

QE 2021 ICTP Summer School: Lecture Notes

Ignacio José Chevallier-Boutell

Índice general

1. Día 1	1
1.1. Teórico	1
1.1.1. DFT	1
1.1.2. SCF	2
1.1.3. PW como base del espacio de Hilbert	3
1.1.4. Periodicidad	4
1.1.5. Desde los orbitales KS hacia la densidad de carga	6
1.1.6. Teorema de Bloch: primera zona de Brillouin	7
1.1.7. Sistemas no 3D-periódicos	8
1.1.8. Pseudopotenciales	9
1.2. Q&A	10
1.3. Hands-on	13
1.3.1. Benzene	13
1.3.2. Graphene	15
2. Día 2	19
2.1. Teórico	19
2.1.1. DFT	19
2.1.2. DFT y orbitales KS	20
2.1.3. FFT	21
2.1.4. Dual space technique	21
2.1.5. Simetrización de la densidad de carga	21
2.1.6. Augmentation charge	22
2.1.7. Metales	22
2.1.8. Magnetización	24
2.1.9. Potenciales y energía	24
2.1.10. Electrostática en PBC	25
2.1.11. Superficies polares: corrección dipolar	25
2.2. Q&A	25
2.3. Hands-on	28
2.3.1. Si	28
2.3.2. Al	35
2.3.3. Fe	37
3. Día 3	41
3.1. Teórico 1/2	41
3.1.1. Parámetros estructurales	41
3.1.2. Fuerzas	41
3.1.3. Tensión y estrés	42

ÍNDICE GENERAL

3.1.4. Métodos de optimización estructural	43
3.1.5. Born-Oppenheimer MD	45
3.2. Q&A 1/2	45
3.3. Teórico 2/2	47
3.3.1. Procesos activados	47
3.3.2. NEB	48
3.3.3. neb.x	51
3.3.4. Aspectos generales de NEB	52
3.4. Q&A 2/2	53
3.5. Hands-on	54
3.5.1. relax	54
3.5.2. vc-relax	55
3.5.3. NEB: transferencia de H	58
3.5.4. NEB: disociación de H	61
3.5.5. pwtk automation	62

Capítulo 1

Día 1

1.1. Teórico

Topic: QUANTUM ESPRESSO: overview and basic functionalities. The self-consistent cycle. PBC: supercells and k-point sampling.

Speaker: Ralph GEBAUER (ICTP, Italy).

1.1.1. DFT

La energía del estado basal de un sistema de N electrones es un funcional de la densidad de carga electrónica $n(\vec{r})$, i.e. $E^{DFT} = E^{DFT}[n(\vec{r})]$ donde

$$n(\vec{r}) \geq 0 \quad ; \quad \int n(\vec{r}) d\vec{r} = N$$

Matemáticamente se puede probar que dicho funcional existe, pero su forma exacta es desconocida. En el contexto del formalismo de Kohn-Sham (KS) se tiene

$$E^{DFT}[n(\vec{r})] = T_s[\{\psi_i\}] + E_{ext}[n(\vec{r})] + E_{Har}[n(\vec{r})] + E_{ions} + E_{xc}[n(\vec{r})]$$

donde los orbitales KS se introducen para aproximar la energía cinética individual ($s = single$) según

$$T_s[\{\psi_i\}] = -\frac{\hbar^2}{2m} \sum_{i=1}^N \int \psi_i^*(\vec{r}) \nabla^2 \psi_i(\vec{r}) d\vec{r}$$

y satisfacen además que

$$\delta_{ij} = \langle \psi_i | \psi_j \rangle \quad i, j \in [1, N] \quad ; \quad n(\vec{r}) = \sum_{i=1}^N \psi_i^*(\vec{r}) \psi_i(\vec{r})$$

Observación

Los orbitales KS no son many-body, sino que son monoelectrónicos.

Los demás términos energéticos refieren a:

- **Externos:** atracción Coulombica entre los electrones y los núcleos.

$$E_{ext}[n(\vec{r})] = \int n(\vec{r}) V_{ext}(\vec{r}) d\vec{r}$$

- **Hartree:** repulsión Coulómbica interelectrónica.

$$E_{Har}[n(\vec{r})] = \frac{e^2}{2} \int \frac{n(\vec{r})n(\vec{r}')}{\|\vec{r} - \vec{r}'\|} d\vec{r}d\vec{r}'$$

- **Iones:** interacción Coulómbica entre iones (si el sistema en estudio es iónico).

$$E_{ions} = e^2 \sum_{IJ} \frac{Z_I Z_J}{\|\vec{R}_I - \vec{R}_J\|}$$

- **Correlación-intercambio:** es la diferencia entre el funcional E^{DFT} real (desconocido) y todos los demás términos. Presenta la principal dificultad del método, ya que el funcional $E_{xc}[n(\vec{r})]$ debe ser aproximado de alguna manera.

En la práctica entonces se minimiza E^{DFT} respecto a $n(\vec{r})$ y, por lo tanto, respecto a los orbitales KS. Esto lleva a N ecuaciones KS:

$$H^{KS}\psi_i(\vec{r}) = \left(-\frac{\hbar^2}{2m}\nabla^2 + V_{ext}(\vec{r}) + V_{Har}(\vec{r}) + V_{xc}(\vec{r}) \right) \psi_i(\vec{r}) = \epsilon_i \psi_i(\vec{r})$$

donde

$$V_{xc}(\vec{r}) = \frac{\delta E_{xc}}{\delta n(\vec{r})} \quad ; \quad V_{Har}(\vec{r}) = \frac{\delta E_{Har}}{\delta n(\vec{r})} = e^2 \int \frac{n(\vec{r}')}{\|\vec{r} - \vec{r}'\|} d\vec{r}'$$

El problema es que el Hamiltoniano H^{KS} depende de la densidad de carga y, a su vez, la densidad de carga depende de los orbitales KS, los cuales resultan de resolver justamente las ecuaciones de KS: se presenta un problema circular.

1.1.2. SCF

Para superar el problema circular planteado se recurre a un ciclo autoconsistente (Fig. 1.1). Al comienzo debemos plantear una densidad de carga inicial, la cual suele ser la superposición de los OA de átomos libres o funciones de onda áatomicas aleatorias. A partir de esto podemos construir el Hamiltoniano H^{KS} y resolver las ecuaciones KS. Los orbitales KS resultantes nos permiten definir una nueva densidad de carga.

Posteriormente, se compara la densidad de carga nueva con la anterior: si la diferencia entre ellas es mayor que el umbral deseado (criterio de convergencia), se define una nueva densidad de carga que resulta de mezclar de alguna manera ambas densidades de carga. La manera más fácil es como está en el esquema, teniendo $0 \leq \alpha \leq 1$. Esta mezcla es esencial para lograr la convergencia ya que agrega cierta fricción al sistema: si simplemente reemplazamos la densidad de carga vieja con la nueva podemos incurrir en oscilaciones infinitas.

Observación

En QE el valor de α se declara con *mixing_beta*.

Utilizando la densidad de carga mezclada, se reinicia el ciclo: se construye el Hamiltoniano H^{KS} y se resuelven las ecuaciones KS, obteniendo nuevos orbitales KS que definen una nueva densidad de carga. Cuando la comparación entre densidades de carga sea aceptada por el criterio de convergencia, se considerará que el ciclo autoconsistente ha sido convergente, finalizando el bucle. La densidad de carga asociada al estado basal es la última densidad de carga definida en el proceso.

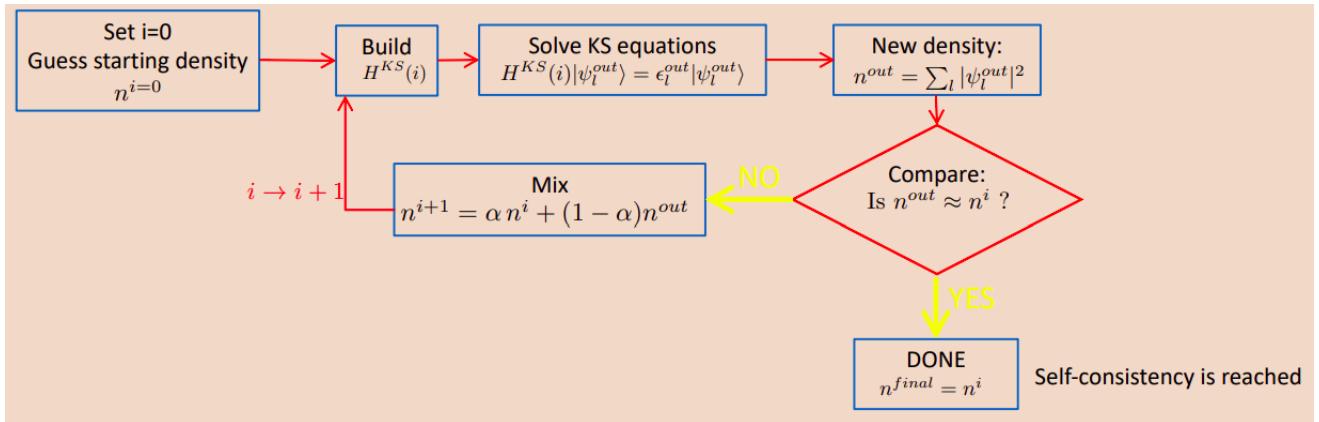


Figura 1.1: Esquema del ciclo autoconsistente.

1.1.3. PW como base del espacio de Hilbert

Dado que la computadora no puede almacenar funciones, sino que simplemente valores, necesitamos una base de funciones $\{b_\alpha\}_{\alpha=1}^M$ a la cual asociarle dichos valores. A partir de esto definimos una cierta función f como una CL de las funciones de la base según

$$f(\vec{r}) = \sum_{\alpha=1}^M c_\alpha b_\alpha(\vec{r})$$

donde los coeficientes de la expansión c_α son los que se almacenan. El valor de M debe ser lo suficientemente pequeño como para ahorrar memoria, pero lo suficientemente grande como para describir correctamente a f .

El código a utilizar es representado en principio por la base que utiliza. En el caso de QE la base está construida por plane-waves (PW):

$$b_\alpha(\vec{r}) = \frac{1}{\sqrt{V}} \exp\left\{i\vec{G}_\alpha \cdot \vec{r}\right\}$$

Observación

En QE básicamente todo está definido en términos de combinaciones lineales de senos y cosenos.

Los beneficios de utilizar PW como base son:

- Analíticamente simples, pudiendo derivar e integrar fácilmente.
- Ortonormales.
- No sesgadas: no asume la localización de las cargas ni de los electrones.
- Independientes de la posición de los núcleos: la base no va cambiando cuando se mueven los átomos (fuerzas de Pulay).
- Fácil control de la convergencia del tamaño de la base.
- Aplicación directa de FT, permitiendo aprovechar al máximo las dualidades entre el R -y el G -espacios.

Una de las principales desventajas es que el tamaño de la base suele ser mucho mayor que otras bases construidas por funciones localizadas, ya que no poseen información atómica.

1.1.4. Periodicidad

Dada la naturaleza periódica de las PW, el uso de esta base está íntimamente relacionado con la periodicidad del sistema físico en estudio: son fácilmente aplicables a sistemas periódicos ya que basta definir la red de Bravais y su motivo.

Si una función es periódica en el R -espacio, su FT es no nula únicamente para un conjunto infinito, pero discreto de vectores de onda.

Caso 1D

En el caso de 1 dimensión, se tiene que

$$\exp\{iG_\alpha x\} \rightarrow G_\alpha = n \frac{2\pi}{L} \quad n \in \mathbb{Z}$$

donde $2\pi/L$ es la distancia entre las componentes no nulas de la FT, siendo L el período (tamaño de las unidades que se repiten). A mayor L , mayor densidad de puntos no nulos en el G -espacio. La longitud de onda asociada a cierto vector de onda \vec{G}_α (Fig. 1.2) cumple

$$\lambda_\alpha \propto \frac{1}{\|\vec{G}_\alpha\|}$$

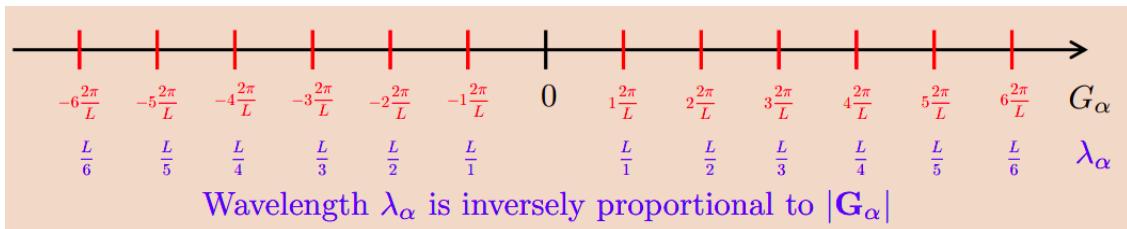


Figura 1.2: Representación de las infinitas componentes de Fourier en el G -espacio, denotando que son discretas. Se tienen además las longitudes de onda asociadas a cada vector de onda con componente no nula.

Aunque hemos logrado una discretización, para poder realizar un cálculo en la práctica necesitamos una cantidad finita de puntos. Para truncar la cantidad de puntos, uno determina una longitud de onda mínima λ_{min} , la cual sea suficiente para describir las características de interés (Fig. 1.3).

Observación

Los orbitales y, por lo tanto, la densidad de carga no varían en la escala de los núcleos atómicos, sino que varían en la escala de los Å

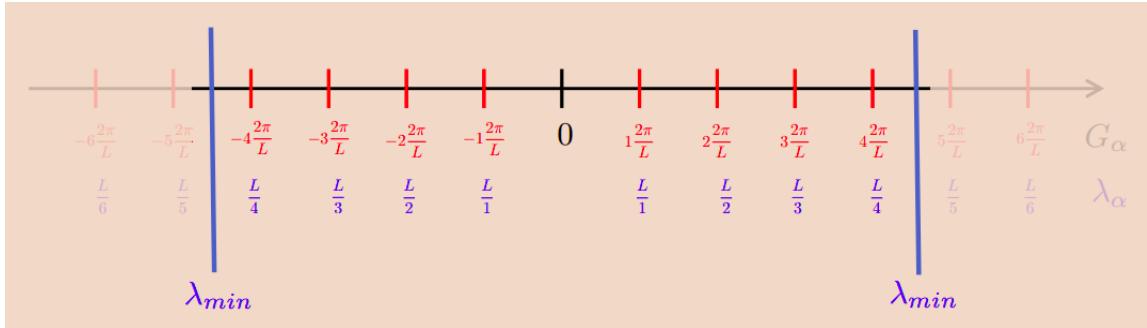


Figura 1.3: Truncamiento de las componentes no nulas en el G -espacio a partir de considerar cierto λ_{min} .

En la práctica uno no le dice al programa cuál es λ_{min} , sino que le indica la norma máxima para el vector de onda.

$$G_{max} = \frac{2pi}{\lambda_{min}} \quad ; \quad G_{max} \equiv \left\| \vec{G} \right\|_{max}$$

En QE esto se hace estableciendo el cutoff de la energía cinética:

$$E_{cut}^\psi = \frac{\hbar^2}{2m} G_{max}^2 \quad \equiv ecutwfc \text{ [Ry]}$$

Observación

Este truncamiento es el que le indica a QE el tamaño de la base de PW. Como la ubicación de los puntos no nulos está directamente ligada al parámetro de red, se puede ver que la modificación de dicho parámetro altera la base de PWs utilizada: a mayor tamaño de red, la grilla de vectores G se hace cada vez más densa y, manteniendo G_{max} fijo, se irán agregando más PW al cálculo de manera discontinua. Esto se observa en una caída brusca en la energía al agrandar el tamaño de la base. Si se presentan estas discontinuidades, hay que definir otras variables en el input: ecfixed, qcutz y q2sigma.

Caso 2D

Al considerar dos dimensiones, nuevamente tenemos el problema de que aunque son discretos, tenemos infinitos puntos. En este caso, el truncamiento es una circunferencia (Fig. 1.4). Se tiene

$$\exp\left\{i\vec{G}_\alpha \cdot \vec{r}\right\} \rightarrow \vec{G}_\alpha = m\vec{B}_1 + n\vec{B}_2 \quad m, n \in \mathbb{Z}$$

donde $\vec{B}_{1,2}$ son los vectores recíprocos de la red.

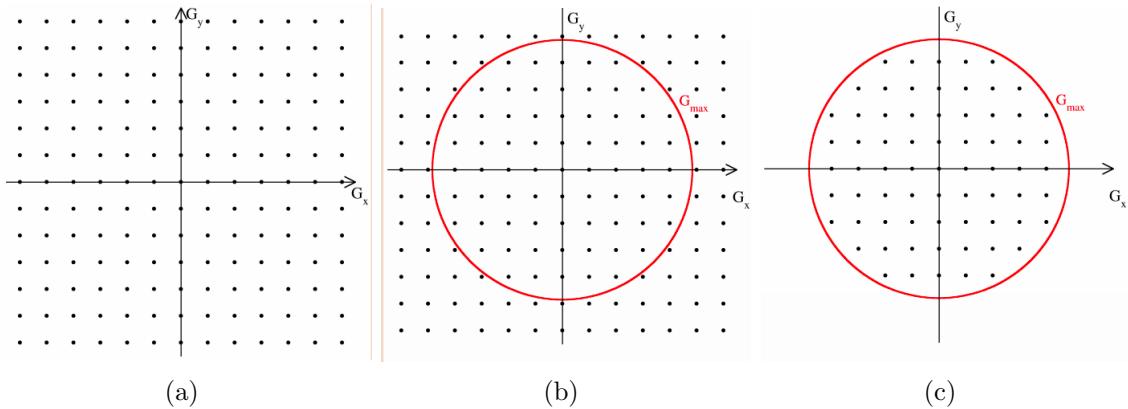


Figura 1.4: En el caso bidimensional el truncamiento se hace estableciendo una circunferencia de radio G_{max} .

Caso 3D

Extendiendo a 3 dimensiones:

$$\exp\left\{i\vec{G}_\alpha \cdot \vec{r}\right\} \rightarrow \vec{G}_\alpha = m\vec{B}_1 + n\vec{B}_2 + p\vec{B}_3 \quad m, n, p \in \mathbb{Z}$$

donde $\vec{B}_{1,2,3}$ son los vectores recíprocos de la red. Ahora el truncamiento es una esfera de radio G_{max} .

1.1.5. Desde los orbitales KS hacia la densidad de carga

Considerando todo lo dicho, los orbitales KS expresados como CL de PWs con un cutoff G_{max} son

$$\psi_i(\vec{r}) = \sum_{\|\vec{G}\| < G_{max}} c(\vec{G}) \exp\left\{i\vec{G} \cdot \vec{r}\right\}$$

En el R -espacio dijimos que la densidad de carga viene dada por $n(\vec{r}) = \sum_{i=1}^N \psi_i^*(\vec{r}) \psi_i(\vec{r})$. Al convolucionar obtenemos la densidad de carga en el G -espacio:

$$\tilde{n}(\vec{r}) = \sum_{i=1}^N \sum_{\|\vec{G}'\| < G_{max}} \tilde{\psi}_i^*(\vec{G}') \tilde{\psi}_i(\vec{G} - \vec{G}')$$

Al limitar $\|\vec{G}'\| < G_{max}$ estamos limitando las contribuciones de $\tilde{\psi}_i^*(\vec{G}')$. Sin embargo, ahora la norma de \vec{G} puede ser mayor que G_{max} pues incluso cuando $\|\vec{G}\| = 2G_{max}$ y $\|\vec{G}'\| = G_{max}$, se cumple que $\tilde{\psi}_i(\vec{G} - \vec{G}') \neq 0$.

A partir de esto el análisis de Fourier nos dice que como la densidad de carga es un producto de 2 funciones truncadas según G_{max} , la densidad tendrá un cutoff mayor. Las componentes de Fourier no nulas estarán limitadas por $2G_{max}$. Este establece que en el cálculo tendremos dos bases: una para los orbitales, limitada por G_{max} , y una para la densidad, conteniendo más puntos en el espacio recíproco ya que está limitada por $2G_{max}$. Esto lleva a la siguiente relación:

$$E_{cut}^n \geq 4E_{cut}^\psi \equiv ecutrho \quad [Ry]$$

Observación

Si no se hace esta distinción, se puede incurrir en aliasing errors. El código automáticamente establece la relación $ecutrho = 4ecutwf$, teniendo que indicarla de manera explícita cuando necesitamos que un $ecutrho$ mayor.

1.1.6. Teorema de Bloch: primera zona de Brillouin

Aunque el sistema sea periódico, los orbitales no son necesariamente periódicos. En un sistema periódico, los orbitales de Bloch se relacionan con los orbitales KS según

$$\psi_{\vec{k},i}(\vec{r}) = \exp\left\{i\vec{k} \cdot \vec{r}\right\} u_{\vec{k},i}(\vec{r})$$

donde sólo la función $u_{\vec{k},i}$ es periódica ya que \vec{k} no necesariamente es un punto de la grilla en el G -espacio, provocando que $\psi_{\vec{k},i}$ no sea periódica. Se tiene que

$$u_{\vec{k},i}(\vec{r}) = \sum_{\|\vec{G}\| < G_{max}} c_{\vec{k},i} \exp\left\{i\vec{G} \cdot \vec{r}\right\}$$

la cual está definida sobre la grilla de G -vectores antes definida. Luego

$$\psi_{\vec{k},i}(\vec{r}) = \sum_{\|\vec{G}\| < G_{max}} c_{\vec{k},i} \exp\left\{i(\vec{k} + \vec{G}) \cdot \vec{r}\right\}$$

obteniéndose una grilla corrida de $(k + G)$ -vectores. A pesar del corrimiento, el cutoff no se ve alterado (Fig. 1.5).

