

**Guión de las prácticas de
simulaciones cuánticas en sólidos del
Máster de Nuevos Materiales**

Edu Ogando

9 de mayo de 2023

Índice

1. Preliminares	1
1.1. En caso de clases online	1
1.2. Preliminares	1
2. Compilar	2
3. Lanzar trabajos al sistema de colas	2
3.1. Sistema de colas durante la clase	3
4. Simulaciones de Si	3
4.1. Cálculo simple	3
4.1.1. Estructura de bandas	6
4.1.2. Densidad de estados	8
4.2. Análisis de convergencia	10
4.2.1. Convergencia en el espacio recíproco	11
4.3. Convergencia en el espacio real	12
4.4. Convergencia de la base de orbitales	13
5. Simulaciones de Al	14
5.1. Cálculos de análisis de convergencia	14
5.2. Estructura de bandas y DOS	15
6. Simulaciones de Fe	21
6.1. Cálculos de polarización de spin	21
6.2. DOS	22

1. Preliminares

Vamos a trabajar en el cluster de computación de la UPV/EHU. Para ello hay que entrar en la red de la UPV/EHU a través de VPN-anyconnect con nuestras credenciales de LDAP de la UPV/EHU.

Una vez dentro de la UPV/EHU nos conectaremos al cluster de la UPV/EHU con el cliente de terminal `putty.exe` o con el de `X2Go`, este último nos permitirá abrir ventanas gráficas. Para transferir ficheros usaremos `Filezilla`. Esto ya deberá de estar instalado en los ordenadores del alumnado.

1.1. En caso de clases online

Para poder teletrabajar y que el profesor realice un seguimiento de lo que está ejecutando el alumnado, se trabajará con `tmux`. El alumnado ejecutará en la terminal:

```
mitmux.sh
```

que es un script que usa por debajo `tmux` y abrirá una terminal virtual. El profesor entonces podrá conectarse a esta terminal virtual y compartirlo con la alumna o alumno.

1.2. Preliminares

Subimos el material al cluster con `Filezilla`. Lo descomprimos ejecutando el comando:

```
gunzip xvzf siestaFile.tgz
```

El disco `/home` es muy lento, no está pensado para calcular sino es que un disco de alta capacidad. Moveremos nuestra carpeta de trabajo al disco local `/scratch`.

```
cp -r SIESTA /scratch/ikasleXX/  
cd /scratch/ikasleXX/SIESTA  
ls
```

2. Compilar

En este ejercicio hacemos una pequeña compilación para entender el proceso de creación de ejecutables a partir del código fuente. Es muy común en computación que se proporcionen códigos fuente para que estos puedan ser compilados y optimizados para el servidor de cálculo. Además, de este modo podemos mejorarlo o ajustarlo a nuestras necesidades.

Compilar es transformar programa de texto en un ejecutable. Vamos a la carpeta `COMPILAR`. Vemos el programa `hola.f90` en Fortran, lo compilamos y lo ejecutamos

```
cat hola.f90  
gfortran hola.f90 -o hola  
./hola
```

3. Lanzar trabajos al sistema de colas

Para ver el software instalado en Kalk2020 ejecutamos

```
module av  
module av |grep -i siesta  
module av siesta
```

Para lanzar un trabajo al sistema de colas Slurm podemos usar el simple script `siesta.slurm` que encontramos en la carpeta `SLRUM`. Lo copiamos al directorio de trabajo y ejecutamos:

```
sbatch siesta.slurm
```

Podemos ver el estado de los trabajos y eliminar nuestro trabajo tras identificar el `job_id` en el sistema de colas con los comandos:

```
squeue  
squeue -u $USER  
ni  
scancel job_id
```

3.1. Sistema de colas durante la clase

Para calcular en interactivo durante la clase usaremos una reserva que tenemos realizada y el comando

```
interactive -N 1 -n 1 -c 4 -t 6:00:00 -m 10 -r nmmaster01
```

Con este comando se nos abre la terminal en un nodo de cálculo y se nos asignan 4 cores y 10 GB de RAM durante 6 horas de la reserva `nmmaster01` (`nmmaster02` el segundo día). Así podemos trabajar sin interferir con otros usuarios.

4. Simulaciones de Si

4.1. Cálculo simple

Entramos en la carpeta `Si`, creamos la subcarpeta `RUN` y copiamos allí los ficheros para mantener intacta las fuentes.

