

## Parámetros Mossbauer con ORCA

Omar González Franco  
fisica-unam@ciencias.unam.mx

### 1. Mossbauer con ORCA

Para utilizar ORCA hay que conectarse a una máquina remota que lo contenga. Para acceder a la máquina hay que escribir en la terminal (de Linux o con WSL en Windows) lo siguiente: ssh, el nombre del usuario y la correspondiente IP de la maquina remota, -p y el puerto de acceso a la máquina, así como la contraseña:

```
ssh orca@xxx.xxx.x.xxx -p XXX
```

```
password: xxxxxx
```

Dentro de la máquina se puede correr ORCA para hacer cálculos moleculares.

Si no se tiene instalado Secure Shell (SSH) se puede instalar escribiendo en la terminal:

```
sudo apt-get install openssh-server.
```

#### 1.1. Acerca de los parámetros Mossbauer.

##### 8.15.9 Mössbauer Parameters

$^{57}\text{Fe}$  Mössbauer spectroscopy probes the transitions of the nucleus between the  $I = \frac{1}{2}$  ground state and the  $I = \frac{3}{2}$  excited state at 14.4 keV above the ground state. The important features of the Mössbauer spectrum are the isomer shift ( $\delta$ ) and the quadrupole splitting ( $\Delta E_Q$ ). An idealized spectrum is shown in Figure 8.29.

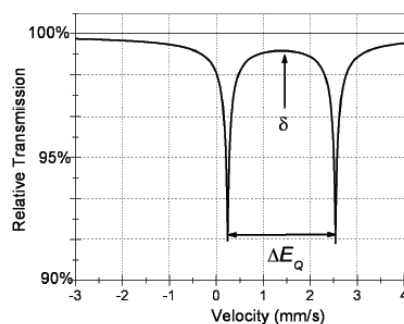


Figure 8.29: An idealized Mössbauer spectrum showing both the isomer shift,  $\delta$ , and the quadrupole splitting,  $\Delta E_Q$ .

The isomer shift measures the shift in the energy of the  $\gamma$ -ray absorption relative to a standard, usually Fe foil. The isomer shift is sensitive to the electron density at the nucleus, and indirectly probes changes in the bonding of the valence orbitals due to variations in covalency and 3d shielding. Thus, it can be used to probe oxidation and spin states, and the coordination environment of the iron.

The quadrupole splitting arises from the interaction of the nuclear quadrupole moment of the excited state with the electric field gradient at the nucleus. The former is related to the non-spherical charge distribution

in the excited state. As such it is extremely sensitive to the coordination environment and the geometry of the complex.

Both the isomer shift and quadrupole splitting can be successfully predicted using DFT methods. The isomer shift is directly related to the s electron density at the nucleus and can be calculated using the formula

$$\delta = \alpha(\rho_0 - C) + \beta \quad (8.15)$$

where  $\alpha$  is a constant that depends on the change in the distribution of the nuclear charge upon absorption, and  $\rho_0$  is the electron density at the nucleus [194]. The constants  $\alpha$  and  $\beta$  are usually determined via linear regression analysis of the experimental isomer shifts versus the theoretically calculated electron density for a series of iron compounds with various oxidation and spin states. Since the electron density depends on the functional and basis set employed, fitting must be carried out for each combination used. A compilation of calibration constants ( $\alpha$ ,  $\beta$  and  $C$ ) for various methods was assembled. [195] Usually an accuracy of better than 0.10 mm s<sup>-1</sup> can be achieved for DFT with reasonably sized basis sets.

The quadrupole splitting is proportional to the largest component of the electric field gradient (EFG) tensor at the iron nucleus and can be calculated using the formula:

$$\Delta E_Q = \frac{1}{2}eQV_{zz} \left(1 + \frac{\eta^2}{3}\right)^{\frac{1}{2}} \quad (8.16)$$

where  $e$  is the electrical charge of an electron and  $Q$  is the nuclear quadrupole moment of Fe (approximately 0.16 barns).  $V_{xx}$ ,  $V_{yy}$  and  $V_{zz}$  are the electric field gradient tensors and  $\eta$ , defined as

$$\eta = \left| \frac{V_{xx} - V_{yy}}{V_{zz}} \right| \quad (8.17)$$

is the asymmetry parameter in a coordinate system chosen such that  $|V_{zz}| \geq |V_{yy}| \geq |V_{xx}|$ .