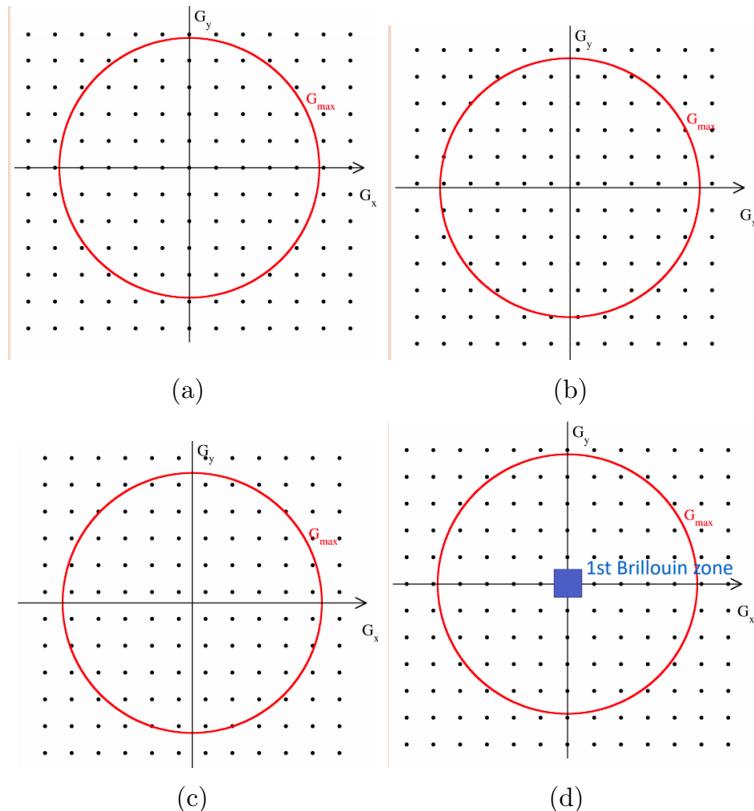


Figura 1.5: Grilla de G -vectores (a) comparada con dos posibles grilals de $(k + G)$ -vectores, según dos posibles valores para k (b y c). Se esquematiza además la 1BZ (d).

Si al vector \vec{k} se le suma un vector recíproco el resultado es indistinguible dada la periodicidad del sistema en el espacio recíproco. Esto lleva a que sólo sea necesario considerar la primera zona de Brillouin (1BZ): cualquier punto por fuera de la 1BZ será mapeado por el corrimiento a algún punto dentro de la 1BZ dada la periodicidad. De este modo, aunque en un sistema periódico debemos sumar sobre todos los posibles valores de k -vectores, los únicos relevantes son aquellos que yacen dentro de la 1BZ. Luego, es importante hacer un buen muestreo de 1BZ.

Supongamos que tenemos en 2 dimensiones una 1BZ rectangular cuyos vectores recíprocos son \vec{B}_1 y \vec{B}_2 . Deseamos muestrear la 1BZ con una grilla de k -puntos de 4x4. Si la grilla es corrida en la mitad de la distancia que separa dos k -puntos consecutivos (Fig. 1.6), la convergencia será más rápida: dada la simetría que se genera, disminuyen la cantidad de cálculos que deben ser llevados a cabo. Sin embargo, en el caso no corrido, siempre tendremos presente el punto $k = 0$ sin importar la simetría del sistema, pero se pierde al hacer el corrimiento.

Observación

En QE el corrimiento se hace cambiando los 0 por 1 en la carta de K.POINTS automatic. Los cálculos con corrimiento no son comparables entre supercells (sistemas no 3D-periódicos) de tamaños diferentes por la pérdida de $k = 0$. Generalmente conviene usar cuando tengo un sistema periódico que quiero que converja rápido.

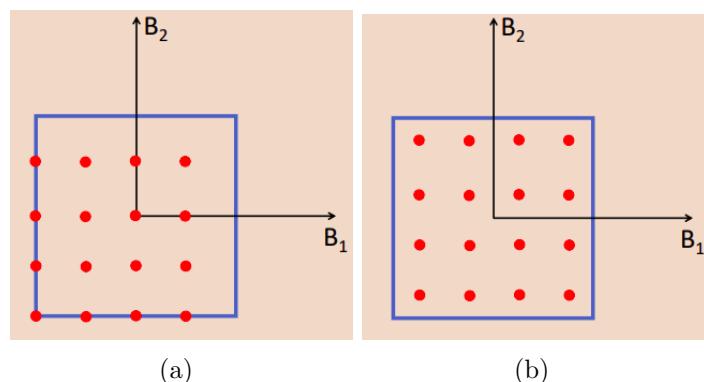


Figura 1.6: Muestreo de una 1BZ en 2 dimensiones con una grilla 4x4: original (a) y corrida (b).

1.1.7. Sistemas no 3D-periódicos

Cuando se tiene un sistema que no es 3D-periódico, la solución es utilizar una supercelda: se agrega suficiente vacío en la celda como para que el sistema no vea a sus imágenes. Suficiente quiere decir tal que las interacciones entre ellas sean despreciables. De este modo evitamos interacciones espúreas.

El problema de esto es cuando el sistema en estudio es cargado o dipolar: debido a la lentitud con la que decaen estas interacciones, deberíamos usar superceldas enormes. Para superar este problema existen dos soluciones:

1. Usar la variable `assume_isolated` en el input, la cual es muy útil en sistemas 0D (moléculas, clusters, etcétera).
2. Introducir una capa dipolar en el vacío entre dos slabs consecutivos, utilizando la variable `dipfield` en el input.

1.1.8. Pseudopotenciales

Si se considera la contribución radial de una función de onda para un cierto núcleo, se observa que en las cercanías del núcleo se presentan oscilaciones muy rápidas. Si quisieramos muestrear esta primera región, necesitaríamos un λ_{min} muy pequeño, lo que implicaría un cutoff muy grande para la energía cinética y, por lo tanto, una base de PWs mucho mayor (cálculo extremadamente caro). Sin embargo, en esta distancia no está la química pues es el core: todo ocurre a distancias mayores respecto al núcleo, donde la función ya es mucho más suave.

El pseudopotencial da lugar a una pseudofunción de onda que tiene exactamente el mismo comportamiento que la función de onda original para todo radio mayor a cierto radio de corte (r_{cut}), pero que por debajo de dicho valor tiende suavemente a cero (Fig. 1.7). Esta pseudofunción de onda queda descripta por un λ_{min} mayor, requiriéndose una base más pequeña, dando lugar a un cálculo más barato.

Observación

Los pseudopotes afectan V_{ext} en la ecuación KS.

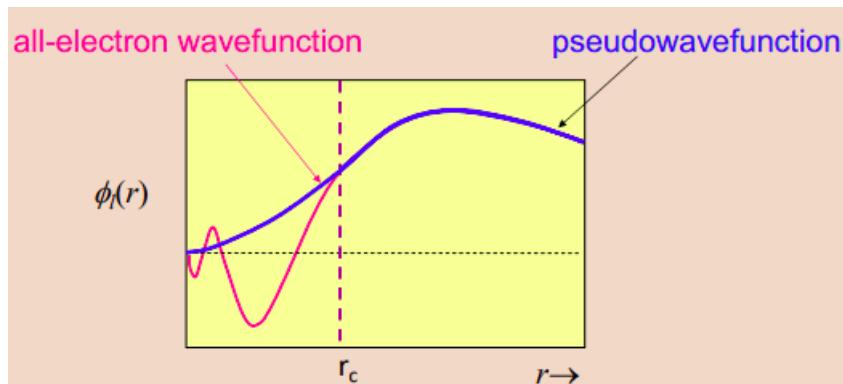


Figura 1.7: Comparación de la función de onda original (all-electron) y la función de onda generada a partir del pseudopotencial.

La condición necesaria para que el pseudopotencial sea transferible es que sea norm-conserving, *i.e.* que la norma de ambas funciones sea la misma para valores por debajo del r_{cut} . De lo contrario, el átomo no tendrá las propiedades de dispersión adecuadas.

$$\int_0^{r_{cut}} \phi_{AE}^*(r) \phi_{AE}(r) dr = \int_0^{r_{cut}} \phi_{PS}^*(r) \phi_{PS}(r) dr$$

Observación

En la página materialsclouding.org/discover/sssp se tienen pseudopotenciales confiables.

Las características de un pseudopotencial son (Fig. 1.8):

- Más débiles que un potencial coulombico.
- No presentan singularidades en $r = 0$.
- Iguales para r grandes, pero dependientes del momento angular l para r pequeños.

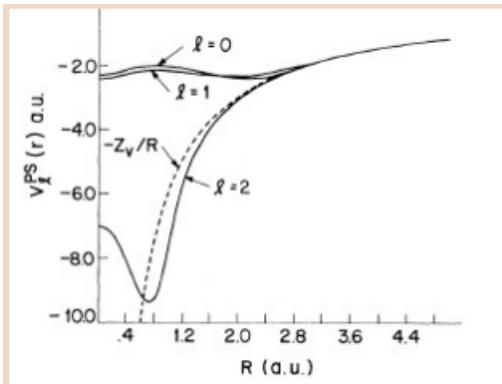


Figura 1.8: Comparación de un pseudopotencial con el potencial Coulómbico.

En QE los pseudopotenciales (PP) son provistos por el usuario en forma de archivos, que contienen la descripción del PP en una grilla radial. Los PP son específicos para cada elemento químico. Además, son dependientes del funcional utilizado: el código detecta automáticamente cuál fue el funcional utilizado y lo utilizará para resolver las ecuaciones KS. En este sentido, es conveniente que todos los PP que se usan en una misma corrida hayan sido creados considerando el mismo funcional.

Problemas con la conservación de la norma

Existen ciertos elementos para los cuales la condición de conservación de la norma termina siendo un problema: se dan cutoffs mayores de lo que uno querría. Esto ocurre especialmente cuando la función de onda no tiene nodos.

En estos casos uno puede remover la condición de conservación de norma. Sin embargo, para que los resultados sean confiables, debemos recurrir a una expresión más complicada para la densidad de carga electrónicas

$$n(\vec{r}) = \sum_{i=1}^N \psi_i^*(\vec{r}) \psi_i(\vec{r}) + n^{aug}(\vec{r})$$

donde el término n^{aug} es la augmentation charge density, la cual está localizada en torno al núcleo y da cuenta de la norma que se estaba perdiendo, expresada en términos de proyectores.

Este es el caso de los ultrasoft PP (USPP) y los PAW PP. Tenemos entonces que, a pesar de que la norma no se conserva mediante la misma condición de antes, logramos alcanzar la misma norma al sumarle este nuevo término. La consecuencia de esto es que necesitamos cutoff mayores ($E_{cut}^n > 4E_{cut}^\psi$) y, por lo tanto, debemos darle a QE ambos valores necesariamente: tanto ecutwfc como ecutrho. Esto lleva a que haya que chequear la convergencia de ambos parámetros.

1.2. Q&A

What is the reason for the weight α for the density? Why not just take n_out in the scf cycle?

The mixing parameter (not weight!) α is important for the SCF cycle to converge. If you simply take n_out , then it is very likely that you only oscillate between various densities. The mixing (which is in reality more sophisticated than what I have shown in the slides) acts like a friction term. You add only a (normally small) piece of the new density, guaranteeing that step by step you approach the self-consistent solution

what is the convergenc criterion? what does the total scf accuracy in every iteration measure? the difference between the current and previous total energy?

What the code prints as "scf accuracy" is in fact an energy: it first calculates the difference of the density between ".in" and ".out", and then estimates an energy by plugging this " $\delta \rho$ " into the formula for a Hartree energy. It is in a sense the classical Coulomb interaction energy of the change in density. As " $\delta \rho$ " goes to zero, so does this quantity. Reference: function rho_ddot in scf_mod.f90

Do we also check the density separately? How is the convergence checked?

The convergence is checked by this quantity (called dr2 in the code). It is not a change in estimated total energy, but a measure of the norm of the change in charge density

US and PAW, what should one chose and when? This ties maybe to a deeper question about how much should the final user care about their choice of PP. To what extent are the PP influencing the results? how realistic is a scenario where I completely have to throw away work because of the wrong choice of pseudopotential? What are the main pitfalls to avoid? Does this availability of multiple pseudos influence reproducibility?

From a user's perspective, the choice of either US or PAW is only relevant if you plan to do calculations where you need to obtain the additional info which PAW can provide you: in PAW you can reconstruct" the shape of the all-electron wavefunctions close to the nucleus. This info is needed for certain properties, for example if you are interested in NMR chemical shifts. If you are not interested in such (very particular) applications, then you can use US and PAW both in the same way.

About pseudopotentials in general: in the past, this was a huge issue: many old calculations you find today in the literature are in fact wrong - because of wrong pseudopotentials. Either the potentials were outright wrong, or (more common) people used pseudopotentials where some semi-core electrons were "hidden" in the core, while it would have been important to include them in the (explicitly treated) valence. Today, one knows about the importance of such "semi-core" electrons, and pays more attention.

As a user, make sure that you do not blindly download PPs from some obscure (and perhaps old) database, where you do not know if the PPs have been checked and controlled carefully. One very exemplary source for PPs (limited to PBE and PBEsol functionals) is this one: <https://www.materialscloud.org/discover/sssp/table/efficiency> Here, they have made a very thorough and exemplary job of checking everything against all-electron calculations. But also the pslibrary are generally reliable <https://www.quantum-espresso.org/pseudopotentials>

As a user, still in every case, check which is the valence configuration for every pseudopotential ("how many electrons are in the valence?") and in case of doubt, compare with another pseudopotential which contains more electrons

I'm confused about the mixing number alpha and beta, since the mixing_beta is what I include in calculations and I thought this was the mixing parameter responsible of combining old and new densities. Am I mistaken?

Sure, this was my fault. What I called α in my talk is called β in pwscf ...

In the lecture, there was this dipfield tag recommended for 2D systems. Would that apply to graphene? Could you please comment further on that tag?

No need for that in the case of pristine graphene as it does not have the electrostatic dipole moment. The dipole layer correction should be applied only in cases when you have a 2D material (which lies in x0y plane) that is an insulator and has considerable dipole moment pointed in the z-direction. In such cases the electrostatic potential is not the same on both sides of 2D material, thus the dipole layer correction should be applied. Usually you apply it when you want to calculate the work function.

If I doing scf calculation i don't know how me bands looks like. How do I know that I have insulated system or the system have dipole moment?

Best way: plot density of states with dos.x after you do scf and nscf. I recommend this.

Quickest way: in the &CONTROL flag of the pw.graphene.scf.in set the verbosity='high' and do the calculation with smearing turned on. Then in the output file you will get the occupation numbers for all the k-points. If some of them are fractional numbers then you got a metal.

NB. In case of graphene, when you do not have included the special point K in the k-point grid and you have very small smearing you will falsely get a narrow gap semiconductor (the gap is decreasing with the number of k-points but will never go away unless you include point K or increase the smearing).

When I have to use vdw correton (vdw_corr) in scf calculation?

You use vdw_corr when you have two subsystems A and B that are interacting very weakly, usually by means of dipol-(induced) dipol interaction. Those kind of interactions are called van der Waals interactions. For instance, when you adsorb graphene on top of Ni(111) and you want to calculate adsorption energy with PBE functional, if you do not put vdw_corr you won't be able to obtain the adsorption energy, it will be basically zero.

Or even simpler example - you have bilayer graphene and you use PBE functional to obtain the interaction (adhesion) energy between two sheets. If you plot E_adh vs distance you will see that there is no minimum. On the other hand, if you put vdw_corr you will clearly see the minimum at some distance between 2 and 3 angstroms

In the second example from today, why do the number of k-points change from the scf to the nscf calculation? (from 9 9 1 to 12 12 1)

because we compute the DOS with the tetrahedra method that need a denser mesh

it is not so crucial as in the scf case, in particular for DOS and pDOS is more about about the quality of the plot, but in principle yes one should check whether denser mesh change the result significantly

What are the main reasons for choosing one Pseudopotential over another?

1. accuracy ; 2. economy; 3. a tradeoff between 1 and 2

I have a question regarding k-grid. As this parameter is a consequence of periodicity - what do I do with amorphous cell? Gamma-only because of no long-range order or test k-grids?

A test is necessary... Even if you think that the system is "aperiodic", for the electrons it is not. If you have a small cell, your electrons will "see" the periodicity, and thus there is band dispersion, and thus sampling of the Brillouin zone is needed. If you have a large cell, the periodicity through the cell is no longer "seen" by the electrons, and Gamma-only is sufficient. Your system is probably between these two extrema, very small and extensively large, and thus, depending on the size and type of electronic structure, you might be forced to test for the convergence indeed.

PS: for a disordered system, you can think of k-point sampling as of a sampling over boundary conditions for the wavefunctions. At k=0, wavefunctions are periodic over the simulation cell. At zone border, they are antiperiodic, etc. As the choice of boundary conditions is to a large extent a matter of convention, it makes sense to sample different choices for them. For large enough systems, calculated properties are independent of the boundary conditions, and sampling just the Gamma point would do.

how do I choose an appropriate mixing_beta parameter for each system?

You start from the default value. Which is 0.7. In most cases this is ok. If the scf cycle does not converge, then use a smaller value. Sometimes as small as 0.1 is needed. However, if you encounter serious convergence problems in the scf cycle, then often it helps to introduce some

smearing with finite temperature. This avoids that some states discontinuously cross the Fermi level and are therefore occupied/empty in subsequent iterations. Very often, smearing helps more to make your cycle converge than changing mixing_beta

¿Podría correr con shift en los k-points para hacer una relajación y, una vez que esté relajado, seguir los cálculos sin el corrimiento?

1.3. Hands-on

Topic: How to compile Quantum ESPRESSO. SCF calculations + post-processing – part 1. XCrySDen, PWgui, QE-emas-modes, and PWTK

Speaker: Pietro DELUGAS (SISSA, Italy).

1.3.1. Benzene

Objetivo

Aprender a calcular y plotear orbitales moleculares en benceno.

Pasos

1. Hacer un cálculo SCF (pw.x) para determinar los estados KS.

```
pw.x < pw.benzene.scf.in > pw.benzene.scf.out
```

2. Calcular todos los MO de valencia y el LUMO (pp.x) según $sign(\psi(\vec{r}))\|\psi(\vec{r})\|^2$. Los resultados se escriben en un archivo psi2.benzene_K001_B0*.xsf.

```
pp.x < pp.benzene.psi2.in > pp.benzene.psi2.out
```

3. Graficar uno de los archivos XSF generados con xcrysden. Por ejemplo

```
xcrysden --xsf psi2.benzene_K001_B006.xsf
```

Luego, mejorar la calidad de la gráfica y guardar el estado (File-*Save Current State*). Supongamos que el estado fue guardado como MO-state.xcrysden, luego el estado puede levantarse con otro orbital haciendo

```
xcrysden --xsf psi2.benzene_K001_B005.xsf --script MO-state.xcrysden
```

4. Para graficar todos los MO ejecutar en la terminal

```
./plot-psi2.sh
```

Resultados

En los inputs, QE toma como comentarios a aquellas líneas que comienzan con ! o con #. Si no se escriben los valores para *pseudo_dir* y *outdir*, pw.x va a recurrir a las variables de ambiente: ESPRESSO_PSEUDO y ESPRESSO_TMPDIR, respectivamente (esto lo definimos en el .bashrc).

En la carpeta de archivos temporales se genera, por cada prefix, una carpeta (prefix.save/) y un archivo XML (prefix.xml): la carpeta contiene binarios para hacer procesamientos posteriores, mientras que en el xml se tiene información general acerca de la corrida.

Para los cálculos se utilizaron USPP con PBE como funcional:

- C.pbe-rrkjus.UPF
- H.pbe-rrkjus.UPF

Se puede usar emcas para que reconozca las variables relevantes del QE, sino se pueden crear unos archivos que permiten que vim las detecte (https://github.com/leseixas/quantum_espresso-vim).

Se conocen como namelists a aquellos bloques que comienzan con &, mientras que los demás bloques se denominan cards.

Para correr sobre M procesadores (teniendo mpi):

```
mpirun -np M pw.x -i input\_file.in > output\_file.out
```

Si escribimos

```
mpirun -np M pw.x -i input\_file.in | tee output\_file.out
```

nos irá mostrando el cálculo en pantalla además de guardarla en el archivo de destino.

En pp.x dentro del namelist &INPUTPP las variables kband(1) y kband(2) le indican desde y hasta qué orbital queremos calcular, respectivamente. En este caso queremos los 16 orbitales. Además lsign le pide que conserve el signo de la función cuando es .true.

Luego en la namelist &PLOT tenemos:

- **iflag:** como es 3 le estamos pidiendo que grafique en 3 dimensiones.
- **nfile:** le indica que sólo debe considerar un archivo.
- **weight(1):** en este caso sólo hay un archivo (nfile=1) y por eso se le pone peso 1. Sin embargo, podría tener más de un archivo que considerar y con esta variable puedo ponderar dicha consideración.

Al correr pp.x se van a generar 2 arhivos por orbital: uno propio de QE y otra para que pueda ser levantado por xcrysden.

Para que xcrysden considere la información del orbital, debemos ir a *Tools > Datagrid* y seleccionar lo que deseamos graficar (en este caso sólo habrá una opción disponible ya marcada dentro de la gráfica de árbol). El multiply factor es la densidad de carga dividido el volumen de la celda. Al estar usando una molécula asilada, la celda es mucho mayor que la densidad de carga. Luego debemos usar un valor grande: usaremos 100.

Al dar *ok* se abre un nuevo menú. Lo que nos interesa es ingresar un isovalor para la densidad de carga electrónica (cuadrado rosa vacío). Este valor debe encontrarse entre los extremos que están por encima de este cuadrado y en grisito. Esto permitirá que se genere una isosuperficie.

Conviene tildar *Render +/− isovalue* para detectar las zonas con densidad de carga de diferente signo. Al poner *Submit* vemos que sobre la molécula se grafican los orbitales.

Observación

No hay una regla para elegir el isovalor. Debe ser uno tal que permita observar claramente todas las características de lo que queremos mostrar. Si vamos a comparar entre varias isosuperficies, hay que chequear que todas queden bien presentadas para su análisis utilizando el mismo isovalor. Valores más cercanos a 0 tienden a mostrar isosuperficies más grandes, tendiendo a perderse un poco la cuestión.

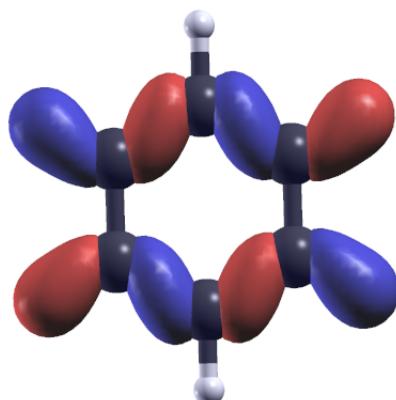


Figura 1.9: Doceavo MO para el benceno.

1.3.2. Graphene

Objetivo

Calcular y plotear DOS y estructura de bandas (spaghetti) de una hoja de grafeno.