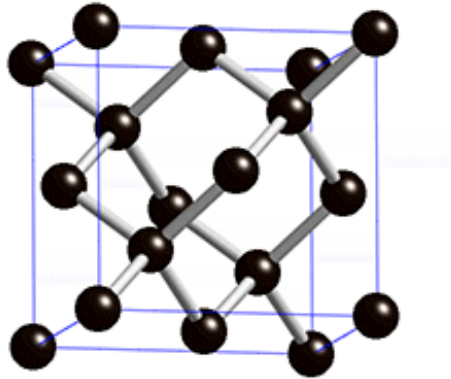


Figura 1: Celda unidad del Si.

```
cd Si
mkdir RUN
cp * RUN/
cd RUN
```

Vamos a calcular la estructura de diamante del Si. En la figura 1 se muestra la estructura atómica del Si con 8 átomos en la celda convencional. Vemos que hay 2 ficheros, en el fichero del pseudopotencial `Si.psf` y el input del sistema `Si.fdf`. Con este fichero de input vamos a:

1. Calcular el parámetro de red del Si
2. Dibujar la estructura de bandas
3. Dibujar la Densidad de estados

Examinaremos el fichero primero y luego ejecutaremos SIESTA.

```
gedit Si.fdf
module load Siesta/4.1.5-intel-2020a
mpirun -np 4 siesta < Si.fdf > Si.out
```

Puede dar el error:
srun: error: nh007: task 0: Floating point exception
srun: Terminating job step 96232.5
[mpiexec@nh007] wait_proxies_to_terminate
que se soluciona con:
export LMPLHYDRA_TOPOLIB=ipl

El fichero fdf-*.log contiene todas las variables de SIESTA y sus valores:

```
wc -l fdf-*.log
more fdf-*.log
```

Se generan varios ficheros. El fichero Si.out tiene la información estándar de la ejecución. Examinemos que se ve en el fichero Si.out:

- Los parámetros iniciales.
- Ciclos de convergencia.
- Resultados finales (Energía, fuerzas, autovalores,...).

Unos comandos de interés

```
tail Si.out
cat MESSAGES
grep "outcell:␣Cell␣ve" Si.out
grep "outcell:␣Cell␣ve" Si.out |awk '{print␣NR,$7}'
grep "Cell␣ve" Si.out |awk '{print␣NR,$7}'|xmgrace -pipe
```


4.1.1. Estructura de bandas

Vamos a dibujar la estructura de bandas. Para ellos debemos crear primero el fichero de datos `Si.bands.dat` para el programa de dibujo.

```
$ ./gnubands -F < Si.bands > Si.bands.dat
$ module load gnuplot/5.4.1-GCCcore-9.3.0
$ gnuplot
```

La opción `-F` de `gnubands` nos sitúa el nivel de Fermi en 0. Al ejecutar `gnuplot` entramos en el programa de dibujo y nos aparece en terminal el prompt de `gnuplot` para introducir los comandos de dibujo. La figura 2 se obtiene si ejecutamos dentro de `gnuplot`:

```
gnuplot> p 'Si.bands.dat' u 1:2 w l
gnuplot> set yrange [-15:5]
gnuplot> p 'Si.bands.dat' u 1:2 w l
gnuplot> quit
```

Tanto para la DOS como para la estructura de bandas siempre es importante dar la referencia del nivel de Fermi y/o dibujar respecto a este ($E_F = 0$). El nivel de Fermi se encuentra en:

```
grep Fermi Si.out
```

o en el fichero `Si.bands.dat`. Donde están situados los puntos de simetría en el dibujo lo podemos encontrar al final del fichero `Si.bands`.

```
more Si.bands.dat
tail Si.bands
```

