atoms)



Conclusions

- We have different choices to make and parameters to determine, every one has to be carefully checked:
 - XC functional
 - Pseudopotential
 - Basis set
 - K-points
 - Grid cutoff
 - ...
- The best way to handle it is an incremental approach: start from a reasonable guess and converge the parameters until the desired precision is reached
- All the parameters are system dependent, everything has to be tested

TEST, TEST, TEST

Thank you very much!