Pasos

1. Hacer un cálculo SCF (pw.x) para determinar los estados KS.

```
pw.x < pw.graphene.scf.in > pw.graphene.scf.out
```

2. Plotear DOS:

- a) Hacer un cálculo no SCF (pw.x) con una grilla más densa de puntos k (aumenta la *definición* de la DOS).

```
pw.x < pw.graphene.nscf.in > pw.graphene.nscf.out
```

- b) Hacer el cálculo para genera el datafile de la DOS (dos.x). El resultado se escribe en el archivo graphene.dos.

```
dos.x < dos.graphene.in > dos.graphene.out
```

- c) Plotear la DOS con gnuplot.

```
gnuplot dos.gp
```

3. Plotear estructura de bandas:

- a) Hacer un cálculo de bandas (*calculation =' bands'* en pw.x) para determinar los autovalores sobre los puntos k a lo largo de cierto camino.

```
pw.x < pw.graphene.bands.in > pw.graphene.bands.out
```

- b) Hacer un cálculo para generar un archivo fácil de plotear (bands.x). El resultado se escribe en el archivo graphene.bands.dat.gnu.

```
bands.x < bands.graphene.in > bands.graphene.out
```

- c) Plotear el spaghetti con gnuplot.

```
gnuplot spaghetti.gp
```

4. Colocar el valor correcto de la energía de Fermi en los gnuplot files.

- a) Encontrar la energía de Fermi en el output (pw.graphene.nscf.out).

```
grep Fermi pw.graphene.nscf.out
```

- b) Editar los archivos dos.gp y spaghetti.gp con el valor hallado. Descomentar la línea *setyzeroaxisl - 1*

- c) Replotear ambas gráficas.

```
gnuplot dos.gp  
gnuplot spaghetti.gp
```

Resultados

Para los cálculos se utilizaron USPP con PBE como funcional (el mismo que usamos en el benceno): C.pbe-rrkjus.UPF

Las diferencias entre los inputs para el SCF y el non-SCF son:

- **calculation:** '*scf*' \Rightarrow '*nscf*'
- **k-mesh grid:** 9 9 1 \Rightarrow 12 12 1

- **occupations:** *vacio* \Rightarrow 'tetrahedraopt'

• : \Rightarrow

Como queremos calcular la DOS, necesitamos una malla más densa de puntos para tener más detalle de lo que está pasando. Básicamente hacemos que el cálculo sea más fino.

En el archivo *graphene.dos* encontramos tanto la energía de Fermi como la DoS y la DoS integrada.

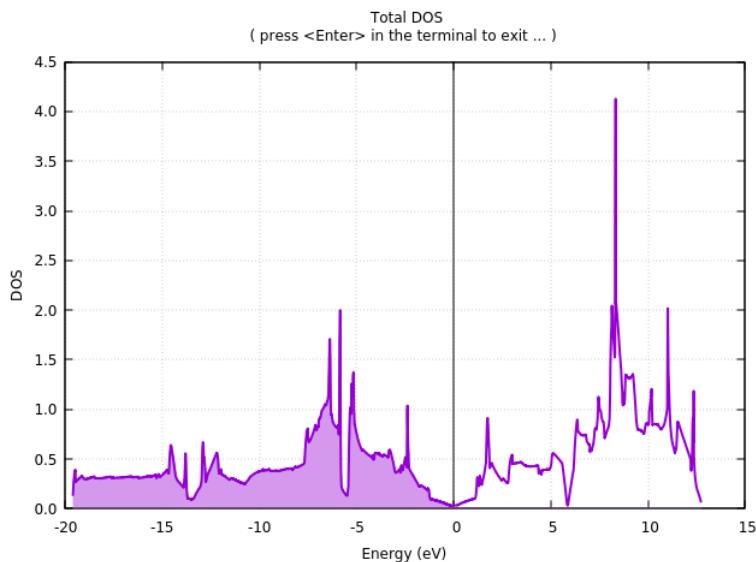


Figura 1.10: DoS para la capa de grafeno: la zona sombreada representa ocupación electrónica, llegando hasta la energía de Fermi (línea negra).

Cuando hacemos el cálculo de las bandas, debemos pedir *calculation = bands* y con *nbnd* especificamos el número de bandas: por default el programa sólo graficará las ocupadas, por lo que debemos poner un número mayor para observar bandas desocupadas. Otra diferencia es que le ponemos el camino de puntos K que queremos que siga.

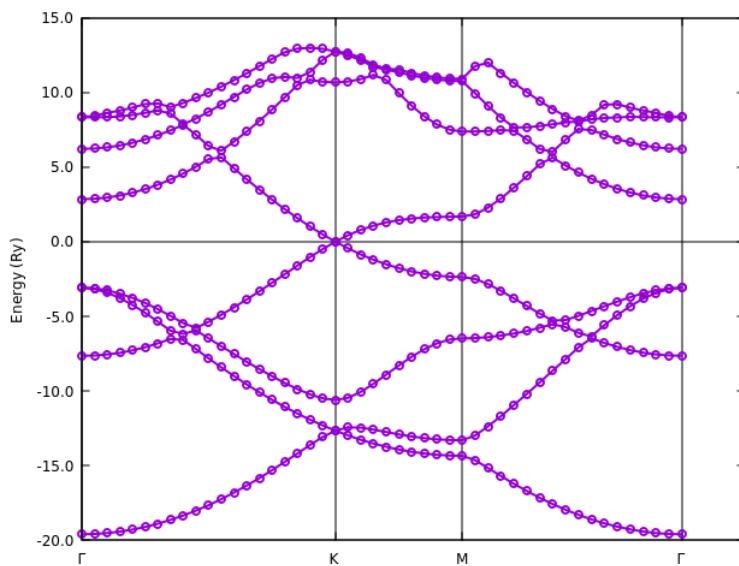


Figura 1.11: Estructura de bandas para el grafeno. Se observa el punto de Dirac al valor de la energía de Fermi.

Para seleccionar el camino de puntos k, podemos usar xcrysden. Para ello debemos ir a *Tools > k-path selection* y esto nos permitirá seleccionar interactivamente el camino deseado.

Capítulo 2

Día 2

2.1. Teórico

Topic: Charge densities and potentials. Systems in 0-1-2-3D. Metals vs. insulators. Non-magnetic vs. magnetic systems.

Speaker: Paolo GIANNOZZI (University of Udine, Italy)

2.1.1. DFT

Dada una función unidimensional L -periódica f , la transformada de Fourier (FT) nos permite establecer que

$$f(x) = \sum_n \tilde{f}(q_n) e^{iq_n x} ; \quad \tilde{f}(q_n) = \frac{1}{L} \int f(x) e^{-iq_n x} dx$$

donde

$$q_n = n \frac{2\pi}{L} ; \quad n \in \mathbb{Z}$$

En el caso de la transformada de Fourier discreta (DFT) se considera un conjunto finito de valores, tanto para q como para x , volviéndose ambas variables periódicas: los valores negativos son mapeados al correspondiente positivo.

$$\begin{aligned} f(x) &\rightarrow f_m \equiv f(x_m) ; \quad x_m = m \frac{L}{N} ; \quad m \in [0, N-1] \\ \tilde{f}(q) &\rightarrow \tilde{f}_n \equiv \tilde{f}(q_n) ; \quad q_n = n \frac{2\pi}{L} ; \quad n \in [0, N-1] \end{aligned}$$

El valor de N tiene que ser lo suficientemente grande como para incluir correctamente todas las componentes de Fourier en el q -espacio.

Reuniendo todo esto, la DFT se escribe como

$$f_m = \sum_{n=0}^{N-1} \tilde{f}_n e^{i2\pi \frac{nm}{N}} ; \quad x\text{-espacio} \quad \tilde{f}_n = \sum_{m=0}^{N-1} f_m e^{-i2\pi \frac{nm}{N}} ; \quad q\text{-espacio}$$

Para extender esto a 3 dimensiones, consideramos primero un conjunto de vectores de la red recíproca \vec{G} centrados en el origen, dados por

$$\vec{G} = n'_1 \vec{G}_1 + n'_2 \vec{G}_2 + n'_3 \vec{G}_3$$

siendo $\vec{G}_{1,2,3}$ los vectores primitivos. La grilla sobre el G -espacio es periódica por construcción. Análogamente en el R -espacio se tiene que

$$\vec{r} = \frac{m_1}{N_1} \vec{R}_1 + \frac{m_2}{N_2} \vec{R}_2 + \frac{m_3}{N_3} \vec{R}_3 \quad ; \quad m_s \in [0, N_s - 1]$$

siendo $\vec{R}_{1,2,3}$ los generadores de la red de Bravais. En el caso 3D también debemos tomar un conjunto finito. Para ello

$$f(\vec{r}) = \sum_G \tilde{f}(\vec{G}) e^{i\vec{G}\cdot\vec{r}} \rightarrow f(m_1, m_2, m_3) \quad \tilde{f}(\vec{G}) = \frac{1}{\Omega} \int f(\vec{r}) e^{-i\vec{G}\cdot\vec{r}} d\vec{r} \rightarrow \tilde{f}(n_1, n_2, n_3)$$

$$f(m_1, m_2, m_3) = \sum_{n_1, n_2, n_3} \tilde{f}(n_1, n_2, n_3) \prod_j e^{i2\pi \frac{n_j m_j}{N_j}}$$

$$\tilde{f}(n_1, n_2, n_3) \frac{1}{N} \sum_{m_1, m_2, m_3} f(m_1, m_2, m_3) \prod_j e^{-i2\pi \frac{n_j m_j}{N_j}}$$

donde $N = N_1 N_2 N_3$ y $\vec{G}_i \cdot \vec{R}_j = 2\pi\delta_{ij}$.

2.1.2. DFT y orbitales KS

Ya vimos que

$$\psi_i(\vec{r}) = \frac{1}{V} \sum_G c_{\vec{k}+\vec{G}} \exp\left\{i(\vec{k} + \vec{G}) \cdot \vec{r}\right\} \quad ; \quad \frac{\hbar^2}{2m} \left\| \vec{k} + \vec{G} \right\|^2 \leq E_{cut}$$

La densidad de carga es entonces

$$n(\vec{G}') = \sum_G \sum_{i,k} f_{i,k} c_{i,\vec{k}+\vec{G}}^* c_{i,\vec{k}+\vec{G}+\vec{G}'}$$

con lo que aparecen componentes \vec{G}' tales que $\max(\|\vec{G}'\|) = 2\max(\|\vec{G}\|)$.

Esto lleva a que sea necesario el cutoff de la energía cinética para las componentes de Fourier, tanto en la densidad de carga como en los potenciales, sean al menos 4 veces mayores que el cutoff de a base de PWs.

$$\frac{\hbar^2}{2m} \left\| \vec{G} \right\|^2 \leq 4E_{cut}$$

Esta inecuación es la que nos indica qué tan grande debe ser la grilla: debe ser suficiente para incluir todos los G -vectores hasta satisfacerla. Generalmente la grilla incluye también componentes de Fourier que no son útiles (más allá del cutoff recién dicho).

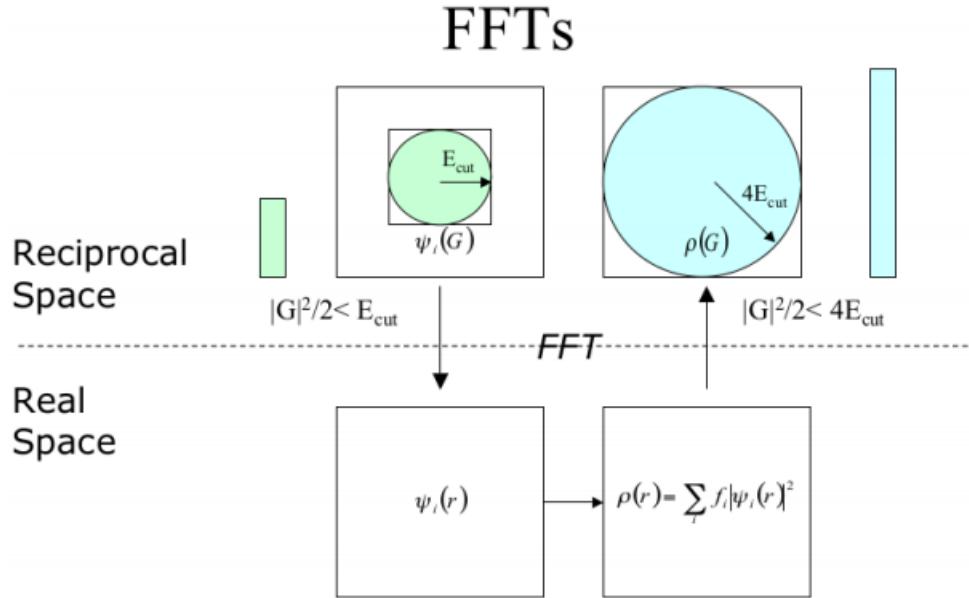


Figura 2.1: El cuadrado representa la caja (grilla) sobre la que se efectúa la FT. Para los orbitales KS sólo se requiere $1/8$ de la misma (izquierda), mientras que para la densidad de carga requiero ocupar todavía más.

2.1.3. FFT

Mientras que el costo computacional de la FT convencional para n partículas es tal que $t_{CPU} = \mathcal{O}(n^2)$, el costo para la transformada rápida de Fourier (FFT) va como $t_{CPU} = \mathcal{O}(n \log n)$. Esto da lugar a una enorme mejora en los cálculos PW.

2.1.4. Dual space technique

Teniendo

$$H\psi = (T + V_{NonLoc} + V_{loc} + V_H + V_{xc})\psi$$

se cumple que

- $T\psi$ es más fácil de calcular en el G -espacio: $t_{CPU} = \mathcal{O}(n)$
- $(V_{loc} + V_H + V_{xc})\psi$ es más fácil de calcular en el R -espacio: $t_{CPU} = \mathcal{O}(n)$
- $V_{NonLoc}\psi$ es fácil en ambos espacios siempre que el operador sea separable en proyectores: $t_{CPU} = \mathcal{O}(mn)$ siendo m la cantidad de proyectores utilizados.

Con esto en mente, se utiliza la FFT para ir y volver entre el R -espacio y el G -espacio a conveniencia.

2.1.5. Simetrización de la densidad de carga

La densidad de carga calculada por el código está no simetrizada:

$$\tilde{n}(\vec{r}) = \sum_{k \in IBZ} \sum_v \omega_k \|\psi_{k,v}(\vec{r})\|$$

donde la suma corre sólo sobre estados ocupados y k -puntos simétricamente inequivalentes, los cuales pertenecen a la zona de Brillouin irreducible (IBZ), la cual es la 1BZ reducida por

todas las operaciones de simetría del grupo puntual al que pertenece el cristal. Los coeficientes ω_k pesan la simetría: es la cantidad de k -puntos que son equivalentes a k por simetría.

Observación

Tanto la grilla de k -puntos sobre la IBZ como los ω_k pueden ser calculados o provistos por el usuario.

La verdadera densidad de carga se obtiene via simetrización:

$$n(\vec{r}) = \frac{1}{N_S} \sum_{n=1}^{N_S} \hat{O}_n \tilde{n}(\vec{r}) = \frac{1}{N_S} \sum_{n=1}^{N_S} \tilde{n}(O_n^{-1} \vec{r})$$

donde \hat{O}_n es la n -ésima simetría del cristal y O_n es la matriz de rotación asociada. Para grupos que contienen operaciones de simetría con translaciones fraccionarias, lo anterior se generaliza a

$$n(\vec{r}) = \frac{1}{N_S} \sum_{n=1}^{N_S} \tilde{n}(O_n^{-1} \vec{r} - \vec{f}_n)$$

siendo \vec{f}_n la translación fraccionaria con respecto al vector de red para la n -ésima operación.

Observación

En pw la simetrización es llevada a cabo en el G -espacio por lo que

$$n(\vec{G}) = \frac{1}{N_S} \sum_{n=1}^{N_S} \tilde{n}(O_n^{-1} \vec{G}) \exp\left\{-i O_n^{-1} \vec{G} \cdot \vec{f}_n\right\}$$

2.1.6. Augmentation charge

Para USPP y PAW PP se agrega el término de augmentation charge. Así

$$n^{aug}(\vec{r}) = \sum_{\mu, l, m} \langle \psi_i | \beta_l \rangle Q_{\mu, l, m}(\vec{r}) \langle \beta_m | \psi_i \rangle$$

donde μ corre sobre los núcleos mientras que β_l y $Q_{\mu, l, m}$ son funciones de corto alcance centradas en el átomo μ .

El primer término de $n(\vec{r})$, dado por $\|\psi_i(\vec{r})\|^2$, se calcula fácilmente con la misma grilla de FFT utilizada en el cálculo de $H\psi$, conteniendo las componentes de Fourier del G -espacio hasta la energía cinética de $4E_{cut}$. El término de augmentation charge se calcula en una grilla más grande y densa, conteniendo G hasta un cutoff mayor que $4E_{cut}$.

2.1.7. Metales

Densidad de carga en metales

La manera más directa de calcular con metales es introducir un ensanchamiento de los niveles discretos. Así

$$n(\vec{r}) = \sum_i f_i \|\psi_i(\vec{r})\|^2$$

donde $f_i \in [0, 1]$ son ocupaciones fraccionarias dadas por

$$f_i = \int_{\epsilon < \epsilon_F} \delta(\epsilon - \epsilon_i) d\epsilon ; \quad \sum_i f_i = N_{\epsilon_F}$$

y la energía de Fermi ϵ_F queda determinada por la condición $N_{\epsilon_F} = N_{elec}$. El integrando $\delta(x)$ es la función (normalizada) de ensanchamiento. Aunque uno podría utilizar la función de Fermi-Dirac en principio, no es conveniente ya que se necesitaría una temperatura muy alta y la distribución tendría colas muy largas.

$$\delta_{FD}(x) = \frac{1}{e^{x\beta} + 1}$$

Energía en metales

La manera de calcular la energía es

$$E_{KS} = \sum_i \int_{\epsilon < \epsilon_F} \epsilon \delta(\epsilon - \epsilon_i) d\epsilon = \tilde{E}_{KS} + \delta E_{met}$$

donde

$$\tilde{E}_{KS} = \sum_i \epsilon_i \int_{\epsilon < \epsilon_F} \delta(\epsilon - \epsilon_i) d\epsilon = \sum_i f_i \epsilon_i ; \quad \delta E_{met} = \sum_i \int_{\epsilon < \epsilon_F} (\epsilon - \epsilon_i) \delta(\epsilon - \epsilon_i) d\epsilon$$

Esta contribución δE_{met} aparece en el output como *smearin contrib.* ($-TS$) [Ry].

Funciones de ensanchamiento

Una posibilidad es la función Gaussiana normalizada

$$\delta(x) = \frac{1}{\sigma \sqrt{\pi}} e^{-\frac{x^2}{\sigma^2}} ; \quad \int \delta(x) = 1$$

donde σ (medido en Ry y escrito como *degauss* en el input) da lugar a una temperatura ficticia $T = \sigma/k_B$ a la cual se corresponde un funcional de energía libre $E_{KS}[\sigma]$.

Se puede ver que $E_{KS}[\sigma] \propto \sigma^2$ por lo que a mayor σ (lo cual hace falta para una convergencia rápida de k -puntos) dan lugar a mayores diferencias entre la energía *libre* y la real.

Observación

A mayor σ , menor es la cantidad de k -puntos necesarios. Lo ideal sería minimizar ambas situaciones en sentido de que la convergencia sea más económica.

Otra posibilidad es utilizar los *smearing fríos* de Methfessel-Paxton (MP) o de Marzari-Vanderbilt (MV). En este caso el término cuadrático desaparece haciendo que $E_{KS}[\sigma] \propto \sigma^4$, dando lugar a una mejor convergencia (ver <http://theosrv1.epfl.ch/Main/ElectronicTemperature>). A pesar de esto:

- Con MP se pueden dar ocupaciones negativas o mayores a 1.
- Con MV las ocupaciones son no negativas, pero igual pueden superar la unidad.
- En ambos casos no se asegura que $N(\epsilon)$ sea una función monotónica. En consecuencia, la energía de Fermi puede no estar únicamente definida.

2.1.8. Magnetización

En el caso no polarizado, cada orbital se encuentra doblemente ocupado y la densidad de carga será entonces

$$n(\vec{r}) = 2 \sum_i f_i \|\psi_i(\vec{r})\|^2$$

En el caso de magnetización colineal (LSDA), cada orbital tiene o spin up o spin down. Esto da lugar tanto a una densidad de carga como a una magnetización

$$\begin{aligned} n(\vec{r}) &= n_+(\vec{r}) + n_-(\vec{r}) \quad ; \quad m(\vec{r}) = n_+(\vec{r}) - n_-(\vec{r}) \\ n_{\pm}(\vec{r}) &= \sum_i f_{i,\pm} \|\psi_{i,\pm}(\vec{r})\|^2 \end{aligned}$$

Observación

En QE el índice \pm está oculto en el indexado de los k -puntos: se dobla la cantidad de k -puntos, donde el primer conjunto corresponde a spin up y el segundo conjunto a spin down.

Para el caso más general donde la magnetización es no colineal, la base de PWs debe ampliarse con los spinores quedando

$$\Psi(\vec{r}) = \psi_+ \chi_+ + \psi_- \chi_-$$

Tenemos entonces la densidad de carga

$$n(\vec{r}) = \sum_i f_i [\|\psi_{i,+}(\vec{r})\|^2 + \|\psi_{i,-}(\vec{r})\|^2]$$

y un vector de magnetización

$$\vec{m}(\vec{r}) = \sum_i f_i \Psi_i^\dagger(\vec{r}) \vec{\sigma} \Psi_i(\vec{r})$$

donde $\vec{\sigma}$ corresponde a las matrices de Pauli.

2.1.9. Potenciales y energía

Los siguientes potenciales de largo alcance son divergentes (separadamente) para $\vec{G} = 0$: V_{Har} y V_{loc} . Esto se da sólo cuando hay cargas, ya que cuando el sistema es neutro no hay divergencia en dicho origen. Los demás términos son de corto alcance (V_{xc} y V_{NonLoc}).

A partir de los potenciales la energía total se escribe como

$$E = E_{KS} - E_{Har} - \int n(\vec{r}) V_{xc} d\vec{r} + E_{xc} + E_{Ewald}$$

donde

- $E_{KS} \rightarrow$ suma de las contribuciones monoelectrónicas de los orbitales KS, incluyendo la energía electrón-ión.
- $E_{Har} \rightarrow$ energía electrónica electrostática con la singularidad levantada. Se resta porque se considera dos veces en E_{KS} .
- $E_{KS} \rightarrow$ se extrae de E_{KS} y se lo agrega como E_{xc} .
- $E_{KS} \rightarrow$ interacción interiónica en presencia de un fondo neutralizante.

2.1.10. Electrostática en PBC

Como consecuencias de la periodicidad se tiene que

- La carga neta por unidad de celda debe ser nula o la energía será divergente.
- Todos los potenciales son periódicos con la red, no pudiendo utilizar campos eléctricos macroscópicos.
- El cero de la energía es arbitrario.
- El momento dipolar por celda unitaria suele estar mal definidos.

En el caso de sistemas cargados, la singularidad en el origen del G -espacio se redefine la energía con una redefinición del potencial en dicho punto, por lo que la energía es dependiente de la elección. Además la comparación entre energías con diferente N_{elec} para el mismo sistema no es confiable. Por último no hay garantías de que la optimización estructural arroje resultados confiables.