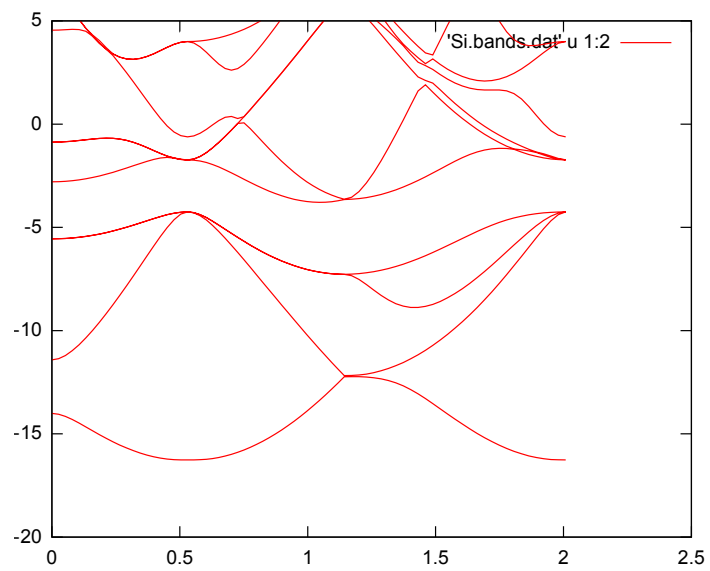


Figura 2: Estructura de bandas del Si. Figura tal y como sale de gnuplot.

Podemos escribir un programa complejo para `gnuplot` y ejecutarlo. En este caso nos crea un fichero `svg`.¹

```
more bands.gnu
gnuplot bands.gnu
```

4.1.2. Densidad de estados

Vamos a dibujar la densidad de estados. Para ello ya hemos introducido los parámetros necesarios en nuestro fichero de entrada y se nos han creado los ficheros `Si.DOS` y `Si.PDOS`. Pero debemos de tratarlos con el programa `fmpdos` para conseguirlos en el formato adecuado para dibujarlos. Vamos a dibujar la DOS de los estados *s*, *p* y *d* para lo que tenemos que ejecutar 3 veces `fmpdos` una por cada tipo de orbital. Para los orbitales *s* ejecutamos `fmpdos` y respondemos:

```
\textbf{./fmpdos}
  Input file name (PDOS):
Si.PDOS
  Output file name :
Si.s.PDOS
  Extract data for atom index (enter atom NUMBER,
  or 0 to select all), or for all atoms of
  given species (enter its chemical LABEL):
0
  Extract data for n= ... (0 for all n ):
3
  Extract data for l= ... (-1 for all l ):
0
  Extract data for m= ... (9 for all m ):
9
```

¹Podemos eliminar las líneas iniciales para que nos muestre el gráfico en la pantalla, la salida por defecto, y poner un `pause 10` al final para que se detenga 10 s. antes de salir.

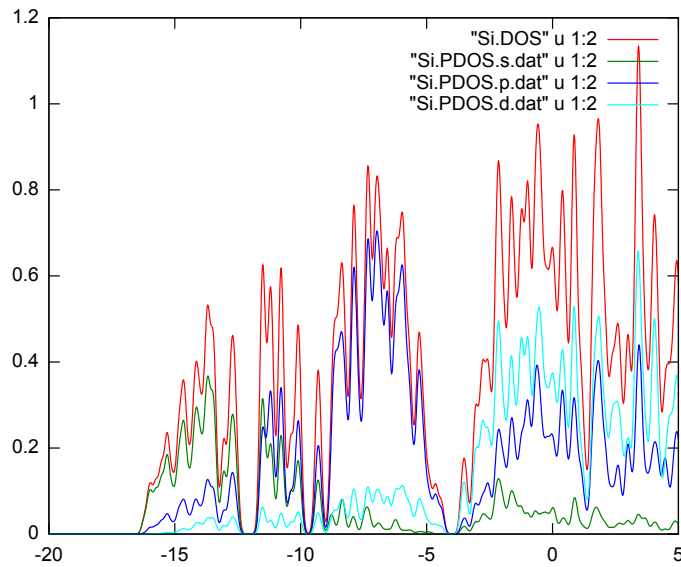


Figura 3: DOS del Si sin suficientes puntos k_{MP} . Figura tal y como la muestra gnuplot.

Y nos ha generado el fichero Si.PDOS.s.dat. Ídem para los demás seleccionando otro número l . Ahora procedemos a dibujarlo, figura 3 ejecutamos *gnuplot* y dentro de la consola de gnuplot:

```
gnuplot> p "Si.DOS" u 1:2 w l
gnuplot> rep "Si.s.PDOS" u 1:2 w l
gnuplot> rep "Si.p.PDOS" u 1:2 w l
gnuplot> rep "Si.d.PDOS" u 1:2 w l
gnuplot> q
```

La figura 3 ha sido la DOS con los puntos k_{MP} (os referimos al parámetro para Monkhorst Pack) con los que hemos calculado, puede haber sido suficiente para obtener un buen resultado del parámetro de red, pero en cualquier caso no es lo suficientemente buena para el gráfico de la DOS, pues se pueden apreciar los picos de la convolución con la gaussiana.