An example of how to calculate the electron density and quadrupole splitting of an iron center is as follows:

```
%prnmr
nuclei = all Fe {fgrad, rho}
end
```

If the core properties basis set CP(PPP) is employed, one has to increase the radial integration accuracy for the iron atom, for example:

```
%basis newgto Fe "CP(PPP)" end
end

%method
SpecialGridAtoms 26
SpecialGridIntAcc 7
end
```

The output file should contain the following lines, where you obtain the calculated quadrupole splitting directly and the  $\text{RHO}(0)$  value (the electron density at the iron nucleus). To obtain the isomer shift one has to insert the  $\text{RHO}(0)$  value into the appropriate linear equation (Eq. 8.15).

```
Moessbauer quadrupole splitting parameter (proper coordinate system)
e**2qQ = -0.406 MHz = -0.035 mm/s
eta    = 0.871
Delta-EQ=(1/2e**2qQ*sqrt(1+1/3*eta**2)) = -0.227 MHz = -0.020 mm/s
RHO(0)= 11581.352209571 a.u.**-3 # the electron density at the Fe nucleus.
```

NOTE:

- Following the same procedure, Mössbauer parameters can be computed with the CASSCF wavefunction. In case of a state-averaged CASSCF calculation, the averaged density is used in the subsequent Mössbauer calculation.

El texto anterior es parte del [manual](#) de ORCA-4.1.2.

## 1.2. Archivo de entrada para ORCA.

A continuación se muestra un ejemplo de un archivo de entrada para ORCA con las líneas de código necesarias para generar los parámetros Mossbauer de compuestos que contienen Hierro (Fe).

```
1 ! DFT B3LYP def2-TZVPP TightSCF Grid5 NoFinalGrid SlowConv
2 $pal
3 nprocs 8
4 end
5 $method SpecialGridAtoms 26
6 SpecialGridIntAcc 7
7 end
8 $basis NewGTO 26 "CP(PPP)" end
9 end
10 # [fe(III)(h2o)6]3+ s=5/2
11 * xyz 3 6
12 Fe -0.000411 -0.000246 -0.000775
13 O -0.167444 1.039129 1.718988
14 H 0.366677 1.821113 1.953619
15 O -1.202644 1.288334 -0.980811
16 H -1.651846 2.054751 -0.579763
17 O -1.605608 -1.105693 0.509300
18 H -2.530881 -0.892003 0.288257
19 O 0.168539 -1.041049 -1.719384
20 H 0.865752 -0.926020 -2.391147
21 O 1.201001 -1.287151 0.983345
22 H 1.461288 -1.224386 1.921214
23 O 1.604203 1.105655 -0.512009
24 H 2.529380 0.891372 -0.291077
25 H -1.462647 1.230426 -1.919134
26 H 1.597565 1.901203 -1.075919
27 H -0.863921 0.926491 2.391934
28 H 1.650092 -2.055598 0.586009
29 H -0.365028 -1.822453 -1.957149
30 H -1.598574 -1.900879 1.073614
31 *
32 # Note that the EPR block needs to be below the coordinate block for
33 # the "all Fe" command (all irons) to be recognized.
34 $eprnmr nuclei = all Fe {rho, fgrad}
35 end
```

Figura 1: Archivo de entrada con extensión .inp.

### 1.3. Descripción del archivo de entrada.

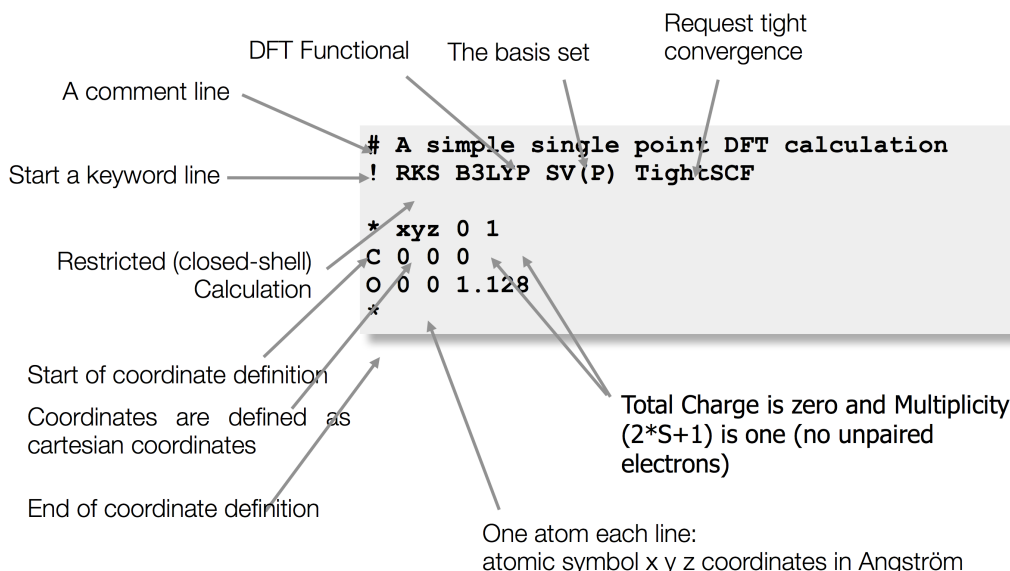


Figura 2: Input general.