Si el sistema es finito (sea cargado o no), *i.e.* no es periódico, se pueden aplicar diferentes trucos que corrijen la energía o el potencial y la energía.

2.1.11. Superficies polares: corrección dipolar

Las superficies polares tienen un dipolo. Al usar un slab en PBC, el dipolo produce interacciones ficticias que caen muy lentamente entre el sistema real y sus imágenes. Para remover esta interacción espúrea uno puede agregar un dipolo que compense la situación en la región vacía del espacio.

Observación

Ver trabajo: potential profile for MoS₂ on Au surfaces (Pedram Khakbaz et al, Solid State Electronics, in press).

2.2. Q&A

Sobre el uso del smearing en la silice

The most likely source of trouble in your case is the fact that models of polar/non-polar interfaces easily get charged and thus develop unphysical electric fields across the sample. An analysis of the intermediate charge-density distribution and/or of the resulting electrostatic potential may help diagnose the problem. The cure is likely a reconstruction of the interface to make it neutral

I think one problem is that you used Gaussian broadening as a smearing function. It is known that this broadening function leads to a strong dependence of the total energy on sigma. For this reason, the “cold” smearing of Marzari-Vanderbilt has been invented: the energy depends less on sigma, for small sigma

You might want to look at chapter 4 of Marzari’s PhD thesis, from here <http://theosssrv1.epfl.ch/Main/T...>

How could I choose the best smearing depending on the material? For example for iron is usual to use M-V instead M-P? Also, is there a way to predict the value of degauss?

You need to do a test. The advantage of M-V with respect to M-P smearing is that the occupation numbers are guaranteed to be always non-negative with M-V. As a rule of thumb,

M-V is always a good choice, except if your aim is to simulate real temperature effects on the electrons. In that case, you must use Fermi-Dirac smearing.

How do we choose the kpoints for the cutoff convergence tests?

a mesh dense enough for the calculation to converge but the smaller as possible. As long as you use the same mesh the convergence test is valid, no matter what mesh you use.

Typically any reasonable k-point mesh will do the job. Convergence tests can be done one parameter at the time, fixing reasonable values of the other parameters. For metals, however, the value of the broadening and the k-point grid are correlated.

what is the unit of Ryin ecut

12*13.6058 eV

What's the criterion to choose the best ecutwfc?

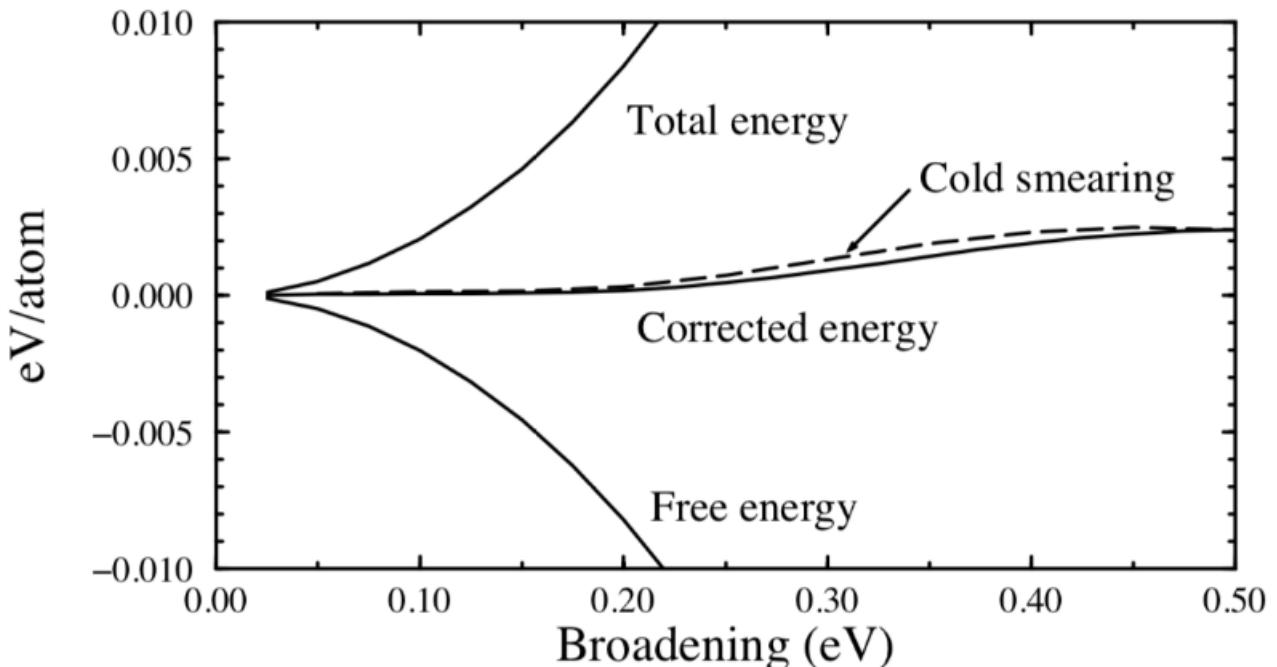
It depends on the accuracy that you desire for the computed quantity. Also for the k-points mesh.

In the case of Al (metal), the converged kpoints are clear, but how to determine an accurate degauss value? so then should i choose the smallest value of the "plateau in the middle of the figure?

in principle it should be as small as possible but not so small as to cause convergence problems. Something like that you can choose. better to be more conservative

Wait a minute! In a metal, the function which you need to integrate over the Brillouin zone is like a "step function" at the Fermi surface. The higher you choose the degauss, the smoother that function becomes. Therefore, for a higher value of degauss you need less k-points (because the function to be integrated is smoother) and inversely, a very small value of degauss entails a huge number of k-points.

In principle you can create a graph like this, where for every value of degauss you plot the energy (with a converged set of k-points, which might be different for different degaus values, see above). Then, you decide which value of the energy is "good enough" for your needs.



I have seen the discussion about relax in zoom. When we should use relax instead of "vc-relax? I have seen relax in papers related to molecules (with complex carbon structures) is this a correct approach?

in general I'd use vc-relax only to obtain the DFT lattice parameters of the crystal bulk structure, which are typically slightly different than the experimental ones. For trying different

atomic configurations in the cell relax should be used (after you already obtained appropriate cell parameters) , but it will be discussed tomorrow yes

if we have a ferromagnetic system, how do we know in which direction the magnetization vector is pointing? Where do we find the components of the magnetization vector?

In order to know this information, you need to setup a non-collinear calculations with spin-orbit- in this way you can explore different directions of the magnetitazion and the relative energy differences for each specific direction you fix in the input file. You can even setup a constrained magnetic moment calculations to adress these energy differences- When performing non-collinear calculations, in the output file you access to the cartesian components of the magnetization for each magnetic atoms-

In the case of a defect calculation, should I first converge the supercell size or is it better to converge first ecut, kpoints, etc., on a smaller cell and do the cell size convergence after?

a smaller cell is better, but you have to keep the density of k-points consistent.

I think its better if you first converge the ecut and kpoints and than the supercell size.

Thank you! Regarding the reducing number of the k points, if the k-points converged for 8x8x8 in a 2x2x2 supercell and I need a 3x3x3 supercell, which k-points can I use (as it is not a divisor)? Should I round up (6x6x6)?

closest integer, preferably rounded up

Question: What is the criteria for cartegorizing a magnetic material as anti-ferromagnetic or ferromagnetic with respect to the total magnetization, absolute magnetization and magnetic moment?

For ferromagnetic materials, the total and absolute magnetization are (almost) the same. For AFM, total magnetization is 0, absolute magnetization is not.

I have a question. If we use very high E_cut in PAW e.g. 120Ry with E_rho = 4E_cut, will the result be acceptable?

Likely yes, but it is not a smart (that is: efficient) choice. The higher Ecut means the more cost of CPU.

Another question is, If I do not have metal, is better to use gauss smearing?

For insulators you usually don't need smearing. In case if you use it for getting faster convergence use gaussian smearing with a very small degauss value.

Why did the professor use starting magnetization 0.6 and -0.6 for AFM iron? My question is about the number, can I use 0.5 and -0.5 or even 1 and -1? Is there a restriccción?

you can also start with 0.5. it will converge at the end. If you have rough idea about the number, the simulation will be faster. What you provide is just an initial guess. The code will eventually converge to you different value

I got some doubts about testing convergence. Suppose I've got a slab of an insulator inside a supercell with, let's say, 20 atoms (10 of element A and 10 of element B) and it's the minimal slab I need (I can create greater surfaces multiplying this unit in the xy plane) 1. How should I run the tests for the cutoffs? I mean should I sweep values for the whole slab or how? Suppose now that in order to converge I need to add smearing. 2. What should be my "first choice" since it's an insulator: gauss or MV? 3. Since it's not a metal I understand that the smearing contribution should be the least possible. Can I apply the same criterion we used in the hands-on (there was a 3D system and here it is a slab)? There we said that the we considered the convergence achieved with 1mRy/atom so here I should seek for a value of smearing contribution to be lesser than 20mRy?

1.) the ecutwfc and ecutrho are properties of the pseudopotential, so you only have to test it for each atomic type in your system (2 in your example), you can do the test for the isolated atom or the bulk structure of the crystal, then you use the highest/hardest cutoff for your supercell calculation. the cell size shouldn't matter for the ecut, what matters is that the energy doesn't change once you increase the cutoff further

2.) the smearing is a bit more difficult, but you can use mv/mp for insulators as well, but the smearing there should be really small (no bigger than 0.01 Ry), basically you add it only to help in the convergence, once you choose it keep it the same for all calculations (supercells included), so in principle you can determine it for the bulk of the insulator, note that the 0.001 Ry/atom is the criterion for the cutoff not the smearing typically.

in any case keep an eye on the smearing contribution written in the output and make sure its not significant and at least consistent between different calculations

2.3. Hands-on

Topic: SCF calculations + post-processing – part 2. Exercises.

Speaker: Anton KOKALJ (Jožef Stefan Institute, Slovenia).

2.3.1. Si

Objetivo

Estudiar la convergencia de cálculos pw.x para diferentes Si bulk: primero se estudia la convergencia del cutoff para las funciones de onda (*ecutwfc*) y luego la convergencia para los puntos k .

Las pruebas de convergencia se llevan a cabo haciendo una serie de cálculos variando algún parámetro. Para esto se pueden utilizar shell scripts o PWTK scripts. La diferencia entre estos puede observarse al comparar, por ejemplo, ex1.ecutwfc.classic/ecutwfc.sh con ex1.ecutwfc/ecutwfc.pwtk, respectivamente.

La lógica a seguir es la siguiente:

1. Convergencia de la base (*ecutwfc*).
2. Convergencia de los puntos k , utilizando el valor óptimo de *ecutwfc*.
3. Convergencia del parámetro de red, utilizando los valores óptimos para *ecutwfc* y la grilla de puntos k .
4. Calcular la estructura de bandas, utilizando los valores óptimos para *ecutwfc*, la k -grilla y el parámetro de red.

Pasos

1. Convergencia de *ecutwfc*: correr alguno de los siguientes comandos a gusto.

```
./ecutwfc.sh          (shell)
pwtk ecutwfc.pwtk    (PWTK)
```

2. Convergencia de k -grid.

```
pwtk kpoints.pwtk
```

3. Convergencia del parámetro de red: agregar parámetros necesarios al script antes de correr. Primero corre pw.x y luego ev.x para obtener el parámetro de red y el módulo de Young recurriendo a la ecuación de estado de Murnaghan.

```
pwtk alat.pwtk
```

4. Estructura de bandas: agregar parámetros necesarios al script antes de correr. Luego de la corrida, poner la energía de Fermi adecuada en el archivo plot.gp. Regraficar la estructura de bandas.

```
pwtk bands.pwtk
grep 'highest occupied level' pw.Si.scf.out
gnuplot plot.gp
```

Resultados: Si

Para los cálculos se utilizaron NCPP con LDA (PZ) como funcional: Si.pz-vbc.UPF.

Para examinar cómo avanza la autoconsistencia, podemos ejecutar

```
grep -e "total energy" -e estimated output\_file.out
```

Notar que hay 9 electrones dentro de la celda ya que hay 2 átomos por celda con 4 electrones cada uno. Como el sistema es no magnético y es aislante, sólo se computan las 4 ($= 8/2$) bandas de valencia (estados KS) más bajas.

Es recomendable usar PWTK ya que está hecho explícitamente para funcionar con QE: la sintaxis se mantiene como en el input original.

<pre>#!/bin/sh rm -f si.etot_vs_ecut.dat for ecut in 12 16 20 24 28 32 do cat > pw.si.scf.\$ecut.in << EOF &CONTROL prefix='silicon', / &SYSTEM ibrav = 2, ceilldm(1) = 10.2, nat = 2, ntyp = 1, ecutwfc = \$ecut, / &ELECTRONS / ATOMIC_SPECIES Si 28.086 Si.pz-vbc.UPF ATOMIC_POSITIONS Si 0.00 0.00 0.00 Si 0.25 0.25 0.25 K_POINTS automatic 4 4 4 1 1 1 EOF pw.x -in pw.si.scf.\$ecut.in > pw.si.scf.\$ecut.out grep -e 'kinetic-energy cutoff' -e ! pw.si.scf.\$ecut.out \ awk '/kinetic-energy/ {ecut=\$(NF-1)} \ !/ {print ecut, \$(NF-1)}' >> si.etot_vs_ecut.dat done</pre>	Unix shell-script	<pre>load_fromPWI .../pw.si.scf.in set fid [open si.etot_vs_ecut.dat w] foreach ecut (12 16 20 24 28 32) { SYSTEM "ecutwfc = \$ecut" runPW pw.Si.scf.\$ecut.in puts \$fid "\$ecut [::pwtk::pwo::totene pw.Si.scf.\$ecut.out]" } close \$fid</pre> PWTK script
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Figura 2.2: Comparación de un script en shell y uno en PWTK.

Los PWTK scripts son Tcl-scripts donde:

- Se mantienen los nombres de las namelists y las cards, salvo contadas excepciones. Su contenido debe ir entre corchetes (pensarlo como una función de C).
- Se pueden escribir tanto números como expresiones matemáticas, pero no debe haber espacios que separen los términos y factores de la expresión.
- Se puede indexar usando, por ejemplo, los nombres de los átomos.
- Es casesensitive, pero el orden de las namelists y las cards no es importante.
- Las variables de los namelists pueden configurarse on-the-fly.
- Tanto las namelists como las cards pueden ser llamadas múltiples veces. Sin embargo, PWTK no las considera de igual forma: las cards son tratadas en overwrite mode, mientras que las namelists son tratadas en append mode.
- Para desactivar una variable dentro de una namelist basta con darle un valor vacío, *i.e.* no poner nada después del igual.

Algunas de las funciones que se usan de PWTK en este hands-on son:

- **load_fromPWI:** carga input data desde un input de pw.x existente.
- **pwo_totene:** devuelve la energía total de un output de pw.x.
- **seq:** devuelve una secuencia de números. El primero y el último número son los extremos del intervalo (cerrado). El valor del medio indica el salto.
- **runPW / runPP / runDOS / runPROJWFC:** construye un input file del programa dado (PW, PP, DOS o PROJWFC) y corre el cálculo.

Respecto al barrido de $ecutwfc$, asumimos que el sistema es convergente cuando la diferencia entre los valores de energía total entre dos mediciones sucesivas de $ecutwfc$ es menor que $1 \text{ mRy}/\text{atom/cell}$. En este caso tenemos 2 átomos por celda así que serían 2 mRy . De la gráfica podemos pensar que 24 Ry sería suficiente. En números: $E_{tot}(24\text{Ry}) = -15,8508 \text{ Ry} \wedge E_{tot}(20\text{Ry}) = -15,8475 \text{ Ry} \Rightarrow \Delta E_{tot} = 0,0033 \text{ Ry} = 3,3 \text{ mRy}$

$E_{tot}(24\text{Ry}) = -15,8508 \text{ Ry} \wedge E_{tot}(28\text{Ry}) = -15,8519 \text{ Ry} \Rightarrow \Delta E_{tot} = 0,0011 \text{ Ry} = 1,1 \text{ mRy}$
Se concluye entonces que el valor óptimo es $ecutwfc = 24$.

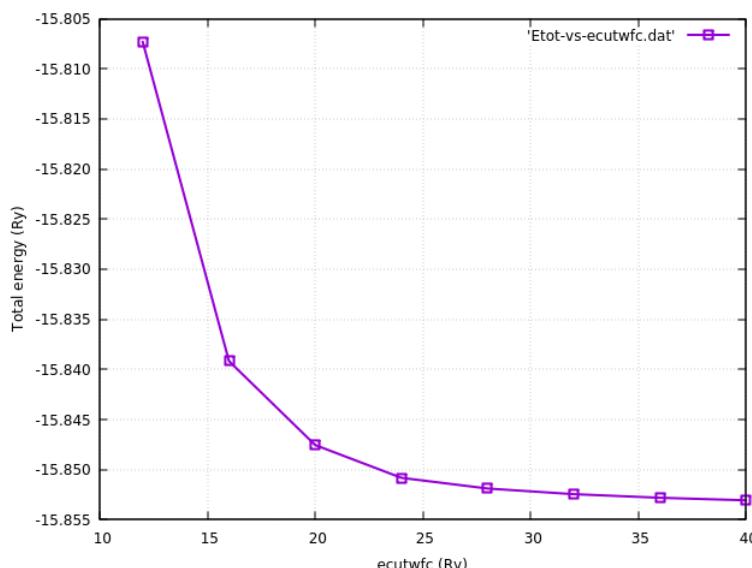


Figura 2.3: Barrido de $ecutwfc$ para Si bulk.

Ahora hacemos el barrido de la k -mesh. Consideramos el mismo criterio de convergencia. $E_{tot}(222) = -15,7945 \text{ Ry} \wedge E_{tot}(444) = -15,8073 \text{ Ry} \Rightarrow \Delta E_{tot} = 0,0128 \text{ Ry} = 12,8 \text{ mRy}$

$$E_{tot}(444) = -15,8073 \text{ Ry} \wedge E_{tot}(666) = -15,8076 \text{ Ry} \Rightarrow \Delta E_{tot} = 0,0003 \text{ Ry} = 0,3 \text{ mRy}$$

Se concluye entonces que el valor óptimo es 4 4 4.

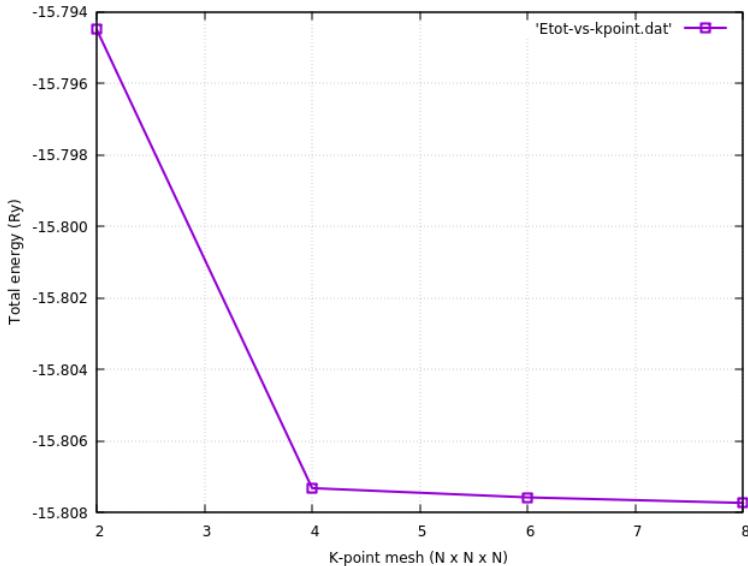


Figura 2.4: Barrido de k -mesh para Si bulk.

Observación

La convergencia de los puntos k no es necesariamente monotónica ya que no existe ningún principio variacional con respecto al número de puntos k .

Ahora analizamos el parámetro de red. El equilibrio en el caso del Si queda determinado por aquel parámetro de red que minimiza la energía ya que no hay fuerzas actuando sobre los átomos dada la simetría.

Observación

La ausencia de fuerzas debida a la simetría puede comprobarse ejecutando `tprnfor = .true.` en `&CONTROL`.

De la gráfica vemos que es 10,2 Bohr. Siendo más precisos, la ecuación de estado nos dice que es 10,2090 Bohr y el módulo de Young es 934 kbar.

A partir de los resultados de la ecuación de estado, notamos además que para parámetros de red menores al óptimo, la energía total es menor que la entalpía. Lo opuesto ocurre para parámetros de red mayores al óptimo.

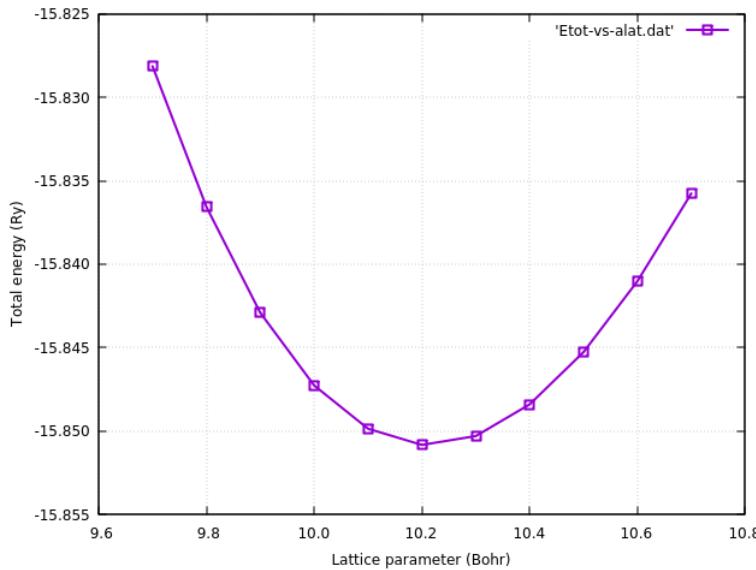


Figura 2.5: Barrido del parámetro de red para Si bulk.

```

# equation of state: murnaghan.          chisq =  0.4600D-09
# a0 = 10.2133 a.u., k0 = 932 kbar, dk0 = 4.06 d2k0 = 0.000 emin = -15.85080
# a0 = 5.40465 Ang, k0 = 93.3 GPa, V0 = 266.34 (a.u.)^3, V0 = 39.47 A^3
#####
# Lat.Par      E_calc      E_fit      E_diff      Pressure      Enthalpy
# a.u.        Ry         Ry         Ry         GPa         Ry
#####
9.70000   -15.82815   -15.82813   -0.00002    20.07    -15.51681
9.80000   -15.83657   -15.83660   0.00003     15.02    -15.59634
9.90000   -15.84290   -15.84291   0.00001     10.60    -15.66810
10.00000  -15.84726   -15.84726   0.00000      6.73    -15.73284
10.10000  -15.84985   -15.84983  -0.00001      3.34    -15.79131
10.20000  -15.85082   -15.85079  -0.00003      0.37    -15.84419
10.30000  -15.85029   -15.85027  -0.00002     -2.25    -15.89202
10.40000  -15.84838   -15.84840  0.00001     -4.55    -15.93532
10.50000  -15.84526   -15.84529  0.00003     -6.57    -15.97460
10.60000  -15.84102   -15.84104  0.00002     -8.36    -16.01027
10.70000  -15.83576   -15.83574  -0.00003    -9.94    -16.04272

```

Figura 2.6: Resultados del fiteo utilizando la ecuación de estado.