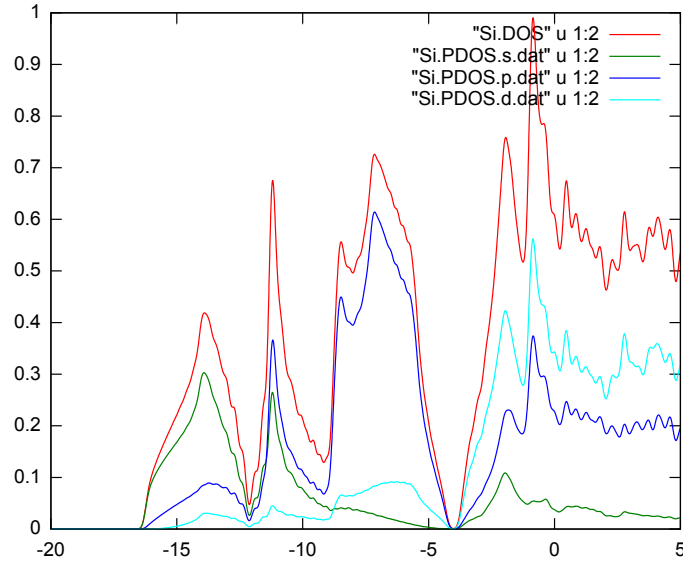


Figura 4: Salida de gnuplot para la DOS del Si con suficientes puntos k_{MP} .

na. Editamos el fichero `Si.fdf` y descomentamos todas las líneas del bloque `PDOS.kgrid.Monkhorst-Pack` y volvemos a lanzar el cálculo, reevaluamos los datos con `fmpdos` y dibujamos nuevamente la DOS con gnuplot. Ahora la curva de la figura 4 no tiene los picos.

4.2. Análisis de convergencia

El análisis de convergencia consiste en comprobar que los parámetros que hemos escogido para la simulación son suficientemente precisos para describir el fenómeno que queremos estudiar. Es decir, ver cuales son los valores de los parámetros más pequeños que nos dan la precisión más pequeña.

A continuación trataremos de buscar cuantos puntos k (malla de puntos en el espacio recíproco) y tamaño de malla en el espacio real nos dan un resultado satisfactorio. El problema de escoger valores arbitrariamente grandes es

que los cálculos se hacen también arbitrariamente grandes en tiempo de ejecución y memoria RAM, lo que los vuelve inviables. Es necesario encontrar un compromiso.

Lo ideal es que todos los parámetros estén congelados salvo el que estemos estudiando. Como nos estamos centrando la propiedad física del parámetro de red, en los ejemplos a continuación esté podrá variar y veremos su valor en base a diferentes parámetros.

Si nos interesase por ejemplo la energía, podría ser conveniente fijar la red, para que sus variaciones no nos influyan en nuestro estudio de la convergencia respecto a por ejemplo el número de puntos k , i.e., que la única fuente de variación sea la debida a los puntos k .

Si tras realizar un análisis de convergencia llegamos a la conclusión de que un parámetro x necesita un cierto valor mínimo, pero otro análisis de convergencia anterior de un parámetro y lo hicimos con un valor de x muy diferente, puede ser necesario comprobar nuevamente la convergencia de y con un valor de x más realista.

4.2.1. Convergencia en el espacio recíproco

Vamos a realizar dos cálculos incrementando el número de puntos k_{MP} , la densidad de puntos en el espacio recíproco, para comprobar que hemos convergido el resultado respecto al número de puntos k_{MP} . Para ello vamos a usar el fichero plantilla `Si2.fdf-tp1`, donde si lo editamos veremos que el número de puntos k_{MP} está definido en una variable, que también aparece en la línea del fichero de input con el prefijo del nombre de los ficheros del cálculo, para que los ficheros de cada valor de k_{MP} se almacenen con prefijos y nombres diferentes. Sustituiremos esa variable por diferentes valores con las órdenes del bash:

```
for i in 3 6 9 12;
do echo $i; sed "s/tp1k/$i/g" Si2.fdf-tp1 > Si2.fdf;
mpirun -n 4 siesta < Si2.fdf > Si2-k$i.out; done
```