En el caso de la figura (1) se tiene lo siguiente:

**Línea 1:** Se inicia con una línea clave indicada con el signo !.

**DFT** (*Density functional theory*) indica el método de modelado mecánico cuántico computacional que usaremos, DFT sirve para investigar la estructura electrónica o nuclear (principalmente el estado fundamental) de sistemas de muchos cuerpos, en particular átomos, moléculas y las fases condensadas.

**B3LYP** es el funcional. **def2-TZVPP** es la base que se está usando en este ejemplo pero comúnmente se usará la base **6-31g**, y **TightSCF** es un método de *convergencia*.

**Líneas 2 a 4:** Las líneas que comienzan con el símbolo % corresponden a bloques de entrada que le indican a ORCA que debe llevar a cabo ciertas especificaciones según el cálculo que se este realizando o la tarea que se este llevando a cabo, para cerrar el bloque que inicia con % se escribe la palabra *end*. En este caso este bloque de código le indica a ORCA que lleve a cabo el cálculo de manera paralela utilizando 8 procesadores de la máquina permitiendo que el cálculo sea más rápido que si no lo hacemos de manera paralela.

**Nota importante:** ORCA corre de manera paralela en la máquina remota usando *openMPI*, esto será importante más adelante cuando se corra el archivo de entrada.

Para conocer cuantos procesadores y núcleos tiene la máquina se puede escribir los siguientes comandos en la terminal:

```
cat /proc/cpuinfo | grep "model name"

cat /proc/cpuinfo | grep "cpu cores"
```

Figura 3: Información sobre los procesadores y núcleos de la máquina.

La primera línea mostrará el número de procesadores así como sus características y la segunda línea mostrará el número de núcleos en cada procesador.

**Líneas 5 a 9:** Corresponden a bloques de código particulares para obtener los parámetros Mossbauer del compuesto, tal como se mostró en la lectura anterior del manual de ORCA.

**Línea 10:** Las líneas que inician con el símbolo # corresponden a comentarios que se pueden hacer en el archivo de entrada y que sirven como guía de lo que se esta haciendo, dichas líneas no afectan en nada la tarea que se este realizando, son ignorados por ORCA. En este caso se esta indicando que el compuesto que se esta usando es  $Fe(III)(H_2O)_6$ , el 3+ indica que la carga total del compuesto es +3, así que no es neutro y  $s = \frac{5}{2}$  es el spin.

**Línea 11 a 31:** Este es el bloque de código que corresponde a las coordenadas **cartesianas** de la molécula. Las letras **xyz** indican que son cartesianas y se observa que inicia y se cierra con el símbolo **\***.

En la **Línea 11** se puede observar que hay dos números a un lado de **xyz**, el primer número corresponde a la **carga eléctrica total** del compuesto, puede ser positiva o negativa, mientras que el segundo número corresponde a la **multiplicidad** y está dada por la ecuación  $m = 2s + 1$ . Anteriormente se mencionó que el compuesto tiene carga positiva +3 y como  $s = \frac{5}{2}$ , si se sustituye  $s$  en la ecuación de la multiplicidad se obtiene que  $m = 6$ . Es importante tener cuidado en estos números sino va a haber problemas al realizar el cálculo.

En este [enlace](#) se puede obtener un archivo con las coordenadas de varios compuestos que contiene Hierro.

El resto de líneas corresponden a comentarios y la indicación de que ORCA debe de reconocer que hay hierro en el compuesto y que debe calcular algunos parámetros que se buscan.

## 1.4. Pasar carpetas o archivos entre máquinas.

El archivo de entrada puede ser generado en nuestra computadora (usando Notepad++ por ejemplo), una vez que esta listo se puede mandar a la computadora remota. A continuación se muestran algunas [líneas de comandos](#) necesarias para enviar y recibir archivos y carpetas de nuestra computadora local a la computadora remota y viceversa

### Pasar carpetas de remoto a local:

```
scp -r -P XXX orca@xxx.xxx.x.xxx:/home/orca/.../carpeta /directorio local
```

### Pasar archivos de remoto a local:

```
scp -P XXX orca@xxx.xxx.x.xxx:/home/orca/.../archivo.ext /directorio local
```

### Pasar carpetas de local a remoto:

```
scp -r -P XXX /directorio local/carpeta orca@xxx.xxx.x.xxx:/home/orca/.../carpeta
```

### Pasar archivos de local a remoto:

```
scp -P XXX /directorio local/archivo.ext orca@xxx.xxx.x.xxx:/home/orca/.../carpeta
```

Donde *scp* (Secure Copy Protocol) es un protocolo de transferencia de archivos en red que permite la transferencia de archivos fácil y segura entre un host remoto y uno local, o entre dos ubicaciones remotas.

*-r* indica que se va a enviar o recibir carpetas, notar que para el caso de archivos no esta presente.

*-P* permite conectarnos por un puerto de acceso *XXX* a la máquina y es el mismo que con el que nos conectamos desde un inicio.

*orca@xxx.xxx.x.xxx:/home/orca/.../carpeta:* indica a qué máquina nos estamos conectando y especificamos la ruta a donde se quiere enviar las carpetas o archivos o bien la ruta de las carpetas o archivos que se quieren traer a nuestra computadora local.