Para hacer el cálculo de bandas, no podemos usar el mesh corrido, sino los valores serán erróneos: no estaríamos incluyendo el punto Γ .

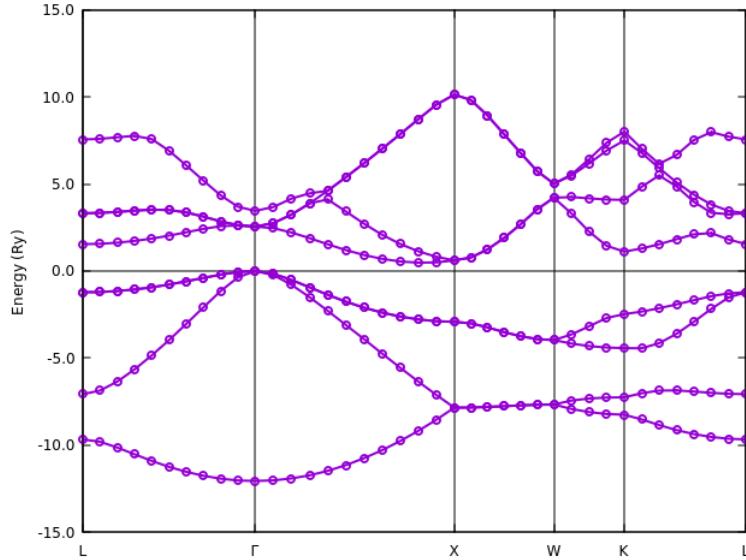


Figura 2.7: Estructura de bandas para Si bulk.

Al analizar las curvas de energía total en función del parámetro de red para diferentes k -mesh y $ecutwfc$ vemos que:

- Para un mismo $ecutwfc$, la energía no es tan sensible al k -mesh: con $2 \times 2 \times 2$ es muy mala la convergencia, pero ya con $4 \times 4 \times 4$ se estabiliza.
- A medida que aumentamos $ecutwfc$, las curvas se van desplazando hacia valores más negativos de energía.

Al comparar los k -mesh volvemos a elegir $4 \times 4 \times 4$. Al comparar los mínimos, vemos que la elección de 24 Ry fue acertada.

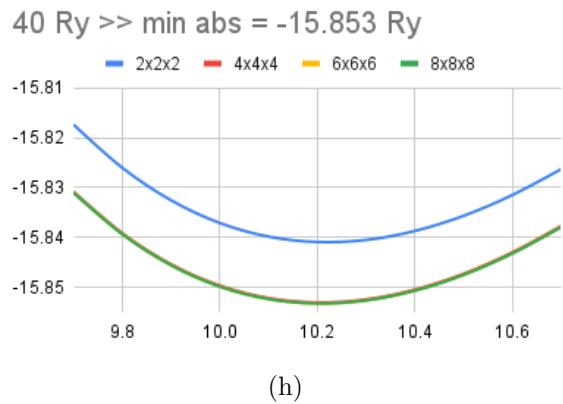
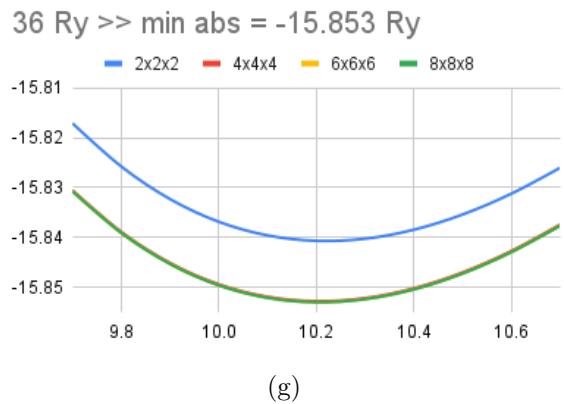
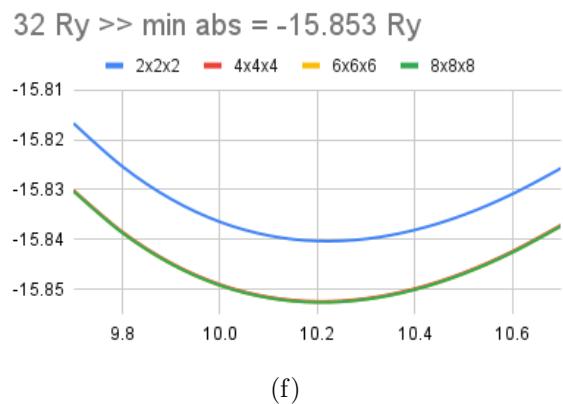
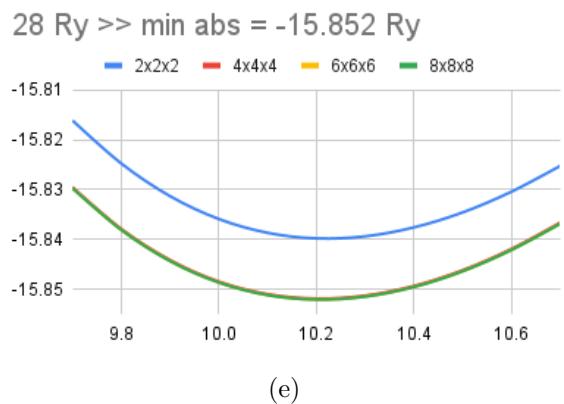
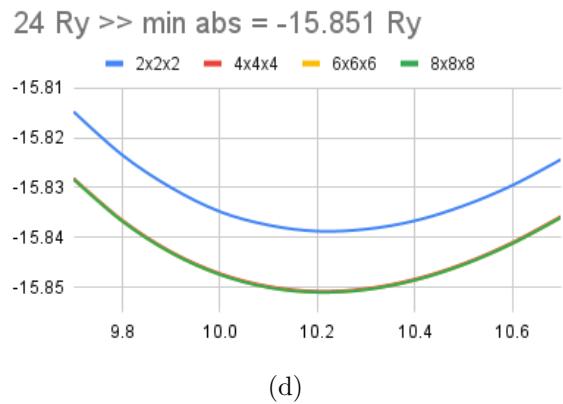
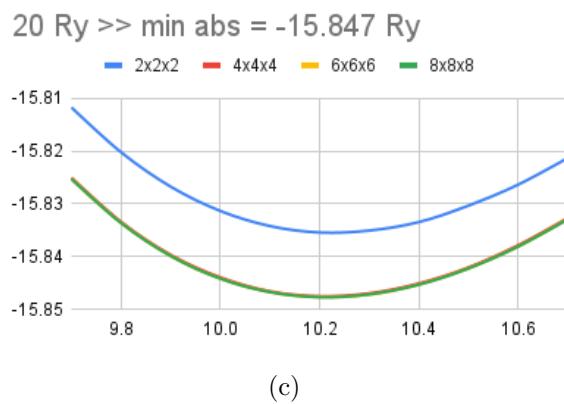
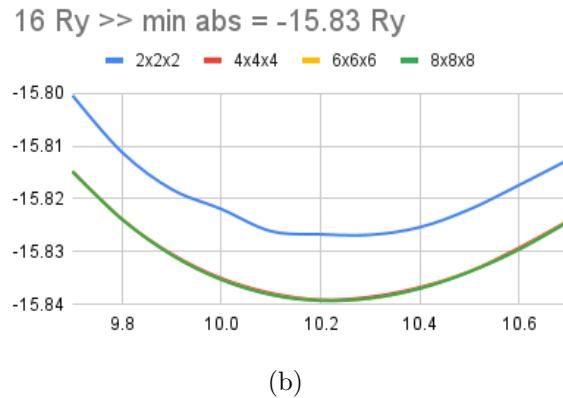
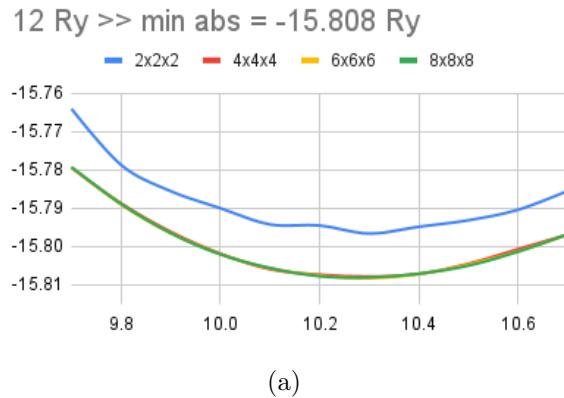


Figura 2.8: Energía en total en función del parámetro de red para diferentes cutoff y diferentes k -mesh.

2.3.2. Al

Objetivo

Cálculo de sistemas metálicos: Al bulk.

Se explora diferentes esquemas de smearing: cómo hacerlo y hasta dónde. Se plotea la densidad de carga electrónica.

Pasos

1. Explorar qué hace el smearing con función de ensanchamiento Gaussiana. Una vez que corra, editar el .pwtk y agregar más puntos de degauss, como 0.15 ó 0.2, dentro del foreach. Para evitar volver a correr lo ya corrido, encender el modo *restart* (descomenta `#restart true`).

```
pwtk degauss.pwtk
```

2. Calcular la densidad de carga (chdens).

- a) **1-chdens.pwtk:** Calcular la densidad de carga de valencia. Editar con las variables adecuadas antes de correr. Ver la densidad de carga con xcrysden.

```
pwtk 1-chdens.pwtk
```

- b) **2-chdens-paw.pwtk:** Calcular la densidad de carga all-electron tanto de valencia como total, mediante el uso de PAWPP. Ver en xcrysden.

```
pwtk 2-chdens-paw.pwtk
xcrysden --xsf all-electron-VALENCE-chdens.xsf -s state2.xcrysden
xcrysden --xsf all-electron-total-chdens.xsf -s state2.xcrysden
```

Resultados: Al

Aunque Al es más simple que Si ya que tiene sólo un átomo por celda dentro de una FCC, se trata de un metal: no será suficiente con sólo conocer las bandas de valecias y pocos puntos k.

El barrido de parámetros es tridimensional: se analizan los puntos k, el tipo de smearing y el valor propio del smearing que se utiliza.

Cuando usamos *g* (gaussian) vemos que la mesh 4x4x4 no es útil para Al como lo era para Si. A mayor smearing, las curvas convergen unas a otras. Con esto en mente, podríamos usar smearing más altos y una mesh con menos resolución. Cuando usamos *m-p* o *m-v* queda claro que la energía no depende tan fuertemente del smearing como en el caso anterior, permitiendo una convergencia más rápida y segura. En el caso del aluminio tenemos entonces que una buena convergencia podría alcanzarse usando *m-p* o *m-v* con un mesh 12x12x12 y un *degauss* entre 0.01 y 0.05 Ry.

El ensanchamiento no puede reducirse demasiado: los niveles energéticos deben tener cierta superposición (es un metal) o de lo contrario la ventaja de hacer smearing se terminaría perdiendo. En otras palabras: valores de degauss muy altos empiezan a romper todo.

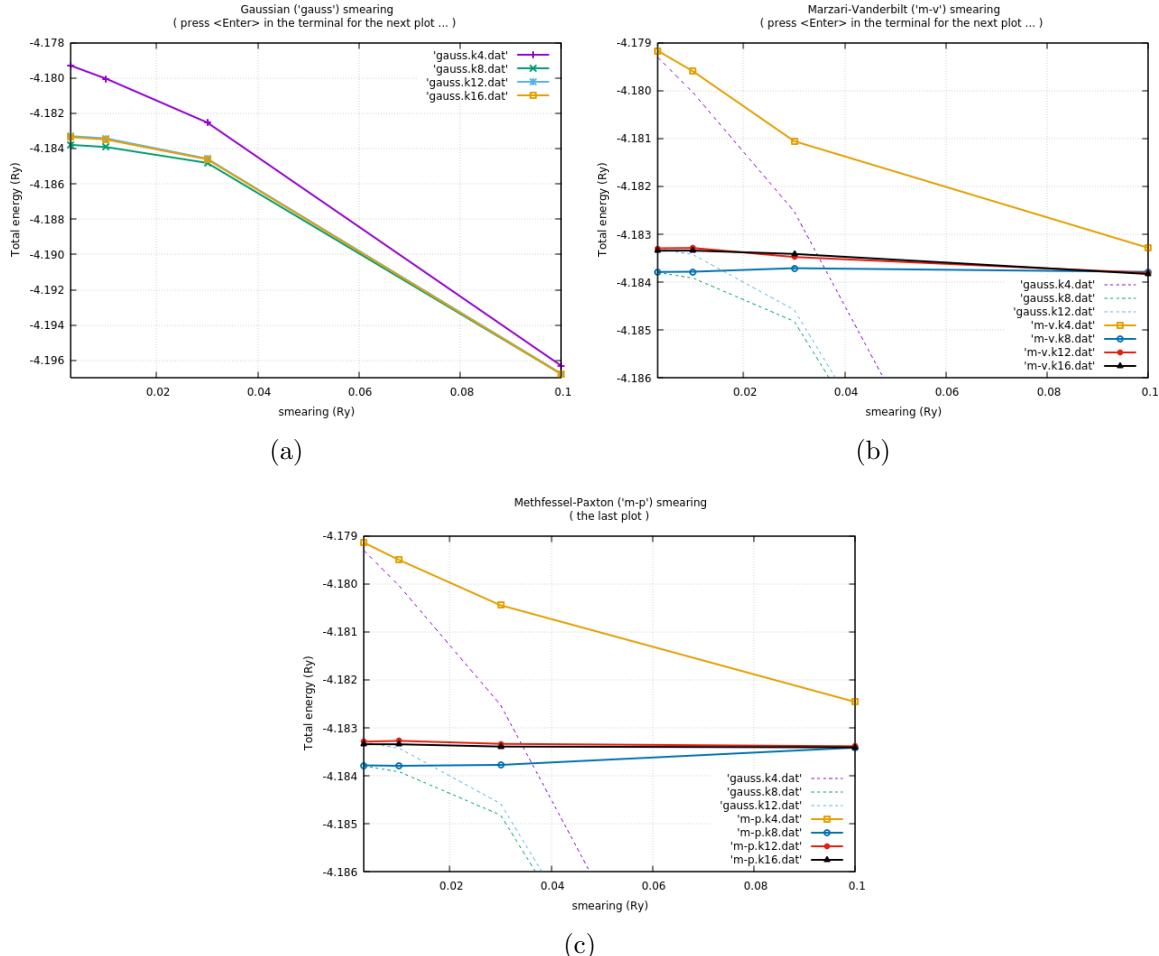


Figura 2.9: Energía total en función del smearing y el ancho de smearing utilizado, variando la cantidad de puntos K. Las ocupaciones son: g (a), m-v (b) y m-p (c).

Dejando de lado los test de convergencia, vamos a ver cómo hacer un post-processing, determinando la densidad de carga. La calculamos con dos potenciales diferentes:

- Un PP: Al.pz-vbc.UPF
- Un PAWPP para hacer un all-electron: Al.pbe-n-kjpaw-psl.1.0.0.UPF

Las corridas con all-electron requieren cutoff gigantes (ecutrho 1000 Ry). A partir de las figuras vemos que usando PP no hay electrones en torno a los núcleos, sino sólo en zonas intersticiales. Esto es por la manera en la que justamente construimos los PP. Para el caso del all-electron, se grafica valencia y full según el input del pp.x que le demos (*plot_num*). En este caso sí se aprecia carga en torno al núcleo: es totalmente brillante debido a la alta concentración y localización de los electrones del core.

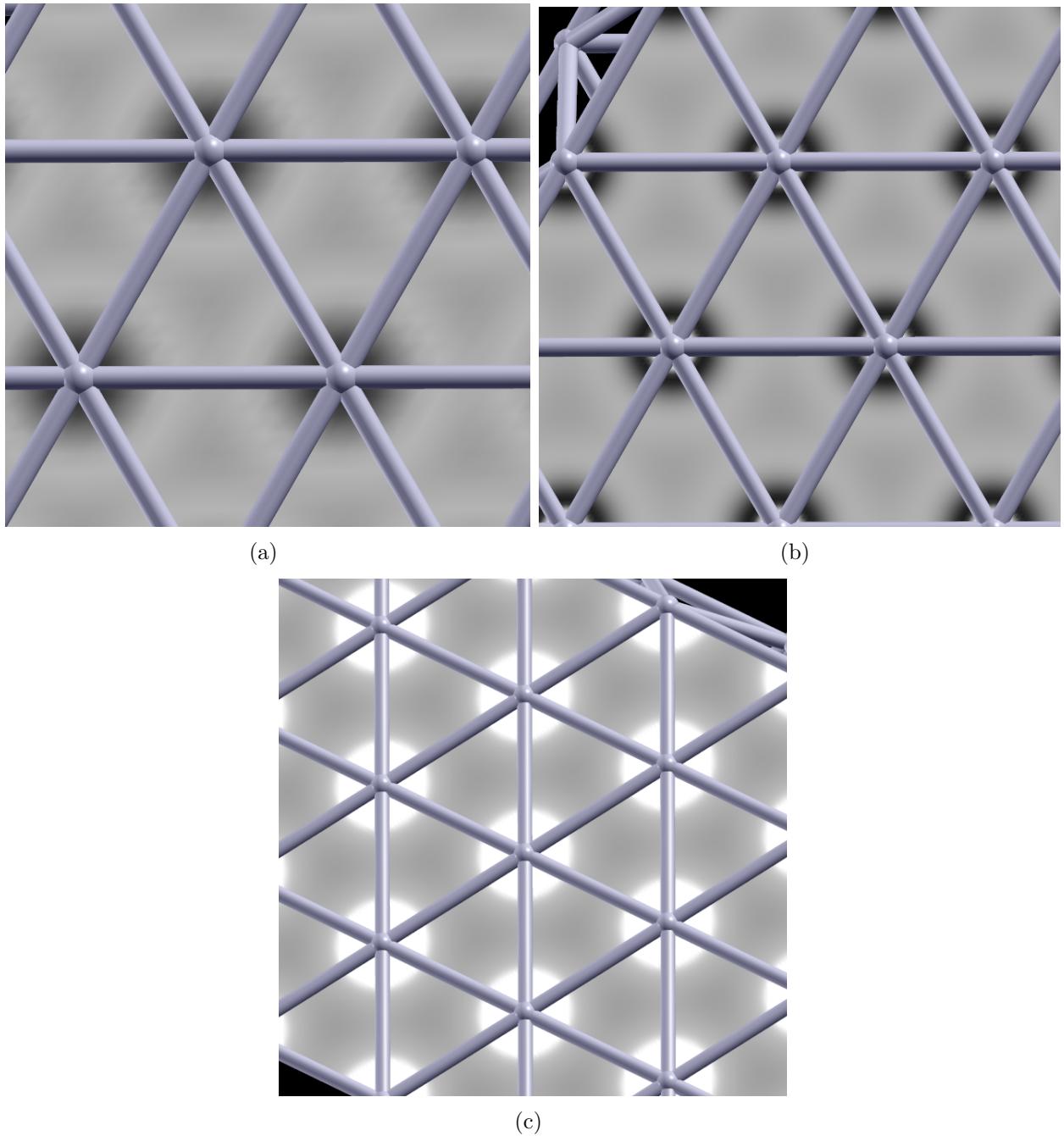


Figura 2.10: Densidad de carga obtenida a partir de dos potenciales diferentes. Negro implica que no hay cargas y cuanto más brillante, mas carga hay. En (a) se usa el PP mientras que en los otros se usan el all-electron: (b) sólo valencia y (c) todo.

2.3.3. Fe

Objetivo

Realizar cálculos con sistemas magnéticos (bulk de Fe tanto ferromagnético como antiferromagnético). Aprender criterios de convergencias para USPP: el parámetro dual. Plotear DOS total y proyectada (PDOS) sobre los orbitales s y d.

Pasos

Las diferencias entre los códigos ferromagnetico y antiferromagnetico pueden verse con

```
diff pw.fe\_fm.scf.in pw.fe\_afm.scf.in
```

Los cálculos se pueden correr con

```
pw.x < pw.fe\_fm.scf.in > pw.fe\_fm.scf.out
pw.x < pw.fe\_afm.scf.in > pw.fe\_afm.scf.out
```

Analizar los outputs, prestando atención a los valores total/absolute magnetization en ambos casos.

1. Realizar un test de convergencia específico para USPP.

```
pwtk ecut.pwtk
```

2. Calcular y plotear DOS y PDOS sobre OA s y d. Antes de correr, ingresar los valore adecuados de las variables necesarias. Luego poner el valor correcto de la energía de Fermi en plot(gp para después hacer el plot.

```
pwtk dos.pwtk
grep 'Fermi energy' pw.Fe.nscf.out
gnuplot plot.gp
```

Resultados: Fe

En la siguiente figura se aprecia la diferencia entre ambos archivos, siendo primero el ferromagnetico y el segundo el antiferromagnetico.

```
10c10
.     ibrav = 3,
< .
.     ibrav = 1,
12,13c12,13
<     nat = 1,
<     ntyp = 1,
---
>     nat = 2,
>     ntyp = 2,
22c22,23
<     starting_magnetization(1) = 0.6
---
>     starting_magnetization(1) = 0.6
>     starting_magnetization(2) = -0.6
31c32,33
<   Fe  1.0  Fe.pbe-nd-rrkjus.UPF
---
>   Fe1  1.0  Fe.pbe-nd-rrkjus.UPF
>   Fe2  1.0  Fe.pbe-nd-rrkjus.UPF
34c36,37
<   Fe 0.0  0.0  0.0
---
>   Fe1 0.0  0.0  0.0
>   Fe2 0.5  0.5  0.5
```

Como vamos a usar un USPP (Fe.pbe-nd-rrkjus.UPF) debemos analizar la convergencia del *ecutrho* también: tener *dual* = 4 no será suficiente.

$$dual = \frac{ecutrho}{ecutwfc}$$

Observación

Se puede estudiar también la variación del resultado respecto a la CI de la magnetización.

Al correr ambos programas vemos que para el caso ferromagnético la magnetización total es de 2.29 Bohr mag/cell y la absoluta es de 2.45 Bohr mag/cell, mientras que para el caso antiferromagnético la total es 0 y la absoluta es 4.13 Bohr mag/cell.