Analizamos los resultados. La energía del sistema corresponde a diferentes configuraciones, diferentes parámetros de celda.

```
for i in 3 6 9 12;do grep "Total_=" Si2-k$i.out;done
```

Como se observa con 6 puntos k_{MP} ya está convergido. Si analizamos el parámetro de red:

```
for i in 3 6 9 12;do grep -H "Cell_ave" Si2-k$i.out|tail -1;done
```

Para cálculos de moléculas se usa un único punto k . No hay interacciones con otras celdas.

4.3. Convergencia en el espacio real

El mallado en el espacio real tiene especial importancia a la hora del cálculo del potencial eléctrico y el posicionamiento de los orbitales en este mallado. Viene definido por el parámetro `Mesh.Cutoff`. Por defecto usa 300 Ry.

- Descomentamos la línea.
- Le daremos de valor la variable `MESH`.
- También lo añadiremos en `SystemLabel` al inicio del fichero, `Si-mMESH`.
- Realizamos cálculos para 100, 200, 300 y 400 Ry.

```
for i in 100 200 300 400;
do echo $i ;
sed "s/MESH/$i/g" Si.fdf > Si2.fdf;
mpirun -n 4 siesta < Si2.fdf > Si-m$i.out;
done
```

y analizamos los resultados como antes

```
for i in 100 200 300 400;
do grep "ll:_Cell_ave" Si-m$i.out|tail -1;
done
for i in 100 200 300 400;do grep "Total_" Si-m$i.out;done
```

4.4. Convergencia de la base de orbitales

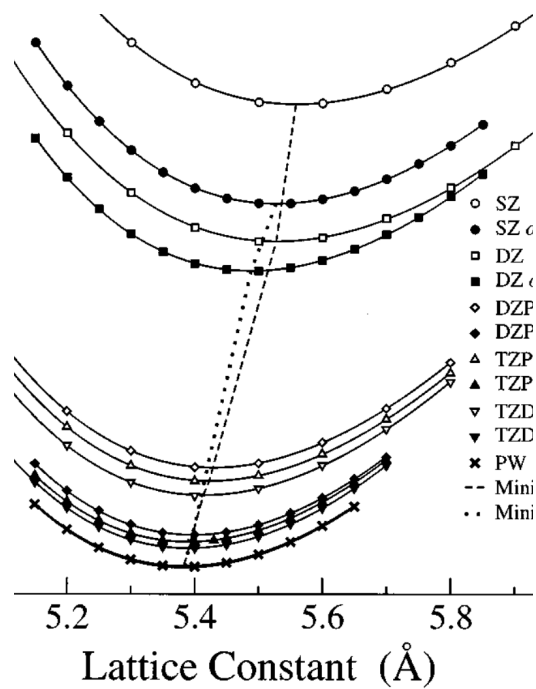


Figura 5: Convergencia del parámetro de red del Si en función de la base.

5. Simulaciones de Al

5.1. Cálculos de análisis de convergencia

Entramos en la carpeta Al, creamos la subcarpeta RUN y copiamos allí los ficheros para mantener intacta las fuentes.

```
cd Al
mkdir RUN
cp * RUN/
cd RUN
```

Vamos a calcular la estructura de FCC del Al. Vemos que hay 2 ficheros, en el fichero del pseudopotencial Al.psf y el input del sistema Al.fdf. Con este fichero de input vamos a:

1. Calcular el parámetro de red del Al
2. Dibujar la estructura de bandas
3. Dibujar la Densidad de estados

Con la técnica de usar una plantilla para generar los input variando una variable calcularemos la convergencia en k_{MP} . Examinaremos el fichero primero y luego ejecutaremos SIESTA para varios k_{MP} .

```
gedit Al.fdf-tpl
for i in 4 8 12 16;do echo $i ;
  sed "s/KPOINT/$i/g" Al2.fdf-tpl > Al2.fdf;
  mpirun -n 4 siesta < Al2.fdf > Al-k$i.out;
done
```

Se generan varios ficheros. Los ficheros Al-kXX.out tiene la información estándar de la ejecución para los diferentes k_{MP} . En este caso tenemos que

fijarnos en la energía libre pues hemos introducido una temperatura finita al sistema. Si miramos la convergencia en la energía libre y el parámetro de red

```
for i in 4 8 12 16;do grep "FreeEng" Al-k$i.out|tail -1; done
for i in 4 8 12 16; do head -1 Al-k$i.STRUCT_OUT; done
```

observamos que ahora debemos recurrir a mayor densidad de k_{MP} que en el caso del Si.