*/directorio local:* es la ruta dentro de nuestra computadora a donde se quiere recibir las carpetas o archivos o bien las carpetas o archivos que se van a mandar.

En cada caso debemos escribir nuestra contraseña para enviar o recibir los archivos.

## 1.5. Correr el archivo de entrada en ORCA.

Una vez que se tiene el archivo de entrada con extensión **.inp**, como el de la figura (1), se manda a la máquina remota en algún directorio, posteriormente se accede a la máquina y se busca el archivo. Una vez que se esta en el directorio donde se encuentra el archivo éste ya se puede correr usando ORCA.

**Nota Importante:** Antes se mencionó que ORCA funciona en paralelo así que para correr el archivo es importante escribir **toda** la ruta en donde se encuentra ORCA, esto es importante para que se reconozca a openMPI.

Para conocer la ruta en donde se encuentra ORCA, en la terminal escribimos *which orca* y aparecerá la ruta de ubicación, dicha ruta se debe escribir en la terminal para correr el archivo.

Ver imagen:

```
omar@DESKTOP-43TV230:~$ ssh orca@132.248.7.196 -p 223
orca@132.248.7.196's password:
Last login: Wed Mar 10 13:19:11 2021 from fixed-187-189-88-171.totalplay.net
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
    LANGUAGE = (unset),
    LC_ALL = (unset),
    LANG = "C.UTF-8"
are supported and installed on your system.
perl: warning: Falling back to the standard locale ("C").
[orca@Tzinkhua ~]$ ls
FeCl4      Mossbauer  openmpi-3.1.4.tar.gz  orca_4_2_1_linux_x86-64_openmpi314.tar.xz
Mossbauer  openmpi-3.1.4  orca_421_314          perl5
[orca@Tzinkhua ~]$ cd Mossbauer
[orca@Tzinkhua Mossbauer]$ ls
fe2cn6 fe3cl4 fe3f6 fe3h12o6 fe4o4 fec14 feco5
[orca@Tzinkhua Mossbauer]$ cd fe3h12o6
[orca@Tzinkhua fe3h12o6]$ ls
fe3h12o6.gbw fe3h12o6.inp fe3h12o6.out fe3h12o6.prop fe3h12o6_property.txt
[orca@Tzinkhua fe3h12o6]$ which orca
~/orca_421_314/orca
[orca@Tzinkhua fe3h12o6]$ /home/orca/orca_421_314/orca fe3h12o6.inp > fe3h12o6.out &
```

Figura 4: Ejemplo en la terminal.

En la imagen, al final, se observa que al escribir *which orca* aparece primero el símbolo (`~`), dicho símbolo indica el home y en este caso es `~/` `/home/orca`, así la ruta de ubicación de *orca* es: `/home/orca/orca_421_314/orca`, dicha ruta se debe escribir en la terminal estando en el directorio donde se encuentra el archivo y delante se escribe el nombre del archivo junto con su extensión, luego se escribe el símbolo `>` y seguido el nombre del archivo más la extensión `.out`, finalmente se escribe el símbolo `&`.

Lo que se está haciendo es decirle a ORCA que va a correr el archivo *fe3h12o6.inp* (que es como nombré al archivo) usando openMPI (correr en paralelo usando 8 procesadores) y de salida (símbolo `>`) se quiere un archivo con extensión `.out`, el nombre de dicho archivo puede ser el que se desee, en este caso fue nombrado de la misma forma, será distinguido por la extensión del archivo. Finalmente se usa el símbolo `&` para poder seguir usando la terminal mientras el cálculo se está llevando a cabo en segundo plano. Se puede escribir en la terminal el comando *top* para ver que procesos se están llevando a cabo en tiempo real en la máquina.

Más adelante veremos que este proceso se puede automatizar y los archivos de entrada sólo los llamaremos *input.inp* y el archivo de salida lo llamaremos *output.out*

Dependiendo la complejidad del compuesto el cálculo puede tardar varios minutos o incluso horas.

Una vez que se concluye el cálculo aparecerán varios archivos, por el momento interesa el archivo con la extensión `.out`. Para ver los resultados será necesario exportar la carpeta completa a nuestra computadora.

## 1.6. Visualizar los resultados.

Una vez que se tienen los archivos en nuestra computadora se pueden visualizar los resultados usando algún software de visualización. En el siguiente [enlace](#) se encuentran algunos de los software de visualización compatibles con ORCA.

En el caso de ChemCraft es posible obtenerlo por tiempo limitado como prueba gratuita o bien Avogadro puede ser una herramienta muy útil y gratuita.

