EasierAxis

Authors:

Letizia Fiorucci, Enrico Ravera

Version: 0.1

Documentation release date:

July 23, 2023

Contents

1. Introduction

`EasierAxis` is a python package for the determination of the optimal orientation of a point charge (PC) distribution for a tripositive lanthanoid complexes.

The development of the program started with `python 3.8` and therefore it is compatible with that version. Nevertheless, it has been also tested on `python 3.10`. Concerning the OS, `EasierAxis` has been developed on Ubuntu 22.04 LTS and also tested on Windows 10. Other OSs should encounter no problems. The requirements for the code to run are very few: `numpy` (tested on v1.22.3), `scipy` (tested on v3.5.1) and `matplotlib` (tested on v1.8.0). The computational cost for this code is negligible.

The code is extensively commented and the implemented equations are reported in section 1.1, or referenced to existing literature. For these reasons, the functions here reported can be easily integrated by the users in their own scripts.

1.1 Theoretical background

In the context of a PC model, the potential energy can be expressed as:

$$E_M = \sum_{k}^{2,4,6} A_k^{JM} \frac{4\pi}{2k+1} \langle r^k \rangle \sum_{n} \frac{q_n Y_k^0(\theta_n, \phi_n)}{R_n^{k+1}} \tag{1.1}$$

where A_k^{JM} are the multipole expansion coefficients (or Stevens factors) from Sievers[1], $\langle r^k \rangle$ are the average radial integrals (tabulated in in reference[2]), R_n, θ_n and ϕ_n are the polar coordinates of each ligand, q_n is the ligand charge defined as fraction of the electronic charge and Y_k^0 is the k-degree and 0-order normalized spherical harmonic.

Since the leading term of the series is the quadrupolar one, in order to find the preferred orientation for the PC distribution it is sufficient to find the direction for which the potential energy to the second order is the lowest. The following $\mathbf{V}$ tensor is therefore calculated and diagonalized:

$$\mathbf{V} = \sum_{n} \frac{q_n[3\mathbf{R}_n\mathbf{R}_n^T - R_n^2\mathbf{I}]}{R_n^5} \tag{1.2}$$

[1]J Sievers. "Asphericity of 4 f-shells in their Hund's rule ground states". In: *Zeitschrift fur Physik B Condensed Matter* 45.4 (1982), pp. 289–296.

[2]Sverker Edvardsson and Mattias Klintenberg. "Role of the electrostatic model in calculating rare-earth crystal-field parameters". In: *Journal of alloys and compounds* 275 (1998), pp. 230–233.

The eigenvalues of this tensor are multiplied by the A_2^{JM} coefficients for $M = 0$ (or $M = 1/2$ in case of Kramers ions) and $M = J$, and the ground term shape of the electron distribution will be dictated by the M correspondant to the lowest eigenvalue. The easy axis will be the associated eigenvector. Since $\mathbf{V}$ is symmetric by construction, its eigenvector basis forms an orthogonal transformation, therefore the PC coordinates can be rotated in its reference frame.

The M state that contributes the most to the ground state wavefunction is assessed by accounting for the higher rank contributions using equation (1.1).

2. User guide

`EasierAxis` input and output files are depicted in the following section for the [Dy(bbpen)Cl] example[1].

The code can be executed typing the following line in the shell from the code folder:

```
$ python3 EasierAxis.py Dybbpn.inp f9
```

where `Dybbpn.inp` is the relative or absolute path to the input filename and `f9` is the number of f electrons for the specific lanthanoid ion (dysprosium(III) in this case).

The input file is structured as follows:

```
Cl -1 0 -0.219172 -2.70291 -0.040269
O -1 0 2.037979 -0.038609 -0.6876060
O -1 0 -1.86217 0.376778 0.935724
```

The columns are separated by a tabulation character. For each row: the first entry is the atom label, followed by the ligand charge (with its sign), the charge displacement along the metal-ligand bond axis, and the ligand coordinates (x, y and z). The metal center is assumed to be in the origin of the coordinate system, i.e. in (0, 0, 0). The atom labels can be any combination of numbers and letters, however in the final plot only the ligands indicated with chemical symbols will be displayed according to the color code shown in figure 2.1.

The output file is saved as *output.out* and it is structured as follows:

```
***********************************************************
**                 ============                        **
**                 EasierAxis                          **
**                 ============                        **
**                                                     **
** Authors: LETIZIA FIORUCCI, ENRICO RAVERA            **
** email: fiorucci@cerm.unifi.it, ravera@cerm.unifi.it **
** CERM and Department of Chemistry ``Ugo Schiff'',    **
** University of Florence (Italy)                      **
***********************************************************


Configuration: f9
```

---

[1]Francielli S Santana et al. "A dysprosium single molecule magnet outperforming current pseudocontact shift agents". In: *Chemical Science* 13.20 (2022), pp. 5860–5871.