Ahora pasamos a ver como converge respecto a la densidad en la malla del espacio real. Para ello editamos `Al.fdf-tp1` y fijaremos $k_{MP} = 12$ y usaremos la variable `MESH` para el valor de `Mesh.Cutoff` y lanzaremos otra serie de cálculos y analizamos los resultados. Podemos usar como nuevo parámetro de red 4.20 \AA , que aproximadamente nos ha dado en el cálculo anterior.

```
for i in 100 200 300 400;
do echo $i ;
sed "s/MESH/$i/g" Al.fdf-tp1 > Al2.fdf;
mpirun -n 4 siesta < Al2.fdf > Al-m$i.out;
done
for i in 100 200 300 400; do grep "FreeEng" Al-m$i.out|tail -1; done
for i in 100 200 300 400; do head -1 Al-m$i.STRUCT_OUT; done
```

5.2. Estructura de bandas y DOS

Vamos a dibujar la estructura de bandas y la DOS. Para ello, vamos al fichero `Al.fdf` y sustituimos los valores correctos que hemos obtenido para el número de puntos k_{MP} , 12, la densidad de la malla, 200 R y el parámetro de red, 4.218 \AA . No vamos a realizar una simulación de DM pues ya hemos realizado este cálculo en los apartados anteriores. En este fichero está incluida la información para obtener la estructura de bandas y la DOS y el calculo será sin DM.

Realizamos el cálculo y generamos el fichero para poder dibujar las bandas a partir de los resultados

```
mpirun -n 4 siesta < Al.fdf > Al.out
./gnubands < Al.bands > Al.bands.dat
gnuplot
```

Al ejecutar `gnuplot` entramos en el programa de dibujo y nos aparece en terminal el prompt de `gnuplot` para introducir los comandos de dibujo. Dentro del programa de dibujo `gnuplot` ejecutamos:

```
gnuplot> p 'Al.bands.dat' u 1:2 w l
gnuplot> set yrange [-15:10]
gnuplot> p 'Al.bands.dat' u 1:2 w l
gnuplot> quit
```

Vemos que se parece a la estructura de bandas de un gas de electrones libres, figura 7.

El nivel de Fermi se encuentra en:

```
siesta:          Fermi =      -4.014
```

Para la DOS usamos la utilidad `fmpdos` para extraer la DOS de los diferentes orbitales (todos, *s*, *p* y *d*) en el formato adecuado para dibujarlos. Para lo que tenemos que ejecutar 4 veces `fmpdos` una por cada tipo de orbital. Para dibujar la DOS total:

```
\textbf{./fmpdos}
  Input file name (PDOS):
Al.PDOS
  Output file name :
Al.all.PDOS
```

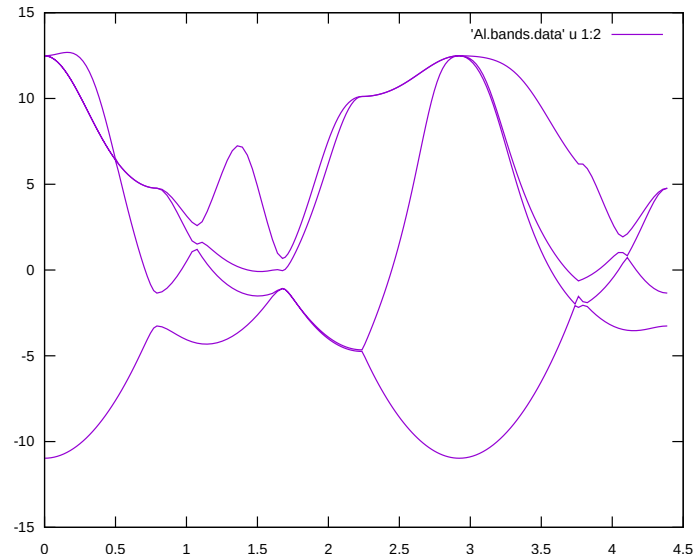


Figura 6: Estructura de bandas del Al, tal y como la muestra gnuplot.