Como ejemplo, usando ChemCraft se puede levantar el archivo con la extensión `.out` y visualizar la molécula. También se puede usar un lector de texto plano como Notepad++ para ver el output de ORCA que corresponde al archivo con extensión `.out`

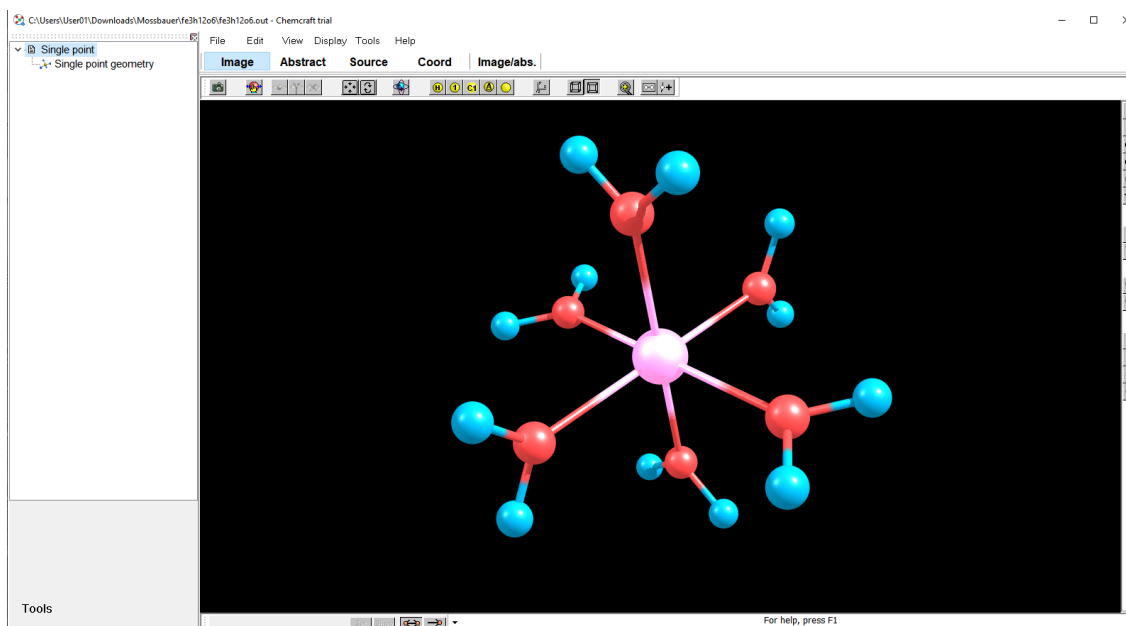


Figura 5: Visualización de la molécula.

Si vamos a *Source* se encontrará toda la información sobre el cálculo hecho por ORCA, entre dicha información se obtienen algunos datos que permitirán obtener los parámetros Mossbauer que se buscan.

```
-----
ELECTRIC AND MAGNETIC HYPERFINE STRUCTURE
-----

Nucleus   0Fe: A:ISTP=   57 I=   0.5 P= 17.2798 MHz/au**3
          Q:ISTP=   57 I=   0.5 Q=  0.1600 barn
-----

Raw EFG matrix (all values in a.u.**-3):
      -0.0127   -0.0128   -0.0069
      -0.0128    0.0407    0.0061
      -0.0069    0.0061   -0.0280

V(E1)      -0.0126   -0.0619    0.0745
V(Nuc)      -0.0012    0.0312   -0.0301
-----
V(Tot)      -0.0137   -0.0307    0.0444
Orientation:
X      0.9098029    0.3460970    0.2290754
Y      0.2506888   -0.0183671   -0.9678935
Z     -0.3307776    0.9380189   -0.1034730

Note: Tensor is right-handed

Mossbauer quadrupole splitting parameter (proper coordinate system)
e**2qQ =   1.669 MHz =   0.144 mm/s
eta     = 0.381
Delta-EQ=(1/2[e**2qQ]*sqrt(1+1/3*eta**2)) =   0.855 MHz =   0.074 mm/s
RHO(0)= 11816.430132276 a.u.**-3
-----

Timings for individual modules:

Sum of individual times      ...    161.555 sec (=   2.693 min)
GTO integral calculation     ...      0.662 sec (=   0.011 min)  0.4 %
SCF iterations               ...    160.447 sec (=   2.674 min) 99.3 %
EPR/NMR calculation          ...      0.446 sec (=   0.007 min)  0.3 %
***ORCA TERMINATED NORMALLY***
TOTAL RUN TIME: 0 days 0 hours 2 minutes 42 seconds 1 msec
```

Figura 6: Archivo de salida.



## 1.7. Obtener los parámetros Mossbauer

Como se explica en la sección 1.1, el espectro Mossbauer puede ser determinado por el *desplazamiento isomérico* (isomer shift,  $\delta$ ) y el *desdoblamiento cuadrupolar* (quadrupole splitting,  $\Delta E_Q$ ) relativo a un material estándar, usualmente el hierro, esto para el caso en el que no existe interacción entre el núcleo y cualquier campo magnético circundante sino se tendría un parámetro más conocido como *desdoblamiento magnético* (magnetic splitting,  $\Delta B_Q$ ). Para conocer acerca de la teoría Mossbauer es recomendable esta [página](#).