(L=5, S=2.5, J=7.5)

Input filename: Dybbpn.inp
label charge r_disp x y z
Cl -1.0000 0.0000000 -0.2191720 -2.7029100 -0.0402690
O -1.0000 0.0000000 2.0379790 -0.0386090 -0.6876060
O -1.0000 0.0000000 -1.8621700 0.3767780 0.9357240

Multipole moments:
M A2 A4 A6
7.5 -0.333333 -0.121212 0.058275
6.5 -0.200000 0.040404 -0.104895
5.5 -0.085714 0.098124 -0.034965
4.5 0.009524 0.089244 0.052896
3.5 0.085714 0.044844 0.077999
2.5 0.142857 -0.010212 0.040344
1.5 0.180952 -0.057276 -0.022413
0.5 0.200000 -0.083916 -0.067240

V tensor:
0.259771 -0.042416 -0.213771
-0.042416 -0.096390 0.028780
-0.213771 0.028780 -0.163380

Eigenvalues * A2 (M=7.5): [0.08441974 0.03376726 -0.118187]
Eigenvalues * A2 (M=0.5): [-0.05065184 -0.02026035 0.0709122]

Ordered Eigenvalues * A2: [0.08441974 0.03376726 -0.118187]
Ordered Eigenvectors: (molecular frame)
0.379055 -0.127899 -0.916493
-0.066834 -0.991600 0.110738
0.922958 -0.019277 0.384419

Quadrupolar approximation: OBLATE
Lower M: 7.5
Rotated coordinates:
label x y z
Cl 0.0604024 2.7090132 -0.1139267
O 0.1404551 -0.2091162 -2.1363974
O 0.1325872 -0.1534810 2.1080992

****

The first part of the output file shows a summary of the input parameters, i.e. the chosen configuration (that can range from f1 to f13, excluded f6 and f7), the input file name and a
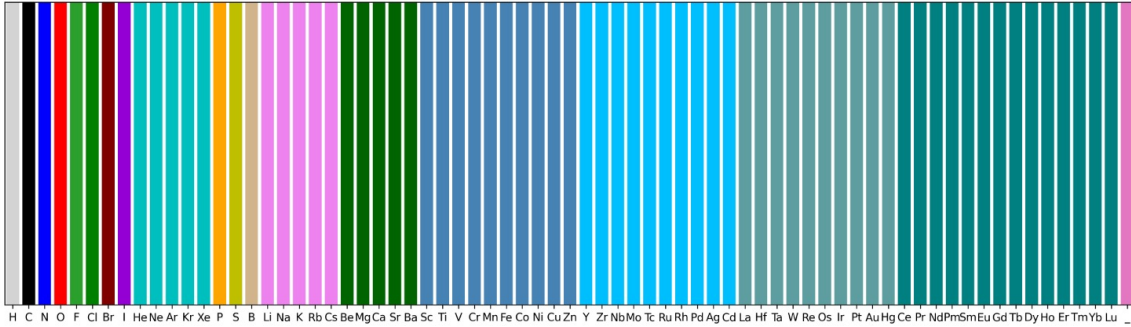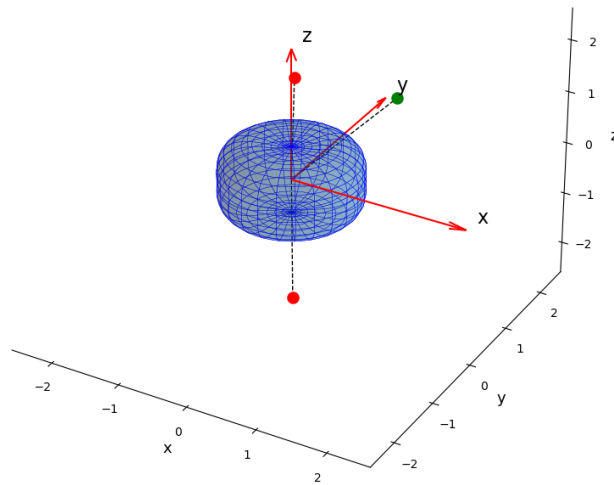
Figure 2.1: Color palette used for ligands.

Figure 2.2: Plot of Sievers surfaces in the **V** reference frame for the [Dy(bbpen)Cl] complex. In this case the M ground state is $|{\pm}15/2\rangle$. The red vectors are the eigenvectors of the **V** tensor and the dots represent the coordinating atoms contributing to the crystal field, colored according to the palette in figure 2.1.

copy of the input file text.

The following section describes the details of the performed calculation. Specifically, it reports the calculated A_k^{JM} parameters, the **V** tensor with the chosen eigenvalues and the ordered eigenvectors.