Ahora vamos a analizar la convergencia de los cutoffs con el caso ferromagnético. Para ello se hace un barrido de *dual* y, para cada uno de ellos, se prueban diferentes *ecutwfc*. Vemos que para un *dual* de 4 tenemos una curva, pero para los duales 8 y 12 las curvas coinciden. Si queremos usar el *dual* de 4 debemos usar *ecutwfc* > 40; en cambio con los otros duales podemos usar *ecutwfc* = 25. En el código se hacen muchísimas más operaciones con las funciones de onda que con las densidades de carga, por lo que si usamos un *ecutwfc* más chico el cálculo irá más rápido.

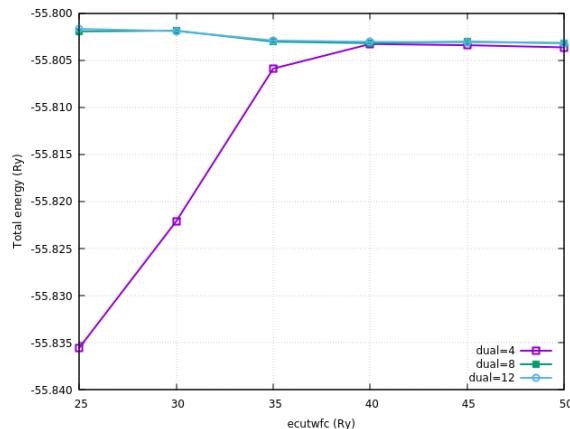


Figura 2.11: Variación de la energía total en función del *dual* para Fe ferromagnético.

Ahora vamos a graficar DOS y PDOS, también para el caso ferromagnético. Recordamos que necesitamos una *k*-mesh más densa. Además *occupations* ='tetrahedra' suele ser útil para este tipo de cálculos. En la DOS vemos que hay más estados ocupados con spin up que down. En la PDOS vemos que

- Las bandas s son muy anchas y bajas: los 2 electrones por átomo en el orbital s ocupan un rango muy grande de energía (15 eV).
- Las bandas d son más picudas: tenemos 10 electrones por átomo en un rango energético de 5 eV.

En realidad falta considerar la energía de Fermi. A partir de la misma, vemos que sólo hay 1 electrón en el s y 7 en el d. Esto también explica el gran pico de spin down a valores tan grandes de energía: estamos muy por encima de la energía de Fermi (antiorbitales).

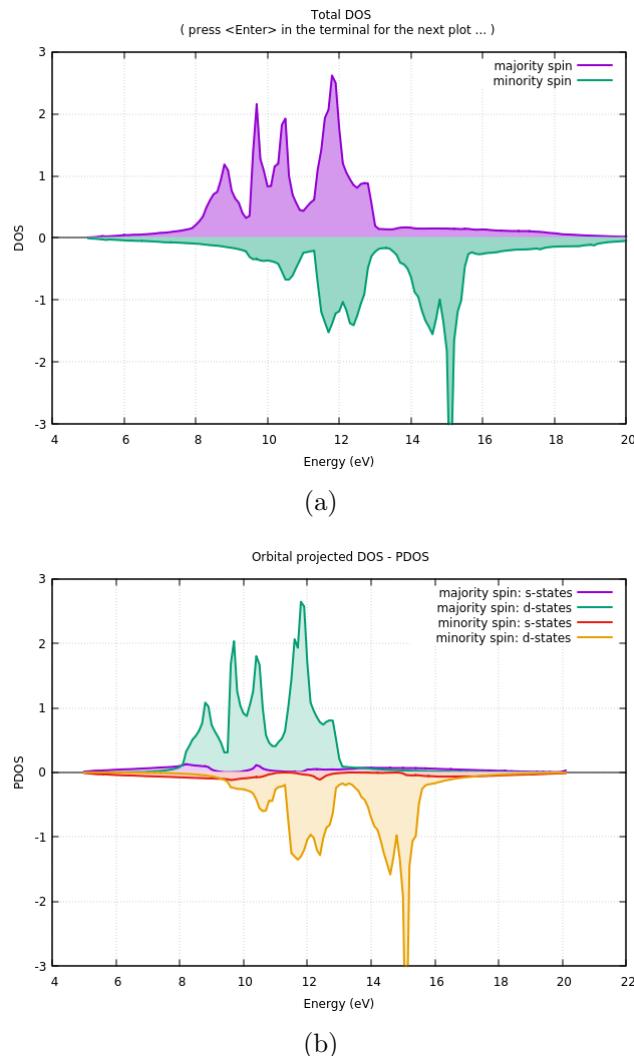


Figura 2.12: DOS y PDOS sobre los OA s y d para el Fe ferromagnético. Majority y minority spin corresponden a spin up y down, respectivamente.

Capítulo 3

Día 3

3.1. Teórico 1/2

Topic: Forces, stresses, geometry optimisation.

Speaker: Pietro DELUGAS (SISSA, Italy).

3.1.1. Parámetros estructurales

Un sistema queda definido a partir del número y tipo de iones, donde el tipo queda descripto a partir del PP, junto a las posiciones que ocupa cada uno, determinando la estructura.

La estructura de un sistema periódico queda descripta por:

- Los parámetros de red: \vec{a}_1 , \vec{a}_2 y \vec{a}_3 .
- Las posiciones atómicas: \vec{r}_I

Como la modificación de los parámetros de red alteran las posiciones atómicas, se utilizan coordenadas cristalinas fraccionarias, donde la posición atómica queda expresada a partir de los parámetros de red.

$$\vec{r}_I = \sum_{i=1}^3 x_i \vec{a}_i \quad ; \quad 0 \leq x_i \leq 1$$

La energía potencial $E(\vec{r}_I, \vec{a}_i)$ termina siendo la PES (Potential Energy Surface) del sistema bajo la aproximación de Born-Oppenheimer: esta es la función a minimizar.

3.1.2. Fuerzas

La PES debe ser derivable ya que la fuerza es el opuesto de la derivada de las interacciones interelectrónicas e internucleares respecto a las posiciones.

$$F_I = -F_I^{el} - \frac{e^2}{2} \frac{\partial}{\partial \vec{r}_I} \sum_{I \neq J} \frac{Z_I Z_J}{\|\vec{r}_I - \vec{r}_J\|}$$

El teorema de Hellmann-Feynman puede ser aplicado en DFT. El mismo establece que la derivada de la energía total E respecto a un parámetro continuo λ es igual al valor de expectación de la derivada del Hamiltoniano respecto a dicho parámetro. Así, una vez que la distribución espacial de los electrones ha sido determinada al resolver la ecuación de Schrödinger, todas las fuerzas del sistema puede usarse recurriendo a la electrostática clásica.

$$\frac{dE}{d\lambda} = \langle \Psi | dH/d\lambda | \Psi \rangle$$

De este modo, la contribución electrónica F_I^{el} puede determinarse como el valor de expectación de las derivadas del potencial externo aplicado.

$$F_I^{el} = -\frac{\partial E}{\partial \vec{r}_I} = -\sum_v f_v \langle \psi_v | \partial V / \partial \vec{r}_I | \psi_v \rangle = -\int n(\vec{r}) \frac{\partial V}{\partial \vec{r}_I} d\vec{r}$$

La idea de calcular las fuerzas como un valor de expectación es válida sólo si estamos realmente en convergencia: la precisión de las fuerzas dependerá de que estemos cerca de la autoconsistencia. Esto se debe a que en realidad tenemos un segundo término al derivar la energía, el cual se anula cuando estamos en un mínimo:

$$\frac{\partial E}{\partial \vec{r}_I} = \int n(\vec{r}) \frac{\partial V}{\partial \vec{r}_I} d\vec{r} + \int \frac{\delta E}{\delta n(\vec{r})} \frac{\partial n(\vec{r})}{\partial \vec{r}_I} d\vec{r}$$

En QE, pw.x imprime un estimado de este segundo término al final de la autoconsistencia que puede servir como una corrección aproximada. Durante una relajación o una dinámica debemos monitorear estos valores: deben ser algún orden de magnitud menor que las fuerzas calculadas. De lo contrario el programa emite una advertencia.

3.1.3. Tensión y estrés

Dijimos que cambiar los valores de los parámetros de red implica una deformación uniforme del sistema. Esta queda descripta por el tensor de tensión $e_{\alpha\beta}$. La transformación es lineal

$$\vec{r}'_\alpha = \sum_\beta (\delta_{\alpha\beta} + e_{\alpha\beta}) \vec{r}_\beta$$

El tensor de estrés es la derivada de la energía por unidad de volumen con respecto a la tensión.

$$\sigma_{\alpha\beta} = -\frac{1}{\Omega} \frac{\partial E}{\partial e_{\alpha\beta}}$$

Sólo la contribución simétrica del tensor $e_{\alpha\beta}$ da lugar a una deformación efectiva, afectando la energía, ya que la contribución antisimétrica es simplemente una rotación.

La traza del tensor de estrés está relacionada a la presión según

$$P = \frac{1}{3} \sum_\alpha \sigma_{\alpha\alpha}$$

mientras que las componentes no diagonales son el estrés de corte (deformaciones no paralelas a los planos de la celda unidad).

Como el estrés es una primera derivada de un parámetro externo, el teorema de Hellmann-Feynman puede aplicarse. En este caso no depende explícitamente de las posiciones iónicas, sino que es la derivada de la energía cinética, la de Hartree y la de XC con respecto a una deformación uniforme. Esto lleva a que involucra las componentes de las funciones de onda, las cuales dependen a su vez de la base de PWs usada.

Smooth cutoff

Evaluar la energía y el estrés puede dar discontinuidades si el cutoff no fue convergente, teniendo que usar valores más grandes. Una alternativa es usar smooth cutoff: creamos una región donde las PW entran y salen del conjunto base con regularidad, teniéndose en cuenta cuando se evalúa el estrés y la energía.

Esto se logra modificando el funcional de energía cinética por lo que las PW de borde se vuelven caras.

$$T(\vec{G}) = \underbrace{\|\vec{G}\|^2}_{\text{original}} + Q_{cut} \underbrace{\left[1 + erf \left(\frac{\|\vec{G}\|^2 - E_{cfixed}}{\sigma} \right) \right]}_{\text{smooth}}$$

Cuando $\vec{G}^2 - E_{cfixed} < 0 \wedge |\vec{G}^2 - E_{cfixed}| \ll \sigma$ la función error tiende a -1 , por lo que $T(\vec{G}) \rightarrow \|\vec{G}\|^2$. A medida que \vec{G}^2 empieza a haber contribuciones no nulas. Sin embargo, la transición es suave y no brusca.

En QE tengo las siguientes variables del input en pw.x (Fig. 3.1):

- *ecfixed* → energía por encima de la cual las PWs alcanzan el máximo costo.
- *qesigma* → ancho a partir del cual se empieza a sumar este costo extra.
- *qcutz* → valor del costo extra de energía.

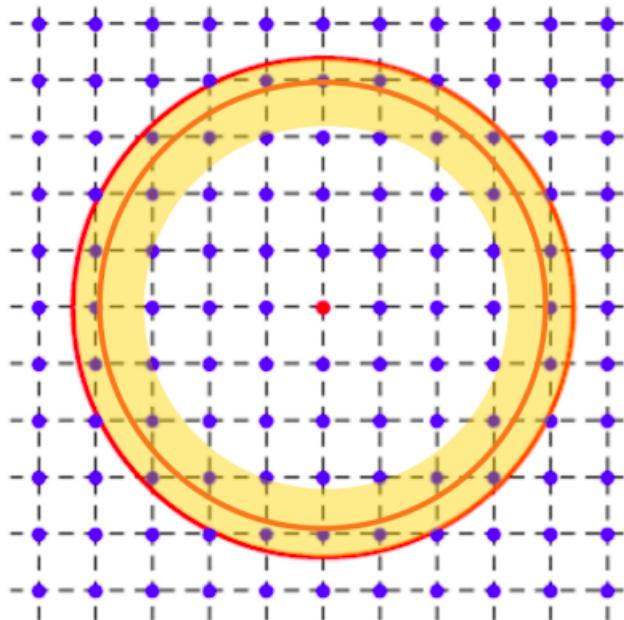


Figura 3.1: Cómo afectan las variables del smooth cutoff en la grilla sobre el G -espacio: *ecfixed* es el círculo naranja, *qesigma* es el anillo amarillo y *ecutwfc* es el círculo rojo.

Tenemos entonces que el smooth cutoff nos permite hacer una evaluación más regular del estrés y, además, no necesitamos aumentar *ecutwfc*.

3.1.4. Métodos de optimización estructural

En pw.x se pueden hacer dos tipos de optimizaciones estructurales:

1. Manteniendo fijo el tamaño de celda: *calculation* = 'relax'. Se dan las posiciones atómicas y las mismas cambian según las fuerzas.
2. Con tamaño de celda variable: *calculation* = 'vc-relax'. Se dan las posiciones atómicas y el tensor de tensión de la celda y las mismas cambian según las fuerzas y el tensor de estrés.

Se tienen 2 algoritmos para para relajar:

1. **BFGS:** es un algoritmo quasi-Newtoniano. Es el que se hace por defecto: $ion_{dynamic} = 'bfgs'$ y $cell_{dynamics} = 'bfgs'$
2. **Quickmin:** es una dinámica de Verlet amortiguada. Debo decirle $ion_{dynamic} = 'damp'$ y $cell_{dynamics} = 'damp - pr'$ o $cell_{dynamics} = 'damp - w'$

Quickmin

El mecanismo de amortiguación extrae energía cinética del sistema a medida que el sistema va cayendo al mínimo. Este algoritmo elimina cualquier componente de velocidad generalizada cuya dirección se oponga a la componente conjugada de la fuerza generalizada.

Aunque el algoritmo es robusto ya que funciona incluso estando lejos del mínimo, requiere cierta experiencia con MD para establecer el time step. Suele ser más lento que BFGS: se suele usar primero un BFGS y luego el quickmin.

BFGS

Es un algoritmo muy útil tanto para celda fija como para celda variable. En las proximidades de un punto de equilibrio \vec{r}^{eq} ($\nabla E(\vec{r}^{eq}) \approx 0$) se asume una forma cuadrática para $E(\vec{r})$ tomando la Hessiana \mathcal{H} .

$$E(\vec{r}) \approx E(\vec{r}^{eq}) + \frac{\vec{r} - \vec{r}^{eq}}{2} \mathcal{H}(\vec{r} - \vec{r}^{eq})$$

Dados dos puntos \vec{r}_1 y \vec{r}_2 y sus respectivos gradientes \vec{g}_1 y \vec{g}_2 , se tiene queda $\vec{g}_2 - \vec{g}_1 = \mathcal{H}(\vec{r}_2 - \vec{r}_1)$. Luego $\vec{g}_2 = 0$ si $\vec{r}_2 = \vec{r}_1 - \mathcal{H}^{-1}\vec{g}_1$ (Paso de Newton-Taphson).

En la práctica se tiene una secuencia de cálculos en las posiciones \vec{r}_i .

$$\vec{r}_{i+1} = \vec{r}_i + T_k^L \frac{\vec{s}_k^{NR}}{\|\vec{s}_k^{NR}\|} \quad ; \quad \vec{s}_k^{NR} = -\mathcal{H}_k^{-1} \vec{g}_k$$

donde T_k^L se conoce como radio de confianza el cual determina cuánto nos vemos en la dirección dictada por \vec{s}_k^{NR} .

Para el paso siguiente debemos actualizar la Hessiana de una manera particular la cual involucra diferencias entre gradientes.

En cada uno de los pasos también se actualiza T_k^L de manera tal que satisfaga las dos condiciones de Wolfe:

1. **Decrecimiento suficiente:** la nueva energía debe ser menor que cierto valor, sino T_k^L debe acortarse.

$$E_{new} \leq E_{old} + \omega_1 T_k^L \vec{g} \cdot \vec{s}_k$$

2. **Curvatura:** si la nueva pendiente es muy alta, el T_k^L es aumentado.

$$\vec{g}_{new} \cdot \vec{s}_k \geq \omega_2 \vec{g}_{old} \cdot \vec{s}_k$$

Los valores ω_1 y ω_2 se pueden fijar en el input. De lo contrario, el programa usa los valores por default.

Características de la optimización estructural

- Permite encontrar únicamente el mínimo más cercano, ya que no tiene la posibilidad de sortear barreras de activación.
- En principio no rompe la simetría del cristal, salvo errores por ruido numérico. Esto permite encontrar mínimos atados a la simetría dada puesto que puede haber diferentes mínimos asociadas a distintas simetrías.
- Durante una vc-relax el conjunto de PWs se mantiene constante a pesar de que los parámetros de red van cambiando. Esto hace que el resultado final no sea exactamente igual a lo que uno obtendría al comenzar la corrida de cero con el mismo cutoff ya que ambas bases no son iguales. Una vez obtenidos los vectores de red optimizados, el programa recalculará la energía con la base asociada a estos nuevos valores.
- Utiliza tanto las energías como las fuerzas para ubicar el mínimo. Si la SCF no fue convergente, el algoritmo no será convergente o dará lugar a errores. El error en las fuerzas es lineal, mientras que el error en las energías es cuadrático.

3.1.5. Born-Oppenheimer MD

Asumiendo un comportamiento clásico para los núcleos y los electrones en el estado fundamental, recurrimos a un Lagrangiano para describir el movimiento nuclear. Las ecuaciones de movimiento son entonces las ecuaciones de Newton usuales, las cuales pueden discretizarse e integrarse. Esto es lo que se conoce como MD *sobre la superficie BO* donde los electrones se encuentran siempre en su estado fundamental instantáneo.

El cálculo puede hacerse recurriendo a diversos algoritmos como Verlet o velocity Verlet.

Algunas cosas a tener en cuenta son:

- El time step debe ser tan grande como sea posible, pero lo suficientemente pequeño como para poder seguir el movimiento nuclear con precisión. Se tiene que

$$\Delta t \approx 0,01 - 0,1\Delta t_{max} \quad ; \quad \Delta t_{max} = \frac{1}{\omega_{max}}$$

donde ω_{max} es la frecuencia del modo vibracional (fonón) más rápido que tengamos en el sistema.

- Las fuerzas deben ser lo suficientemente precisas como para que la energía se conserve paso a paso. De lo contrario, se observa un drift energético.
- Una BOMD es computacionalmente cara (No me la counter strike). A veces conviene hacer CPMD.

3.2. Q&A 1/2

Why keep the previous grid during the optimization cycle? Since the meat of the process is: scf -z optimize -z scf -z... -z convergence ; why does one not consider the new lattice vectors? Why not adjust the cutoff accordingly at every step as it is done in the very last step of the optimization (I think), why keep the ellipse instead of the circle for the entire optimization? Is it a convergence issue?

or computational efficiency. reallocating everything is expensive

Does the penalty for high G components you were mentioning (the one with the erf) alleviate these effects or is it unrelated? I have not understood where this applies. I have not used the related flags so far in my calculations, in what cases should I bother with these parameters?

yes in principle any G in the buffer zone has negligible weight in the computation but any G in there enters smoothly in the computation if its lenght become shorter and the stress derivative of the kinetic energy functional takes into account the change in energy of the Gs in the buffer zone

If local minima are an issue can you give a bigger kick to the system in order for it not to get trapped there (something analogous to what happens in ML algorithm since you are in a very similar optimization problem optimized by gradient descent)? Does it make sense to start a calculation with a big kick, see where the problem starts oscillating and then reduce the kick around that point with another calculation (or even do this automatically) ?

yes with complicated landscapes there are many methods not implemented directly in QE but that use QE as a force engine

When I do the relax calculation, in the .out file there is the information about the forces. What kind of information does this total force and DFT-D3 dispersion force give us?

the one is just the amount of Forces due to DFT-D3 that is a post-DFT correction to forces. If you see there is also the warning about the fact that the dft forces have a magnitude comparable with the force. They are both very small though you should be close to the minimum

In the relax calculation, what does the 'upscale' parameter do to the ions?

Nothing: it sets a maximum value for scf threshold tightening, avoiding to end up with too small scf thresholds and never get to convergence

In which cases I would choose Ion-dynamics instead bfgs?

In cases where bfgs doesn't converge it could be useful

there is any way to get a good trade-off between the energy convergence thresh. and the force conv thresh? A general rule

Usually its a good idea to stick to the defaults, if you need to relax the system more you can tighten the criteria in a follow up calculation

is there a way to estimate how much memory I will use for the process? My question is because I usually use a cluster and I need to give these information before run a code.

There is a memory estimate written in the output of QE

Thinking in the Gamma point trick here, to reduce the expensive cost of optimization, should we use the converged k points or could we assume in general using less points its not going to affect strongly the optimized coordinates. (then use more kpoints in next calculations)

Ideally, we use the converged k points. But, as we use a supercell we reduce the k-point accordingly.

why at this example in vc-relaxation, you use 6 6 4 k points, not symmetric?

As we use a hexagonal cell, $a=b \neq c$, so having the same k points grid wont be accurate. The value of the k points should be inverse proportional to the real space distance

Why does it use 'm-p' for Zn? If I do not know, would I do the study of convergence?

M-P is one of the best smearing schemes for metallic systems. It could gives very serious problems for insulators.

What is the definition of force convergence threshold? Does it mean that the force of each atom less than the threshold?

Yes. Default is 1e-3 Ry/Bohr.

3.3. Teórico 2/2

Topic: Chasing saddle points: the NEB method.

Speaker: Anton KOKALJ (Jožef Stefan Institute, Slovenia).

3.3.1. Procesos activados

Se conoce como proceso activado a cualquier proceso caracterizado por barreras de activación, las cuales quedan determinadas por los saddle. Dada una PES se tienen infinitos caminos de reacción que conectan el estado inicial con el final. Se conoce como MEP (Minimum Energy Path) al camino de reacción más probable a 0 K. Cuando recorremos el MEP, el saddle determina el estado de transición (Fig. 3.2).