La información obtenida sobre los parámetros Mossbauer al correr el archivo en ORCA es la siguiente:

```
Moessbauer quadrupole splitting parameter (proper coordinate system)
e**2qQ = 1.669 MHz = 0.144 mm/s
eta = 0.381
Delta-EQ=(1/2{e**2qQ}*sqrt(1+1/3*eta**2)) = 0.855 MHz = 0.074 mm/s
RHO(0)= 11816.430132276 a.u.**-3
```

Figura 7: Información Mossbauer

Como se puede observar, sí se obtiene el valor del desdoblamiento cuadrupolar  $\Delta E_Q$  pero no el valor del desplazamiento isomérico  $\delta$ , sin embargo se da el valor de la densidad eléctrica en el núcleo  $\rho_0$ , dicho valor puede ser sustituido en la ecuación del desplazamiento isomérico  $\delta = \alpha(\rho_0 - C) + \beta$  pero, ¿cómo se obtienen los valores de  $\alpha$ ,  $C$  y  $\beta$ ? Como se mencionó antes,  $\alpha$ ,  $C$  y  $\beta$  usualmente son determinados con una regresión lineal entre los valores obtenidos experimentalmente de  $\delta$  frente a la densidad electrónica calculada teóricamente para un serie de compuestos de hierro con varios estados de oxidación y espín, sin embargo, en este caso se está considerando este [documento](#) que contiene los valores  $\alpha$ ,  $C$  y  $\beta$  ya calibrados usando varios métodos (funcionales y bases). En dicho documento se puede encontrar la siguiente tabla:

### Linear Fit data for the prediction of $^{57}\text{Fe}$ Mössbauer isomer shift for TPSS-optimized geometries that are not given in table 1:

The parameters are given according to the linear equation  $\delta = \alpha (\rho - C) + \beta$

functional	basis set	$\alpha$	$\beta$	$C$	$R^2$	std. dev. [mm/s]
BP86	def2-TZVPP* <sup>3</sup>	-0.420	2.958	11810	0.945	0.11
TPSS	def2-TZVPP* <sup>3</sup>	-0.414	4.650	11810	0.956	0.10
B3LYP	def2-TZVPP* <sup>3</sup>	-0.362	2.830	11810	0.974	0.09
B2PLYP	def2-TZVPP* <sup>3</sup>	-0.327	0.972	11810	0.972	0.09

\* CP(PPP) basis set for the iron atom and def2-TZVPP basis set for the ligand atoms.

Figura 8: Valores calibrados

La tabla corresponde a este caso pues se está usando la base **B3LYP**, el funcional **def2-TZVPP** y **CP(PPP)**, así que los valores que se requieren son:  $\alpha = -0.362$ ,  $\beta = 2.830$ ,  $C = 11810$  y el valor obtenido por ORCA de  $\rho_0 = 11816.430132276 a.u.^3$ , sustituyendo en la relación lineal  $\delta = \alpha(\rho_0 - C) + \beta$  se obtiene que  $\delta = 0.502292116 \frac{mm}{s}$ . Se puede notar que si sólo se consideran dos decimales de  $\rho_0$ , es decir,  $\rho_0 = 11816.43 a.u.^3$ , se va a obtener que  $\delta = 0.51 \frac{mm}{s}$ , es éste el resultado que se encuentra en el documento de referencia. Importante ubicar el método que usamos.



### B3LYP / def2-TZVPP / CP(PPP)

Compound	$\rho(0)$ [a.u. <sup>-3</sup> ] (calculated)	isomer shift [mm/s] <sup>1,2</sup>
fe2cl4	11815.3	0.9
fe2cn6	11817.74	-0.02
fe2f6	11814.2	1.34
fe2poroac	11815.35	1.05
fe3az	11817.06	0.29
fe3cl4	11816.86	0.19
fe3cl6	11816.31	--
fe3cn6	11817.99	-0.13
fe3f6	11816.3	0.48
fe3h2o	11816.43	0.51
fe3mac	11817.24	0.15
fe3oeppy	11817.26	0.2
fe3poro2	11815.92	0.67

Figura 9: Valores obtenidos

Ahora ya se tienen el *desplazamiento isomérico*,  $\delta = 0.502292116 \frac{mm}{s}$  y el *desdoblamiento cuadrupolar*,  $\Delta E_Q = 0.074 \frac{mm}{s}$ , entonces ya se puede obtener el espectro Mossbauer correspondiente al compuesto  $Fe(III)(H_2O)_6$ .

## 1.8. Visualizar el espectro Mossbauer

Para visualizar el espectro Mossbauer del compuesto  $Fe(III)(H_2O)_6$  se realizó un programa en Python, dicho programa lo mostraremos más adelante pero por ahora al introducir el valor de los parámetros Mossbauer el resultado obtenido es el siguiente espectro Mossbauer.