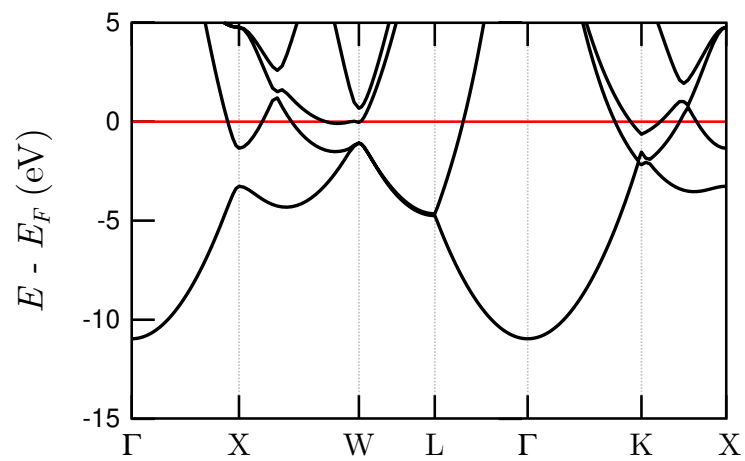


Figura 7: Estructura de bandas del Al editada.

```

Extract data for atom index (enter atom NUMBER,
or 0 to select all), or for all atoms of
given species (enter its chemical LABEL):
0
Extract data for n= ... (0 for all n ):
0

```

Para los orbitales s ejecutamos `fmpdos` y respondemos:

```

\textbf{./fmpdos}
Input file name (PDOS):
Al.PDOS
Output file name :
Al.s.PDOS
Extract data for atom index (enter atom NUMBER,
or 0 to select all), or for all atoms of
given species (enter its chemical LABEL):
0
Extract data for n= ... (0 for all n ):
3
Extract data for l= ... (-1 for all l ):
0
Extract data for m= ... (9 for all m ):
9

```

Y nos han generado los ficheros `Al.all.PDOS` y `Al.s.PDOS`. Ídem para los demás seleccionando otro número l . Ahora procedemos a dibujar con ejecutamos `gnuplot` la DOS total de este cálculo en el que hemos usado un mallado fino explícito de $k_{MP} = 30$, el del cálculo con $k_{MP} = 12$ y el del cálculo con $k_{MP}=4$. Dentro de la consola de `gnuplot`:

```

gnuplot> p "Al.all.DOS" u 1:2 w l
gnuplot> rep "Al-k12.DOS" u 1:2 w l
gnuplot> rep "Al-k4.DOS" u 1:2 w l

```

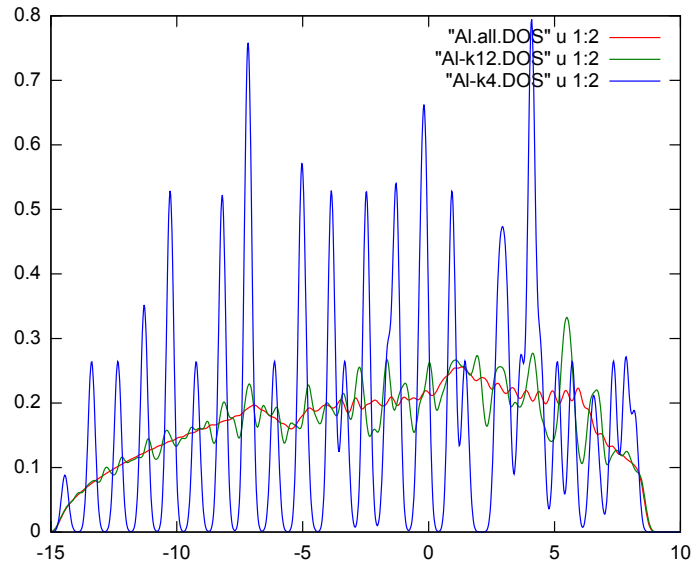


Figura 8: DOS del Al, tal y como la muestra gnuplot. Se comparan los DOS con $k_{MP} = 30, 12$ y 4 .

```
gnuplot> q
```

De la comparación de la figura 8 vemos que 4 puntos k_{MP} son insuficientes para poder definir correctamente el nivel de Fermi y sus propiedades. Pasemos a dibujar la descomposición de la DOS.

```
gnuplot> p "Al.all.DOS" u 1:2 w l
gnuplot> rep "Al.s.PDOS" u 1:2 w l
gnuplot> rep "Al.p.PDOS" u 1:2 w l
gnuplot> rep "Al.d.PDOS" u 1:2 w l
gnuplot> q
```

La figura 9 se observa que en la descomposición no aparecen estados tipo d , como es lógico pues hemos escogido una base SZ que no tiene estados d .