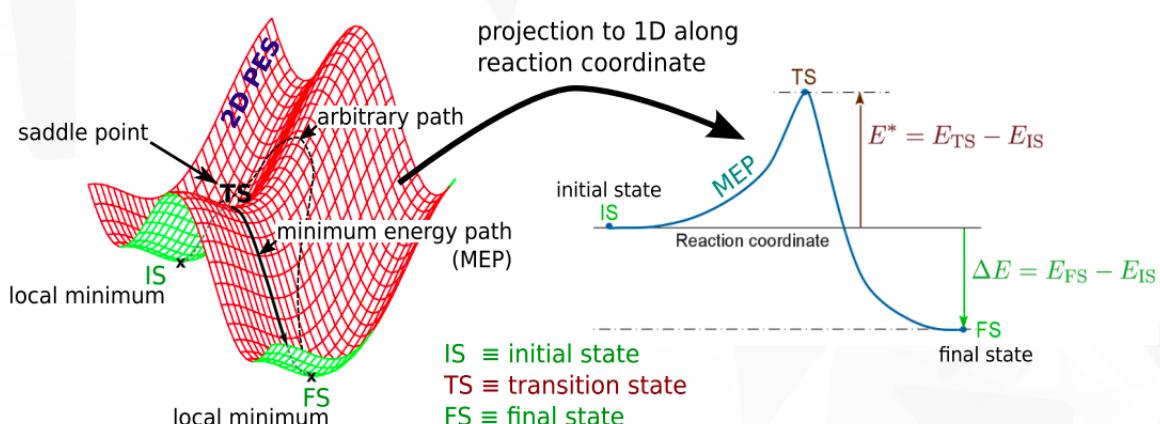


Figura 3.2: Proyección bidimensional de una PES marcando sus puntos más importantes. Tomando el MEP se puede hacer una proyección unidimensional.

La importancia de la energía de activación radica en que es un muy buen criterio para decidir si cierto proceso activado es cinéticamente posible a cierta temperatura y, de serlo, qué tan rápido es. La constante de velocidad es

$$k = \nu \exp(-E^* \beta) \quad ; \quad \nu = \frac{\prod_{j=1}^{3N} \nu_j^{IS}}{\prod_{j=1}^{3N-1} \nu_j^{TS}}$$

Aunque típicamente $\nu \approx 10^{13} s^{-1}$, en realidad depende del tipo de proceso: para la desorción suele ser $\nu \approx 10^{16} s^{-1}$ por ejemplo (ver Surf. Sci. Rep. 12, 183-242 (1991)).

En el contexto del análisis de procesos activados, se conoce como *imagen* a una dada configuración del sistema completo. Vamos a usar además un vector \vec{R} que representa en realidad al vector $3N$ dimensional con todas las coordenadas de los N núcleos. Así $\vec{R}^{(n)}$ es el vector posición de los N núcleos en la n -ésima iteración.

3.3.2. NEB

El estado inicial y el final son mínimos, por lo que deben encontrarse mediante métodos de optimización. Por otro lado, encontrar el saddle es bastante más complicado. Un método para encontrarle es el NEB (Nudged Elastic Band). La idea es la siguiente (Fig. 3.3):

1. Conectamos el estado inicial y el final con una *banda elástica*, la cual es tan elástica como sea necesario, *i.e.* no se puede cortar nunca.
2. Relajamos la bandita mediante fuerzas ortogonales (\vec{F}_\perp) a la misma.
3. Eventualmente se alcanzará $\vec{F}_\perp = 0$. La bandita resultante marca el MEP y determina el saddle.

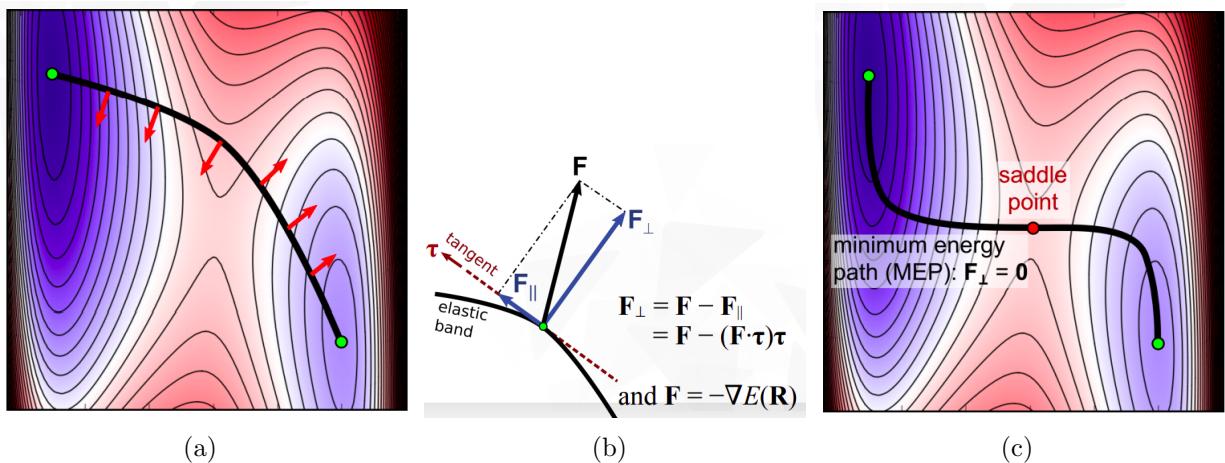


Figura 3.3: (a) Cómo conectamos el estado inicial con el final en NEB: la línea negra representa la banda elástica, mientras que las flechas rojas son las contribuciones ortogonales de la fuerza. (b) Descomposición de la fuerza aplicada sobre la bandita. (c) Resultado final donde $\vec{F}_\perp = 0$: la bandita marca el MEP.

Computacionalmente necesitamos discretizar la bandita elástica en una cantidad finita de imágenes: la fuerza perpendicular es calculada sobre estas imágenes. El problema de la discretización es que no preserva la distancia entre imágenes, lo cual puede dar lugar a distintos problemas a medida que avanzan las iteraciones. Para solucionar este problema, las imágenes son conectadas mediante resortes (Fig. 3.4), los cuales sólo actúan a lo largo del camino de reacción para mantener la distancia inter-imágenes.

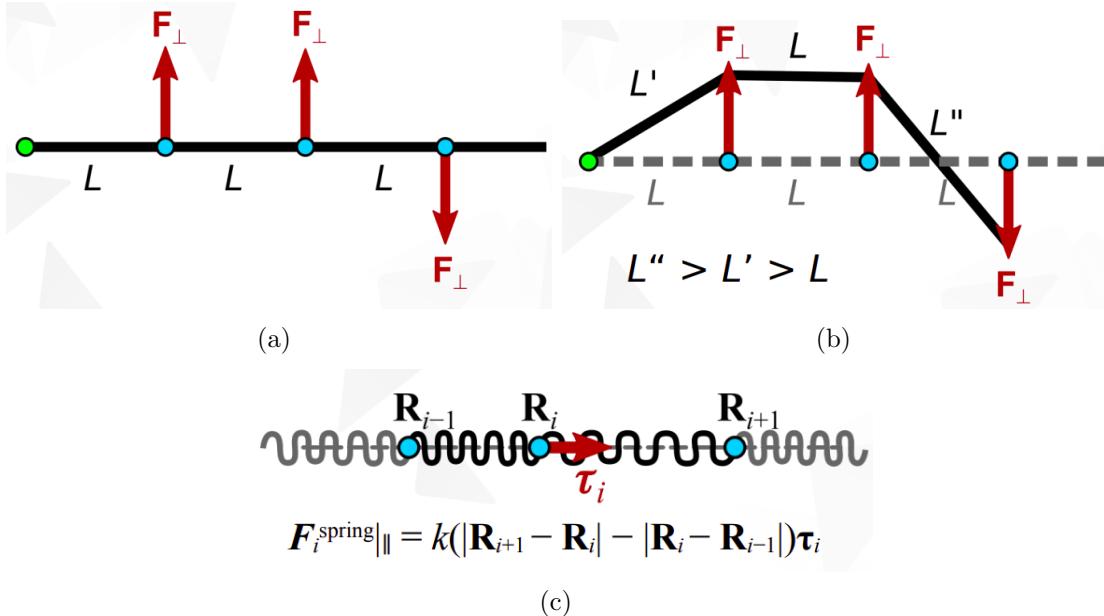


Figura 3.4: Al discretizar (a) las fuerzas pueden provocar que el camino entre imágenes no se conserve (b). Esto se soluciona utilizando resortes (c).

Reuniendo toda la información nos queda entonces que NEB consiste en (Fig. 3.5):

1. Conectar dos mínimos con un camino de reacción. Esto debe proponerse, pudiendo ser simplemente una interpolación lineal (default).
2. Discretizar el camino de reacción mediante imágenes ($\{\vec{R}_j\}_{j=1}^M$).
3. Conectar las imágenes con resortes.
4. Minimizar el camino de reacción mediante NEB: sobre la j -ésima imagen se aplica una fuerza perpendicular a la banda, moviéndola, y una tangencial que es la del resorte. Estos son los *empujones* (nudgings) que dan lugar al nombre.

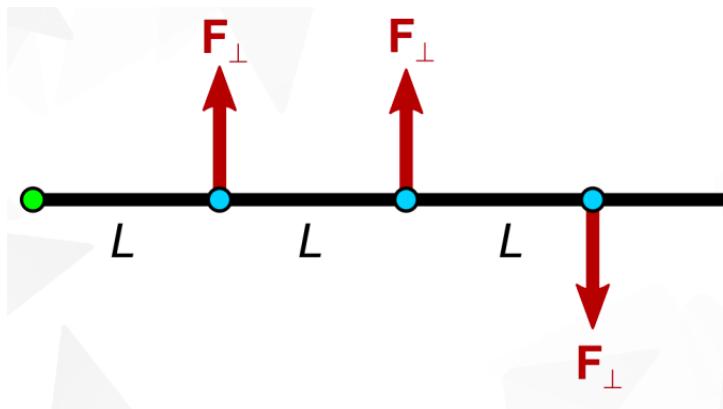


Figura 3.5: NEB considerando discretización y resortes.

La elección de la tangente es crucial para la convergencia. Una buena elección es considerar una tangente local, la cual sólo considera la imagen adyacente de mayor energía. En caso de que la imagen tenga menor energía que ambas imágenes adyacentes se toma un promedio.

Como el interés está centrado en conocer el saddle, debemos aumentar la resolución de la zona. Para ello podemos usar constantes del resorte variables entre dos valores dados. Cuanto mayor sea la energía de la imagen, más rígido debe ser el resorte. Así la densidad de imágenes en la proximidad del saddle será mayor.

En QE se pueden utilizar imágenes intermedias que permiten dar una guía para el camino de reacción inicial (Fig. 3.6). Muy útil cuando se tiene una intuición de la situación. Sólo son una guía, pero no se toman como imágenes para el cálculo.

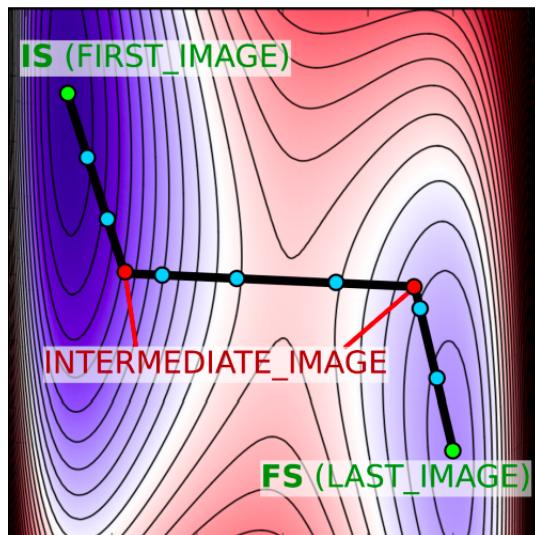


Figura 3.6: Uso de imágenes intermedias.

CI-NEB

Incluso cuando se tienen muchas imágenes y constantes de resorte variables, puede ocurrir que ninguna imagen quede lo suficientemente cerca del estado del saddle. Una solución al problema es recurrir al Climbing Image NEB (CI-NEB), donde se le permite a la imagen de mayor energía *escalar* al desacoplarle los resortes. Para escalar necesitamos invertir la fuerza tangencial sobre la banda.

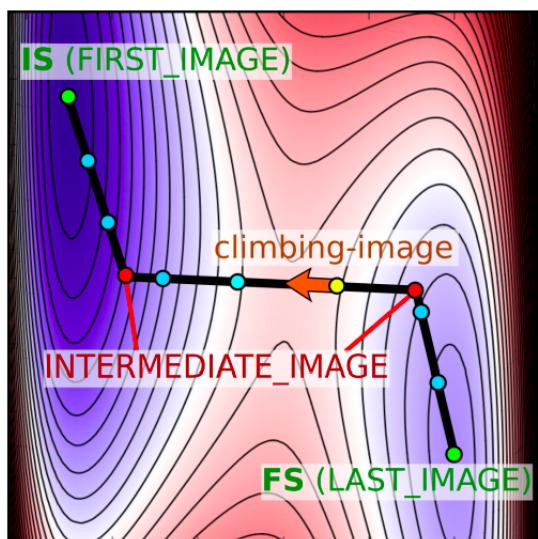


Figura 3.7: Uso de imágenes intermedias en CI-NEB.

3.3.3. neb.x

Se tienen dos bloques en el input: el primero contiene las especificaciones para NEB, mientras que en el segundo están las especificaciones para pw.x. En la parte del NEB tenemos una sección obligatoria donde definimos el camino de reacción y una parte opcional si queremos hacer CI-NEB (Figs. 3.8 y 3.9). En la parte del PW tenemos que poner más cosas en la carta de posiciones (Fig. 3.10).

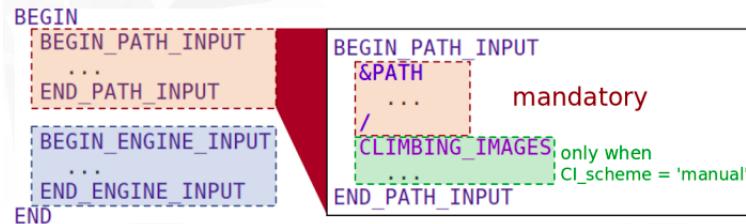


Figura 3.8: Input file: neb part.

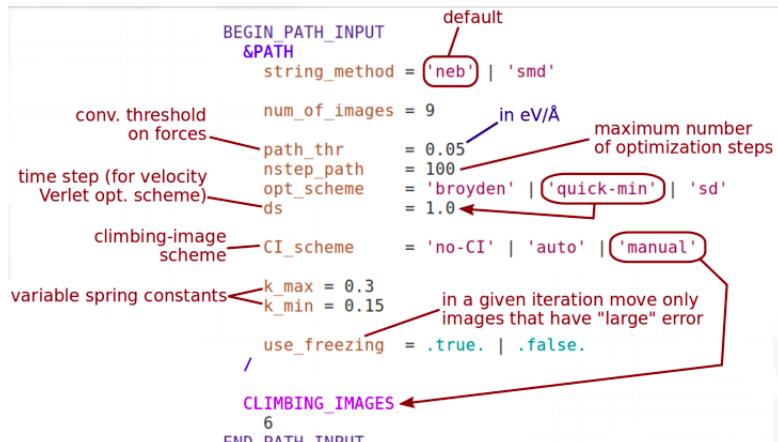


Figura 3.9: Input file: neb part con más detalles.

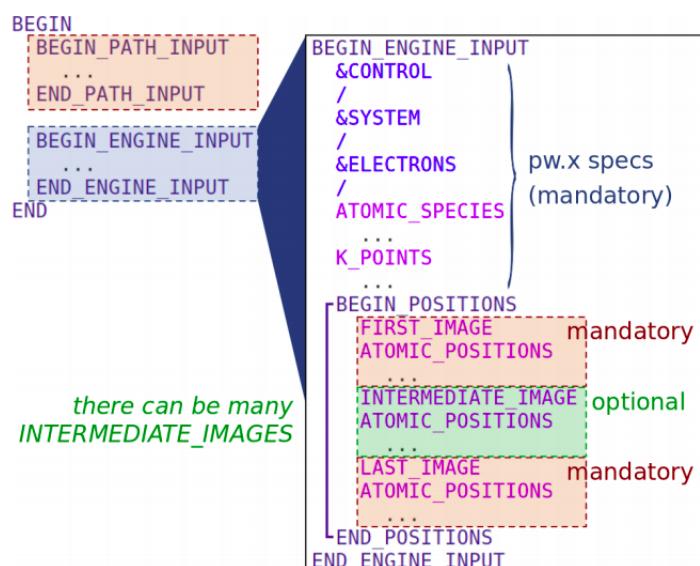


Figura 3.10: Input file: pw part.

Si usamos `use_freezing = .true.`, el cálculo primero moverá aquellas imágenes que tengan mayor error según algún criterio. Esto es conveniente: es mejor primero optimizar sólo aquellas imágenes de mayor error respecto al mínimo de energía (Fig. 3.11).

```

Input variables:
num_of_images = 8
Cl_scheme = 'auto'
use_freezing = .true.

----- iteration 10 -----
tcpu = 33.3 self-consistency for image 2
tcpu = 35.1 self-consistency for image 3
tcpu = 37.6 self-consistency for image 4
tcpu = 39.8 self-consistency for image 5
tcpu = 41.8 self-consistency for image 6
tcpu = 44.2 self-consistency for image 7

activation energy (->) = 1.173353 eV
activation energy (<-) = 1.426101 eV

image      energy (eV)      error (eV/A)      frozen
1 -2178.8704395 0.022370 T
2 -2178.7915492 0.815753 F
3 -2178.4744222 0.653550 F
4 -2177.6970863 0.271615 T
5 -2177.8369509 0.391123 T
6 -2178.3473551 0.315382 T
7 -2178.8816419 0.509630 F
8 -2179.1231876 0.000469 T

climbing image = 4
path length = 5.328 bohr
inter-image distance = 0.761 bohr

----- iteration 11 -----
tcpu = 46.7 self-consistency for image 2
tcpu = 48.6 self-consistency for image 3
tcpu = 50.9 self-consistency for image 7

activation energy (->) = 1.173353 eV
activation energy (<-) = 1.426101 eV

image      energy (eV)      error (eV/A)      frozen
1 -2178.8704395 0.022370 T
2 -2178.7844525 0.925352 F
3 -2178.4348079 0.345968 T
4 -2177.6970863 0.265414 T
5 -2177.8369509 0.391123 T
6 -2178.3473551 0.313012 T
7 -2178.8978837 0.339249 T
8 -2179.1231876 0.000469 T

climbing image = 4
path length = 5.369 bohr
inter-image distance = 0.767 bohr

```

Figura 3.11: Output file.

3.3.4. Aspectos generales de NEB

Los cálculos NEB suelen tener una convergencia difícil. Por lo tanto es conveniente:

- Usar la experiencia es relevante: usar intuición química.
- Usar estados inicial y final ya relajados. En este sentido, no conviene utilizar `first_last_opt = .true..`
- Usar imágenes intermedias.

La cantidad de imágenes a utilizar es variable. Una distancia entre imágenes de 1 ó 2 Bohr suele ser suficiente. Esta distancia se imprime en el output por lo que se recomienda hacer un dryrun haciendo `nstep_path = 0` (Fig. 3.12).

```

activation energy (->) = 0.000000 eV
activation energy (<-) = 0.000000 eV

image      energy (eV)      error (eV/A)      frozen
1 0.0000000 0.000000 F
2 0.0000000 0.000000 F
3 0.0000000 0.000000 F
4 0.0000000 0.000000 F
5 0.0000000 0.000000 F
6 0.0000000 0.000000 F
7 0.0000000 0.000000 F
8 0.0000000 0.000000 F

climbing image = 0
path length = 5.247 bohr
inter-image distance = 0.750 bohr

NEB : 0.39s CPU 0.41s WALL

```

Figura 3.12: Dryrun para conocer la distancia entre imágenes.

Es bueno visualizar previamente el camino de reacción inicial antes de hacer el cálculo. Esto se puede hacer con xcrysden:

```
xcrysden --pwi neb.in
```

Es conveniente calcular un paso elemental a la vez (una única saddle). La opción *CI_scheme = 'manual'* permite calcular varios saddle en simultáneo.

Usar *use_freezing = .true.* suele ser beneficioso.

Respecto a los valores de las constantes de resorte mínima y máxima, no son tan importantes para el cálculo: el default suele funcionar bien. De lo contrario, en el output el programa sugiere valores.

No es conveniente activar CI-NEB desde el comienzo. Primero debemos relajar el camino de reacción para lograr la mayor estabilidad posible. Recién ahí activamos el CI-NEB (*CI_scheme = 'auto'*). Esto es porque a medida que avanzan las iteraciones puede ir cambiando qué imagen es la de mayor energía, dando lugar a oscilaciones. Esto se puede setear en PWTK (Fig.).

```
load_fromNEBI neb.in

PATH {
    CI_scheme = 'no-CI'
    path_thr  = 0.8
}
runNEB neb.noCI.in

PATH {
    restart_mode = 'restart'
    CI_scheme = 'auto'
    path_thr  = 0.05
}
runNEB neb.auto.in
```

Figura 3.13: Cómo escribir en PWTK para tener CI-NEB apagado al comienzo y luego encenderlo.

No deben intercambiarse las posiciones (los índices) de los átomos en el input: las inconsistencias se romperán el programa.

You cannot do: xcrysden -pwo neb.out Why? Look into the neb.out output file and you will notice there are no atomic coordinates there. Instead, neb.x writes a number of prefix.* files, in particular: prefix.dat – energies of images prefix.int – interpolated energies on a denser grid (larger number of images) prefix.axsf – xcrysden XSF file for visualization etc. Hence you can do: xcrysden -xsf prefix.axsf Where prefix is as set in the input (e.g. H2+H, ...). Do "ls." and you will notice these files.

3.4. Q&A 2/2

I have heard that doing a CI-NEB calculation is not enough to assert that the saddle point found is the transition state. In addition to that, a phonon

calculation must be done. The obtained state can be considered as the transition state only if we obtain imaginary frequencies. Is that true ?

In general yes, however the CI-NEB should get you close enough for all practical purposes, at least for systems that you can't afford a phonon calculation

How chemisorbtion is different from physisorbtion in qe and how they are performed

Short answer: QE doesn't care, it calculates from first principles. Note however that dispersion interactions are not described by plain DFT functionals, hence you need to use some special functional or a semi-empiric dispersion correction: you will learn that tomorrow.

In order to have the best IS and FS in neb calculations can we perform a preliminar 'relax' or 'vc-relax' calculations of both the IS and the FS and then use the structural optimized parameters to perform the neb calculation? Or there are other ways to do that?

Yes, this is my recommendation, i.e. optimize the first and the last image, then make intermediate images (if needed) and only then do NEB calc. I am not sure vc-relax NEB is implemented in QE (I have never used it)

we have to keep the same lattice parameters in both IS and FS? right?

Yes. In the input on neb, you have to enter the common lattice parameter.

3.5. Hands-on

Topic: Optimizations, NEB. Automating the workflow (PWTK).

Speaker: Ari Paavo SEITSONEN (École Normale Supérieure, Paris).

3.5.1. relax

Objetivo

Realizar una optimización estructural de las posiciones atómicas, tanto con grafeno como con óxido de grafeno.