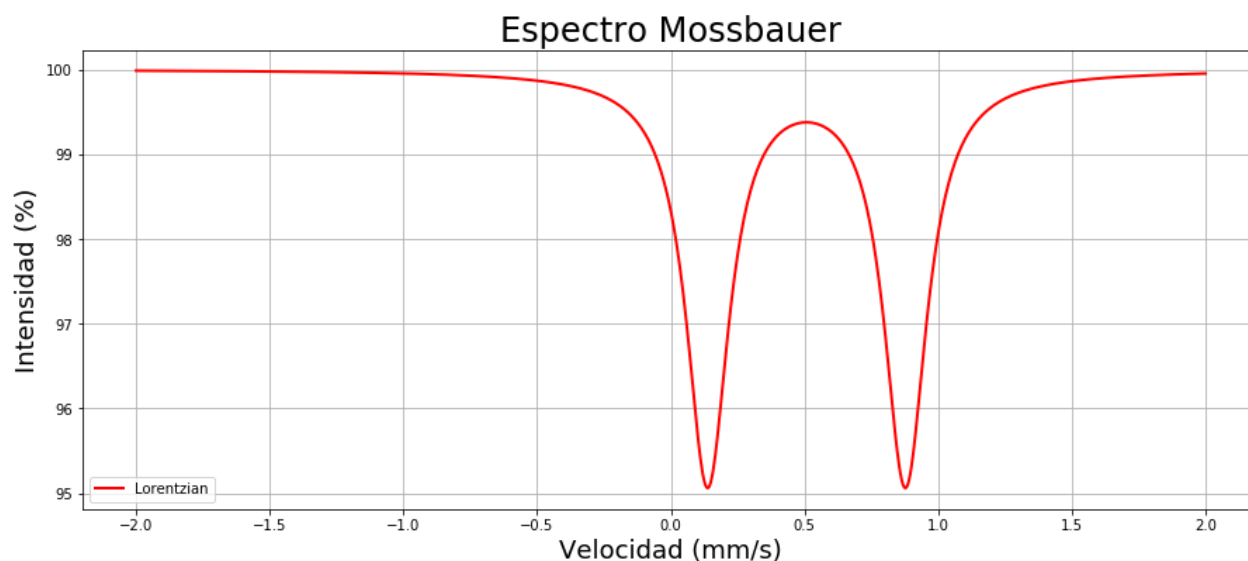


Figura 10: Espectro Mossbauer del compuesto  $Fe(III)(H_2O)_6$ .

## 1.9. Automatización del proceso

Debido a que se pretende desarrollar un sistema de aprendizaje artificial, se requiere de suficientes datos que sirvan como datos de entrenamiento para nuestro sistema. Así que es necesario hacer todo este proceso para cientos o quizá miles de compuestos distintos, es por ello que se realizó un script en Linux que automatice todo el proceso

El trabajo automatizado consiste en lo siguiente: Una vez que los archivos de entrada son generados en nuestra maquina local son almacenados en carpetas distintas y se trasladan a la máquina remota. Estando en la máquina remota se hacen los cálculos con ORCA para cada uno de los compuestos. Una vez hechos los cálculos se extrae la información necesaria que permita calcular los parámetros Mossbauer de cada compuesto. El script extrae las líneas de texto dentro de los archivos de salida generados por ORCA que contienen los parámetros que necesitamos, una vez extraídos divide los datos correspondientes a la densidad de electrones  $\rho_0$  y al desdoblamiento cuadrupolar  $\Delta E_Q$  en dos nuevos archivos. Los valores numéricos de dichos parámetros son transferidos a nuestra máquina local para ser leídos por nuestro programa en Python y así finalmente obtener los parámetros y espectros Mossbauer de cada compuesto que se utilice

El script en Linux realizado es el siguiente:

```
#!/bin/bash

ruta=/home/orca/Mossbauer/compuestos
cd $ruta

for directorio in $(ls)
do
    cd $ruta/$directorio
    /home/orca/orca_421_314/orca input.inp > output.out &
    sleep 10m
done

cd $ruta
mkdir Datos
touch datos.txt

for directorio in $(ls)
do
    mv datos.txt $ruta/$directorio
    cd $ruta/$directorio
    grep -w "RHO(0)" output.out >> datos.txt
    grep -w "Delta-EQ" output.out >> datos.txt
done

cd $ruta
grep -w "RHO(0)" datos.txt > densidad.txt
grep -w "Delta-EQ" datos.txt > desdoblamiento.txt

cut -c 11-25 densidad.txt > datos01.txt
cut -c 62-68 desdoblamiento.txt > datos02.txt

mv datos.txt $ruta/Datos
mv densidad.txt $ruta/Datos
mv desdoblamiento.txt $ruta/Datos
mv datos01.txt $ruta/Datos
mv datos02.txt $ruta/Datos

cd $ruta

for directorio in $(ls)
do
    cd $ruta/$directorio
    rm input.gbw
    rm input.prop
    rm input_property.txt
    rm output.out
done
-- INSERT --
```

Figura 11: Script en Linux que automatiza el proceso.