The final part shows: the type of quadrupolar distortion, the M ground state prediction according to equation (1.1) and the rotated coordinates, which are the ligand coordinates rotated by the eigenvector matrix. If r_{disp} are different from 0, the corresponding ligand coordinates will be "displaced" of the same extent also in the rotated coordinates.

Warning: the rotated coordinates contain also the displacement correction.

At the end of the calculation, before the output is saved, the program generates a plot like the one shown in figure 2.2 from which the figure can be inspected and saved, which depicts the Sievers surfaces in the **V** reference frame. The red vectors are the eigenvectors of the **V** tensor and the dots represent the coordinating atoms contributing to the crystal field.

The program also generates two other files:

- *simpre.dat*, to run a `SIMPRE` calculation of crystal field parameters, energy levels and

eigenfunctions[2]

- *EasyAxis.bild*, to plot the calculated easy axis on the original molecular reference system in an external molecular visualization software.

The only ligands reported in the *simpre.dat* file are the negatively charged ones, since SIMPRE does not handle positive charges.

---

[2]Jose J Baldovi et al. "An updated version of the computational package SIMPRE that uses the standard conventions for S tevens crystal field parameters". In: *Journal of Computational Chemistry* 35.26 (2014), pp. 1930–1934.

3. List of modules and functions

3.1 EFG_tensor(data)

Computes the electric field gradient tensor V according to equation 2 in the manual.

Parameters:

- data : *2darray*
  array of shape (n,5) where n is the number of ligands. Each row consists of: [label, x, y, z, charge]

Returns:

- V : *2darray*
  electric field gradient tensor

3.2 EasierAxis_v0.Wigner_coeff   class

This class is a wrapper around the functions to compute angular momentum coupling coefficients

Methods:

sixj_symbol(matrix)

Computes 6j-symbols using Racah's formula. Racah, Giulio. Physical Review 62.9-10 (1942): 438.

Parameters:

- matrix : *list of lists*
  angular momentum parameters in the form [[a, b, c],[A, B, C]] or [[j1, j2, j3],[j4, j5, j6]]

Returns:

- result : *float*
  computed 6j-symbol

---

threej_symbol(matrix)

Computes 3j-symbols using Racah's formula. Racah, Giulio. Physical Review 62.9-10 (1942): 438.

Parameters:

- matrix : *list of lists*
  angular momentum parameters in the form [[a , b, c],[A, B, C]] or [[j1, j2, j3],[m1, m2, m3]]

Returns:

- result : *float*
  computed 3j-symbol

3.3   coeff_multipole_moments(conf, J, M, L=0, S=0)

Computes the multipole expansion coefficients ($A_k = \mathrm{sqrt}(4pi/(2k+1)) \cdot c_k$) of a 4f electron-density distribution for a given term of a fn configuration. The equations used are eq. 10 and 12 from Ch. II p. 290 of Sievers, J. Zeitschrift für Physik B Condensed Matter 45.4 (1982): 289-296.

Parameters:

- conf : *str*
  fn configuration e.g. 'f3'

- J : *float*
  total angular momentum

- M : *float*
  magnetic angular momentum

- L : *float*
  orbital angular momentum

- S : *float*
  spin angular momentum

Returns:

- coeff : *list*
  list of Ak coefficients

3.4 consecutive_eq_el(arr)

Find consecutive equal elements in an array and return their indices in a list of lists.

Parameters:

- arr : *list*
  The input list containing elements to be checked for consecutive equality.

Returns:

- equal_elements_indices : *list of lists*
  A list of lists where each inner list contains the indices of consecutive occurrences of the same element. If no consecutive equal elements are found, an empty list is returned.

3.5 fact(number)

Computes the factorial of number

Parameters:

- number : *int*

Returns:

- factorial : *int*

3.6 figure_justcoord(data)

Plots the point charge model.

Parameters:

- data : *2darray*
  array of shape (n,5) where n is the number of ligands. Each row consists of: [label, x, y, z, charge]

3.7 freeion_charge_dist(theta, phi, A2, A4, A6, bin=1e-10)

Computation of a given point on the Sievers surface. The equation used is eq. 20 from Ch. III p 292 of Sievers, J. Zeitschrift für Physik B Condensed Matter 45.4 (1982): 289-296.

Parameters:

- theta : *float*
  polar angle in radians

- phi : *float*
  azimuthal angle in radians

- A2 : *float*
  rank-2 coefficient of multipolar expansion

- A4 : *float*
  rank-4 coefficient of multipolar expansion

- A6 : *float*
  rank-6 coefficient of multipolar expansion

- bin : *float*
  threshold below which the returned value is zero

Returns:

- coeff : *list*
  list of Ak coefficients

3.8 from_car_to_sph(coord)

Converts a set of coordinates from cartesian to spherical. They must be centered in [0,0,0] in order to work properly.