Pasos

Para el caso del grafano (grafeno con un H en cada C en trans):

- Notar que en el input ahora hay una nueva namelist: &IONS. Correr con

```
pw.x -in pw.graphane.relax.in > pw.graphane.relax.out &
```

- Analizar el output: son sucesivos pasos de SCF seguidos de cálculos de fuerzas para determinar las nuevas posiciones atómicas.
- visualizar la evolución estructural.

```
xcrysden --pwo pw.graphane.relax.out
```

Para el óxido de grafeno (átomo de grafeno adsorbido sobre una celda 3x3 de grafeno) hacer lo mismo: hacer la relajación y luego visualizarla.

```
pw.x -in pw.graphene3x3-0.relax.in > pw.graphene3x3-0.relax.out &
xcrysden --pwo pw.graphene.relax.out
```

Resultados: relax

Primero analizamos el grafano. En 5 ciclos SCF converge:

```
! total energy = -24.78069777 Ry estimated scf accuracy |1.4E-09 Ry ! total energy = -25.01952054 Ry estimated scf accuracy |2.0E-09 Ry ! total energy = -25.10918098 Ry estimated scf accuracy |6.3E-09 Ry ! total energy = -25.11219615 Ry estimated scf accuracy |1.2E-09 Ry ! total energy = -25.11408866 Ry estimated scf accuracy |9.5E-11 Ry ! total energy = -25.11411542 Ry estimated scf accuracy |2.6E-10 Ry
```

A partir de la figura vemos que se acomoda en su disposición tetraédrica dada la hibridización sp³. Se puede ver que la saturación provoca que pierda parte de sus propiedades eléctricas. Por eso todo lo respectivo al smearing está comentado en el input.

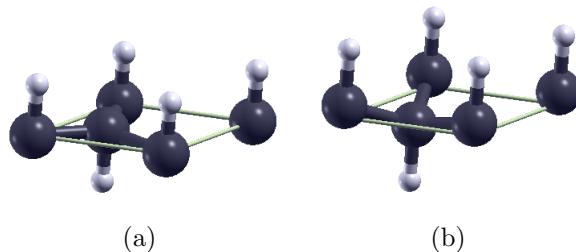


Figura 3.14: Grafano antes (a) y después (b) de la relajación.

Después tenemos un expoxi- sobre una superficie de grafeno en una cupercelda (es 3x3 pero también debería estudiarse este tamaño mediante test de convergencia). En este caso sí usamos ocupación fraccionaria (smearing). Debería usarse una *k*-mesh de 3x3x3, pero se usó gamma para que sea más rápido. La distancia entre átomos de carbono es mayor para los que participan del epóxido respecto a los demás carbonos del grafeno. Esto se puede asociar a hibridización de cada uno de los C.

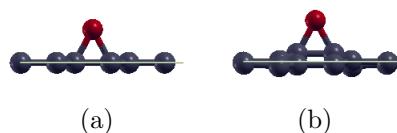


Figura 3.15: Grafeno oxidado con un grupo epoxi antes (a) y después (b) de la relajación.

Observación

Como tenemos 2 átomos diferentes, deberíamos haber chequeado la convergencia de los cutoffs para ambos elementos y tomar el valor más alto.

3.5.2. vc-relax

Objetivo

Realizar un cálculo de optimización estructural, relajando también los parámetros de red.

Pasos

Primero se analiza Zn bulk:

1. Analizar los cambios en el input y correr

```
pw.x -in pw.Zn.vc-relax.in > pw.Zn.vc-relax.out &
```

2. Analizar el output: ciclos SCF, fuerzas y tensores de estrés llevan a nuevos parámetros de red y posiciones atómicas.
3. Visualizar.

```
xcrysden --pwo pw.Zn.vc-relax.out
```

Una alternativa es hacer un barrido bidimensional de ambos parámetros de red con

```
pwtk Zn-scan.pwtk
```

La PES resultante se puede ver con

```
gnuplot plot2D.gp
```

Luego se analiza un cristal molecular de urea:

1. Ejecutar

```
pw.x -in pw.urea.vc-relax.in > pw.urea.vc-relax.out &
```

Observación

Estos cálculos ya son bastante más pesados así que conviene mandarlos al cluster.

Resultados: vc-relax

Empezamos con el cluster de Zn. Primero hacemos una relajación con un par de CI fijas: el programa buscará el mínimo. Vemos que la celda se agranda:

$$celldm(1) = 4,8 \Rightarrow celldm(1) = 4,96 \Rightarrow a = 2,62 \text{ \AA}$$

$$celldm(3) = 1,8 \Rightarrow celldm(3) = 2,02 \Rightarrow c = 5,29 \text{ \AA}$$

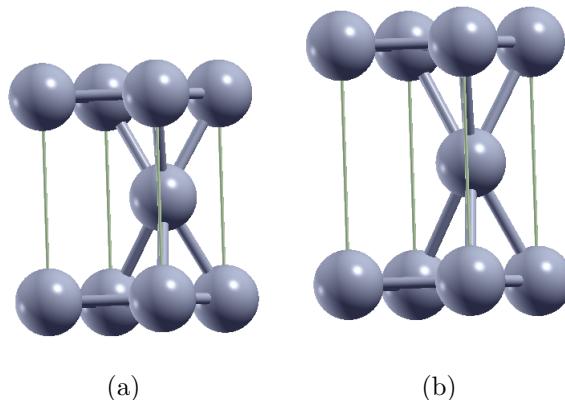


Figura 3.16: Bulk de Zn antes (a) y después (b) de la relajación tanto de posiciones atómicas como de parámetros de red.

Observación

El orden de qué tan fácil es alcanzar la convergencia es: energías, fuerzas y estrés. A medida que avanzamos en dificultad, necesitamos cutoffs más altos.

La segunda opción es hacer un barrido bidimensional. En este caso no hacemos una vc-relax, sino que por cada par de CI hacemos una SCF. Aunque esto no da un valor tan exacto como el vc-relax, permite conocer la región donde se encuentra el mínimo. Luego puedo hacer una interpolación para hallar el mínimo o hacer un vc-relax pero dando CI cercanas al mínimo.

Si vemos el mapa, podemos reconocer que el mínimo está aproximadamente en $a = 2,6 \text{ \AA}$ y $c = 5,2 \text{ \AA}$. Esto está bastante cerca de los valores calculados mediante vc-relax.

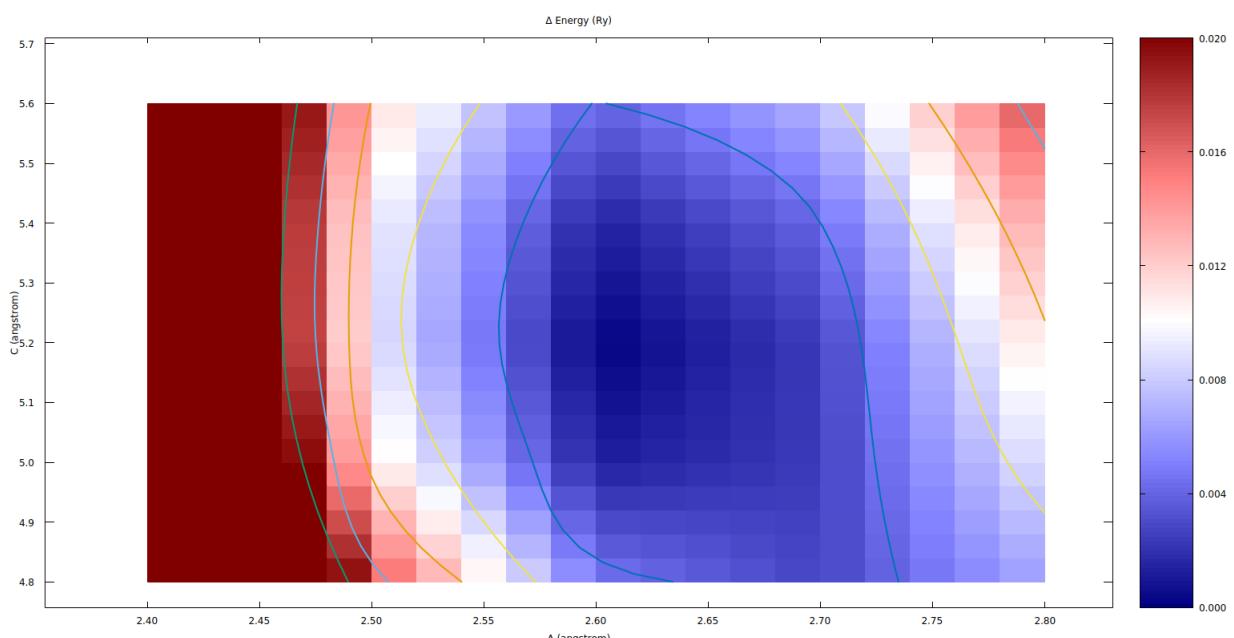


Figura 3.17: Análisis bidimensional de los parámetros de red para el Zn bulk.

Observación

En general ocurre que los parámetros de red calculados no coinciden completamente con los experimentales incluso habiendo elegido los valores que minimizan los errores, *i.e.* habiendo estudiado la convergencia. Esto se debe al término de XC. Conviene usar los valores de nuestra convergencia para hacer los estudios: así los cálculos posteriores tendrán minimizadas sus fuentes de error.

En el caso de la urea también se ve que la celda se agranda, pero el cambio es más sutil.

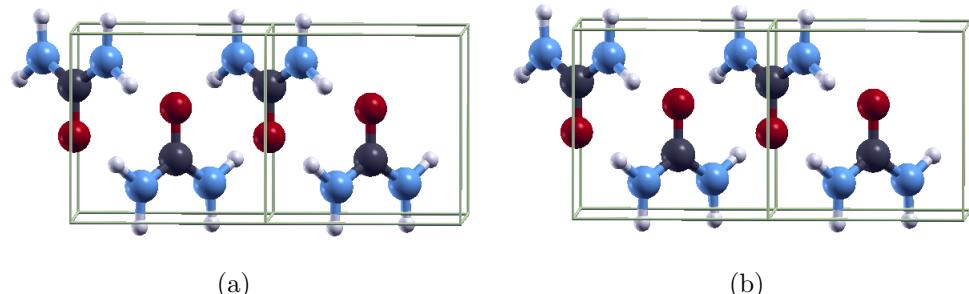


Figura 3.18: Urea antes (a) y después (b) de la relajación tanto de posiciones atómicas como de parámetros de red.

3.5.3. NEB: transferencia de H

Objetivo

Hacer una búsqueda del estado de transición para una reacción química simple utilizando el método NEB (neb.x).

Pasos

En el primero ejemplo tenemos una transferencia de protones: $\text{H} + \text{H}_2 \rightarrow \text{H}_2 + \text{H}$. Primero no utilizamos una imagen intermedia:

1. Solo se dan las imágenes inicial y final y neb.x hace una interpolación lineal con la cantidad de imágenes deseadas. El camino de reacción inicial se puede visualizar con

```
xcrysden --pwi neb.H2+H.in
```

2. Correr el programa.

```
neb.x -in neb.H2+H.in > neb.H2+H.out
```

3. Analizar el output. ¿Cuántos pasos fueron necesarios para la convergencias? ¿Cuánto vale la barrera de activación?
4. Plotear el MEP.

```
gnuplot H2+H.gp
```

5. Visualizar las estructuras del MEP con alguno de los siguientes comandos.

```
xcrysden --xyz H2+H.xyz
xcrysden --axsf H2+H.axsf
```

Luego usamos una imagen intermedia:

1. Visualizar el camino inicial. Comparar con el caso anterior.

```
xcrysden --pwi neb.H2+H.w-inter-image.in
```

2. Correr el programa.

```
neb.x -in neb.H2+H.w-inter-image.in > neb.H2+H.w-inter-image.out &
```

3. Analizar el output. Comparar barreras de activación

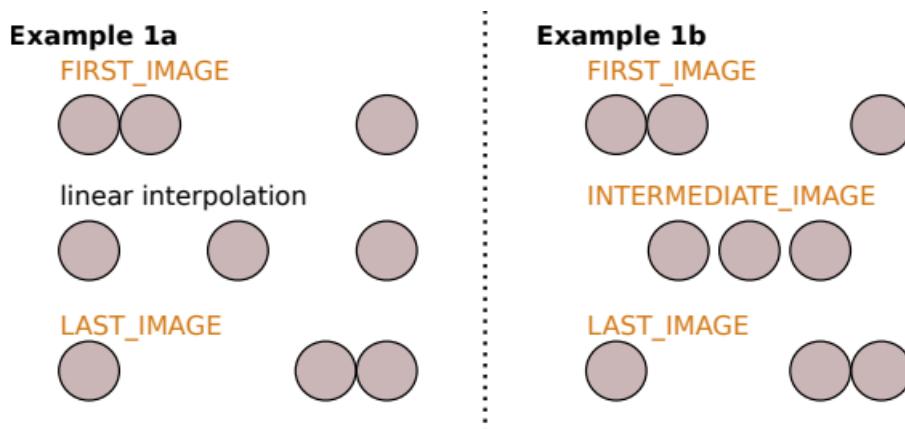


Figura 3.19: Comparación entre no usar (a) y usar (b) imagen intermedia.

Resultados: transferencia de H

La reacción es una transferencia de protones. En el primer ejemplo no utilizamos imágenes intermedias: sólo especificamos la primera imagen (reactivos) y la última (productos). El camino de reacción se discretiza en 7 imágenes y neb.x hace una interpolación lineal para ubicar las faltantes. El segundo ejemplo aprovecha un poco de intuición química para intentar dar un mejor camino de reacción inicial. Dada la ayuda que le damos al programa, la convergencia se alcanzará en menos pasos.

Observación

El umbral de convergencia $path_thr = 0,1$ no es el óptimo, sino que se eligió para que corriera más rápido.

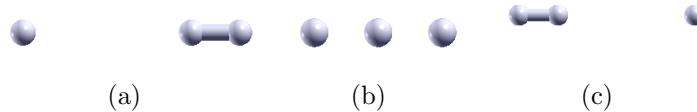
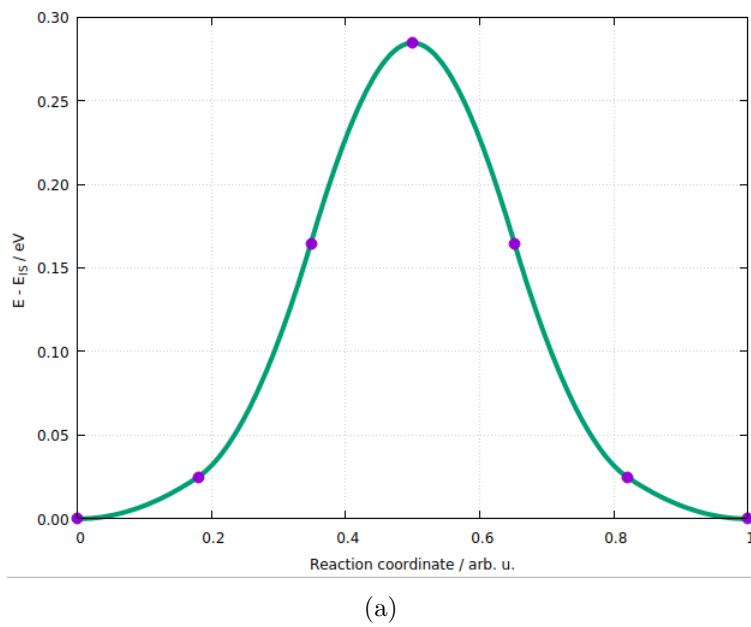
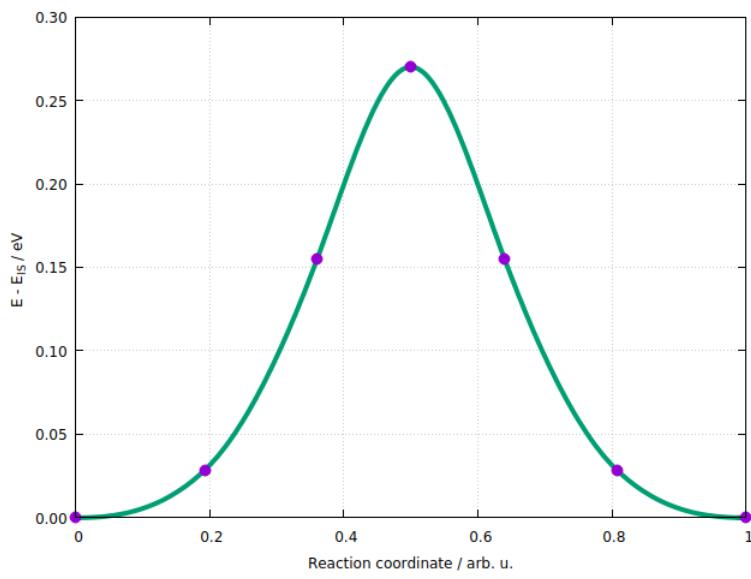


Figura 3.20: Transferencia de protones: punto inicial (a) y final (c) de la reacción. En (b) se tiene la imagen intermedia considerada.

La corrida con imagen intermedia terminó en unos 5 minutos, mientras que la otra ya va más de 45 minutos. Sus perfiles son iguales: la simetría es porque en reactivos y productos tenemos exactamente lo mismo. Sin embargo, se ve que la corrida sin imagen intermedia alcanza una barrera de activación mayor: sin imagen intermedia se llega a 0.2844 eV, mientras que con imagen intermedia tenemos 0.2690 eV. De todas formas la diferencia no es significativa, poniendo en relevancia que es mejor hacer el cálculo con imagen intermedia.



(a)



(b)

Figura 3.21: Perfil de reacción sin (a) y con (b) iamgen intermedia.

3.5.4. NEB: disociación de H

Objetivo

Llevar a cabo una NEB no trivial.

Pasos

No sólo usaremos imágenes intermedias, sino que además dividiremos el cálculo NEB en dos etapas para así ayudar a la convergencia ya que haremos CI-NEB: primero haremos una NEB común (*CI_scheme = 'no - CI'*) para estabilizar un poco el camino de reacción y, luego, permitimos que escale (*CI_scheme = 'auto'*). Esto es más fácil de hacer usando PWTK. En la primera etapa el umbral de convergencia es mayor.

1. Visualizar el camino de reacción inicial.

```
xcrysden --pwi neb.H2-diss.Al100-2x1-2L.in
```

2. Correr el programa.

```
remote_pwtk neb.pwtk
```

3. Cuando termina, traer los resultados.

```
rsync_from_hpc '*.*.out'
```

4. Analizar resultados. ¿Cuántos pasos necesito la convergencia? ¿Cuál es la barrera de activación?

5. Graficar el MEP.

```
gnuplot H2-diss.Al100-2x1-2L.gp
```

6. Visualizar los cambios estructurales.

```
xcrysden --axsf H2-diss.Al100-2x1-2L.axsf
```

Resultados: disociación de H

Tenemos la disociación de hidrógeno molecular sobre AL(100).

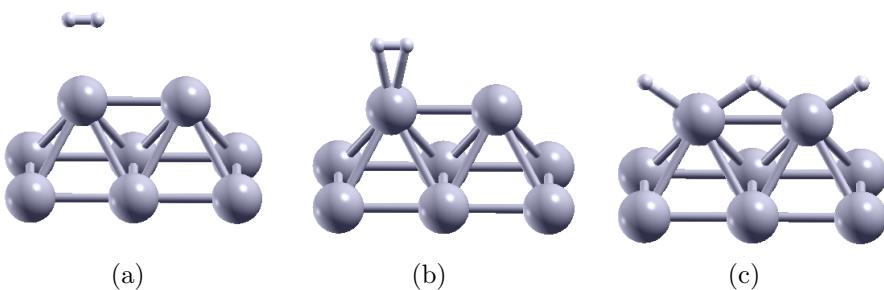


Figura 3.22: Disociación de hidrógeno molecular sobre AL(100): punto inicial (a) y final (c) de la reacción. En (b) se tiene la imagen intermedia considerada.

Se puede ver que la barrera de activación es de 1.17 eV. La asimetría de la curva es porque los productos son más estables que los reactivos.

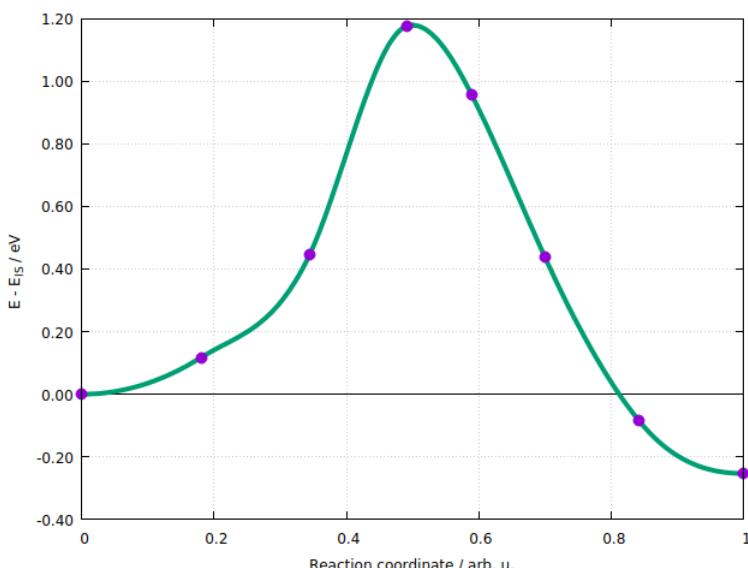


Figura 3.23: Perfil de reacción para la disociación de hidrógeno molecular sobre AL(100).

3.5.5. pwtk automation

Objetivo

Se hacen diferentes pruebas con el fin de explorar las capacidades de PWTK:

1. Cómo usar EOS (Equation of State) utility de PWTK: calcular EOS de Rh bulk.
2. Cómo correr varios cálculos con un script PWTK simple: calcular la PES bidimensional de las posiciones laterales de O @ Al(111).
3. Cómo ensamblar varios cálculos con PWTK: analizar enlace de CO sobre Rh(100).

Observación

Los parámetros son pobres con el fin de acelerar la convergencia. Todos los ejemplos importan el archivo common.pwtk donde se especifican el conjunto común da input data utilizado en todos los ejemplos: PP, cutoffs, smearing, etcétera.

CAPÍTULO 3

Pasos: EOS

Para calcular la EOS de un sistema cúbico usando la EOS utility de PWTK:

1. Ejecutar

```
pwtk eos.Rh-bulk.pwtk > eos.Rh-bulk.log &
```

2. Leer el archivo de salida 'eos.Rh-bulk.log'. Los resultados finales están en 'eos.Rh-bulk.RESULTS'.
3. Notar que el módulo de Young es más sensible al muestreo de datos que al parámetro de red y la energía de equilibrio total. ¿Por qué?

Resultados: EOS

Pasos: O@Al(111)

- 1.

Resultados: O@Al(111)

Pasos: CO@Rh(100)

- 1.

Resultados: CO@Rh(100)