Este script genera una carpeta llamada 'Datos' que contiene los archivos de texto plano donde están los valores de la densidad de electrones  $\rho_0$  y del desdoblamiento cuadrupolar  $\Delta E_Q$  de cada compuesto

El programa realizado en Python levanta estos archivos de datos para generar un DataFrame.

```
In [18]: datos01 = np.loadtxt('datos01.txt')
datos01

Out[18]: array([11815.17021058, 11817.63338393, 11814.33790331, 11816.94425591,
11816.80584327, 11816.24722446, 11817.8802076 , 11816.3315988 ,
11816.30306087, 11820.0618833 , 11818.08932221, 11818.08932221,
11816.85215258, 11816.49872384, 11816.3604875 ])
```

```
In [19]: datos02 = np.loadtxt('datos02.txt')
datos02

Out[19]: array([-3.588e+00,  3.000e-03,  3.898e+00, -1.966e+00, -1.100e-02,
-0.000e+00,  2.212e+00, -0.000e+00,  7.600e-02, -3.100e-02,
 2.537e+00,  2.537e+00,  6.290e-01,  7.640e-01, -6.440e-01])
```

```
In [20]: Datos = {'Desdoblamiento_Cuadrupolar': datos02,
'Densidad_Electronica': datos01}
parametros = pd.DataFrame(Datos)
```

```
In [21]: parametros

Out[21]:
```

	Desdoblamiento_Cuadrupolar	Densidad_Electronica
0	-3.588	11815.170211
1	0.003	11817.633384
2	3.898	11814.337903
3	-1.966	11816.944256
4	-0.011	11816.805843
5	-0.000	11816.247224
6	2.212	11817.880208
7	-0.000	11816.331599
8	0.076	11816.303061
9	-0.031	11820.061883
10	2.537	11818.089322
11	2.537	11818.089322
12	0.629	11816.852153
13	0.764	11816.498724
14	-0.644	11816.360487

Figura 12: Datos generados por ORCA para calcular los parámetros Mossbauer.

Ya con los datos en el programa se pueden obtener los valores de los parámetros Mossbauer así como los correspondientes espectros de cada uno de los compuestos usados.

```
# Función que genera el ajuste Loretziano.
def Ajuste(w, x, IS, QS, I):
    denominador_1 = (x - IS - QS / 2) ** 2 + (w / 2) ** 2
    denominador_2 = (x - IS + QS / 2) ** 2 + (w / 2) ** 2
    Lorentziana = 100 - (I * w / 2 * np.pi) * ((1 / denominador_1) + (1 / denominador_2))
    return Lorentziana

# Definimos Los parámetros necesarios.
#rango = len(parametros.index)      # Número de datos en el dataset.
rango = len(parametros)

x = np.linspace(-4, 4, 1000)      # Rango de valores en el eje horizontal.
w = 0.194                          # Ancho natural de la línea Mossbauer.
I = 0.15                          # Parámetro de ajuste de la intensidad.

alpha = -0.3612                    # Valores para la obtención del desplazamiento isomérico.
beta = 2.830
C = 11810

# Creamos dos listas con Los valores de Los parámetros Mossbauer de cada compuesto
desplazamiento = []
desdoblamiento = []

#Inicio de un ciclo para cada compuesto del archivo:
for i in range(rango):

    # Valor de la densidad electrónica dada en el output de ORCA.
    rho = parametros.Densidad_Electronica[i]
    IS = alpha * (rho - C) + beta    # Valor del desplazamiento isomérico (Isomer Shift)

    # Valor del desdoblamiento cuadrupolar (Quadrupole Splitting) dado en el output de ORCA.
    QS = parametros.Desdoblamiento_Cuadrupolar[i]

    # Obtención del ajuste dado Los parámetros:
    Fit = Ajuste(w, x, IS, QS, I)

    # Almacenamos Los parámetros Mossbauer de cada compuesto.
    desplazamiento.append(IS)
    desdoblamiento.append(QS)

    # Valores de Los parámetros Mossbauer:
    print("\n")
    #print("Compuesto: ", parametros.Compuesto[i])
    print("El valor del desplazamiento isomérico es: ", IS, "mm/s")
    print("El valor del desdoblamiento cuadrupolar es: ", QS, "mm/s")

    # Realización del gráfico:
    plt.figure(figsize = (12, 3))
    plt.plot(x, Fit, color = 'green', lw = 1) #label = parametros.Compuesto[i]
    plt.grid(True)
    plt.title('Espectro Mossbauer', size = 16)
    plt.xlabel('Velocidad (mm/s)', size = 10)
    plt.ylabel('Intensidad (%)', size = 10)
    #plt.legend()
```

Figura 13: Programa en Python para obtener los espectros Mossbauer.