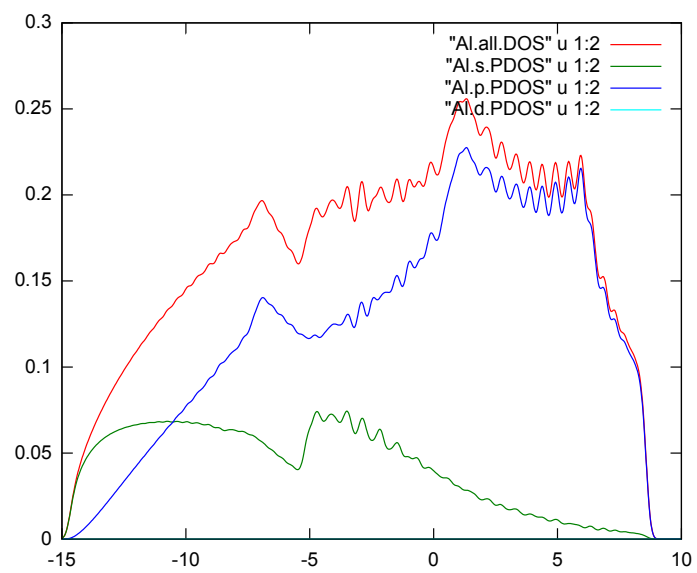


Figura 9: DOS del Al, tal y como la muestra gnuplot, con suficientes puntos k_{MP} .

6. Simulaciones de Fe

6.1. Cálculos de polarización de spin

Vamos a realizar un pequeño ejemplo de cálculo de polarización de spin con Fe en estructura FCC, la estable es la BCC, y no realizaremos una optimización del parámetro de red. El objetivo es visualizar la polarización de spin.

Entramos en la carpeta **Fe**, creamos la subcarpeta **RUN** y copiamos allí los ficheros para mantener intacta las fuentes.

```
cd Fe
mkdir RUN
cp * RUN/
cd RUN
```

Vamos a calcular la estructura de BCC del Fe, un sistema en el que tenemos que tener en cuenta la degeneración de spin. Vemos que hay 2 ficheros, en el fichero del pseudopotencial **Fe.psf** y el input del sistema **Fe.fdf**. Con este fichero de input vamos a:

1. Calcular el momento magnético
2. Dibujar la Densidad de estados

Estudiamos el fichero de input y lanzamos el cálculo. No realizamos una optimización del parámetro de celda y hemos activado la polarización de spin. Además, usamos la base recomendada para el pseudopotencial.²

```
gedit Fe.fdf
mpirun -n 4 siesta < Fe.fdf > Fe.out;
```

²<https://www.simuneatomistics.com/siesta-pro/siesta-pseudos-and-basis-database/>

Comprobamos la convergencia en la energía libre y el momento magnético.

```
grep "FreeEng" Fe.out
grep "outcell: Cell ve" Fe.out |awk '{print NR,$7}'
grep -i spin Fe.out
```

6.2. DOS

El fichero contenía los datos necesarios para realizar el gráfico de la DOS. Extraemos la información:

```
\textbf{./fmpdos}
  Input file name (PDOS):
Fe.PDOS
  Output file name :
Fe.all.PDOS
  Extract data for atom index (enter atom NUMBER,
  or 0 to select all), or for all atoms of
  given species (enter its chemical LABEL):
0
  Extract data for n= ... (0 for all n ):
0
```

Podríamos continuar para dibujar el resto de orbitales individualmente, pero procederemos a dibujar la densidad. En este caso el fichero tiene 3 columnas: la segunda representa la DOS de un spin y la tercera la del otro spin.

```
gnuplot> p "Fe.all.DOS" u 1:2 w l
gnuplot> rep "Fe.all.DOS" u 1:3 w l
gnuplot> q
```