Parameters:

- coord : *2darray*
  set of cartesian coordinates

Returns:

- coord_conv : *2darray*
  set of converted spherical coordinates

3.9 from_sph_to_car(coord_sph)

Converts a set of coordinates from spherical to cartesian. The angles must be expressed in radians.

Parameters:

- coord_sph : *2darray*
  set of spherical coordinates

Returns:

- coord_car : *2darray*
  set of converted cartesian coordinates

3.10 ground_term_legend(conf)

Returns the ground state for a given fn configuration.

Parameters:

- conf : *str*
  fn configuration e.g. 'f3'

Returns:

- legenda[conf] : *str*
  ground state term

3.11 main(conf, coord, charges, labels, filename)

Main program.

Parameters:

- conf : *str*
  fn configuration e.g. 'f3'

- coord : *2darray*
  set of cartesian coordinates of ligands

- charges : *sequence*
  ligand charges (with their sign)

- labels : *sequence*
  ligand labels

- filename : *str*
  input file name

3.12   multipole_exp_tensor(conf, J, M, L, S, data)

Computes the lowest component of the multipole expansion tensor according to equation 1 of the manual.

Parameters:

- conf : *str*
  fn configuration e.g. 'f3'

- J : *float*
  total angular momentum

- M : *float*
  magnetic angular momentum

- L : *float*
  orbital angular momentum

- S : *float*
  spin angular momentum

- data : *2darray*
  array of shape (n,5) where n is the number of ligands. Each row consists of: [label, x, y, z, charge]

Returns:

- eiv : *float*
  component of multipole expansion tensor

- A : *list*
  list of Ak Sievers coefficients

3.13   plot_sievers4efg(conf, J, M, L, S, w, v, data=None)

Given a fn configuration and M, computes and plots the correspondent Sievers surface on the point charge model rotated in the reference frame of V. The eigenvectors of V are also displayed as red arrows in their reference system. The point charge model itself is displayed only if a suitable data matrix is provided.

Parameters:

- conf : *str*
  fn configuration e.g. 'f3'

- J : *float*
  total angular momentum

- M : *float*
  magnetic angular momentum

- L : *float*
  orbital angular momentum

- S : *float*
  spin angular momentum

- w : *sequence*
  eigenvalues of V

- v : *2darray*
  eigenvectors of V column-wise

- data : *2darray*
  array of shape (n,5) where n is the number of ligands. Each row consists of: [label, x, y, z, charge]

3.14   princ_comp_sort(w, v=array([[0., 0., 0.], [0., 0., 0.], [0., 0., 0.]]))

Sorts the eigenvalues of a matrix in descending order, then sorts the associated eigenvectors accordingly

Parameters:

- w : *1darray*
  matrix eigenvalues

- v : *2darray*
  matrix eigenvectors column-wise

Returns:

- wst : *1darray*
  sorted eigenvalues

- vst : *2darray*
  sorted eigenvectors column-wise

3.15   r_expect(k, conf)

Returns the average radial integral of a rare earth ion ($<r\hat{k}>$) for a given k and fn configuration. Tables from S. Edvarsson, M. Klintenberg, Journal of Alloys and Compounds. 1998, 275–277, 230.

Parameters:

- k : *int*
  spherical harmonics rank

- conf : *str*
  fn configuration e.g. 'f3'

Returns:

- legenda[k][conf] : *float*
  $<r\hat{k}>$ value for the given rank and configuration

3.16   simpre_legend(conf)

It is used to assess which SIMPRE index corresponds to a given configuration.

Parameters:

- conf : *str*
  fn configuration e.g. 'f3'

Returns:

- legenda[conf] : *int*
  SIMPRE lanthanoid ion index

3.17 state_legend(L)

Converts the term symbol in the corresponding L value if L is a str, or viceversa if L is a number.

Parameters:

- L : *str or int*
  L symbol

Returns:

- ret_val : *int or str*
  L symbol

3.18 w_bild(axis)

Creates a file named 'EasyAxis.bild', which is used to instruct molecular visualization programs (e.g. chimera) to draw the easy axis as a red cylinder.

Parameters:

- axis : *sequence*
  easy axis cartesian coordinates [x,y,z]

3.19 w_inp_dat(coord_tot, charges_tot, num)

Creates a file named 'simpre.dat', which contains coordinates and charges, to be used as input for a SIMPRE calculation. As such, only the negative charges are considered.

Parameters:

- coord_tot : *2darray*
  cartesian coordinates of the ligands

- charges_tot : *sequence*
  ligand charges (with their sign)

- num : *int*
  SIMPRE index for a configuration