

## Esercitazione: calcolo di fononi per il grafene

Dipartimento di Fisica, Università La Sapienza, Roma

14 Maggio 2018

Lo scopo dell'esercitazione è imparare a utilizzare un cluster di calcolo parallelo per calcolare le proprietà fononiche del grafene.

L'esercitazione sarà svolta sul supercalcolatore Marconi, e nello specifico nella partizione A1 (processori Intel Broadwell).

Si accede a Marconi, attraverso una shell linux (anche disponibile su sistemi MacOS o tramite applicazioni Windows, e.g. PuTTY, etc.), usando SSH:

ssh [username@login.marconi.cineca.it](mailto:username@login.marconi.cineca.it)

Per questo corso sono stati creati degli username ad-hoc:

```
a08tra70
a08tra71
a08tra72
a08tra73
a08tra74
a08tra75
a08tra76
a08tra77
```

che dureranno dal 7/5/2018 fino al 30/07/2018.

Questi username attingono a un budget di ore di calcolo che fa capo all'account

```
train_csap2018
```

Per questo corso sono stati riservati dei nodi della macchina, creando delle "reservation", cioè delle code fittizie a cui possono accedere solo gli username dell'account specificato.

Per poter utilizzare queste risorse bisogna utilizzare, all'interno del job script da passare a SLURM, la seguente sintassi:

```
#SBATCH --reservation=sap_1105 (sap_1405)
#SBATCH --account=train_csap2018
#SBATCH --partition=bdw_usr_dbg
#SBATCH --qos qos_resv
```

Per chiarire meglio le idee vediamo un esempio di job script:

```
#!/bin/bash
#SBATCH -N 1 # no. of nodes
#SBATCH --ntasks-per-node=36 # 36 cores on 1 node
#SBATCH --time=1:00:00 # time limits: 1 hour
#SBATCH --error=myJob.err # standard error file
#SBATCH --output=myJob.out # standard output file
#SBATCH --reservation=sap_1405
#SBATCH --account=train_csap2018
#SBATCH --partition=bdw_usr_dbg
```

```
#SBATCH --qos qos_resv

module load autoload
module load profile/phys
module load qe

export OMP_NUM_THREADS=1
srun --mpi=pmi2 pw.x -nk 4 -input mgb2.scf.in > mgb2.scf.out
srun --mpi=pmi2 ph.x -nk 4 -input mgb2.ph.in > mgb2.ph.out
```

Le righe precedute dal carattere # sono le direttive a SLURM e servono ad allocare le risorse richieste.

Le righe che iniziano con “module load” servono a caricare le variabili d’ambiente per l’esecuzione dei programmi (nel nostro caso qe è Quantum ESPRESSO).

```
srun --mpi=pmi2 pw.x -nk 4 -input mgb2.scf.in > mgb2.scf.out
```

lancia un calcolo di SCF con onde piane per un sistema il cui input è contenuto nel file `mgb2.scf.in` utilizzando un’opzione di parallelizzazione (`-nk`) che raggruppa in gruppi di 4 i processori disponibili per lavorare sui diversi punti k. L’output di questo calcolo viene rediretto su `mgb2.scf.out`

```
srun --mpi=pmi2 ph.x -nk 4 -input mgb2.ph.in > mgb2.ph.out
```

calcola le frequenze fononiche e le matrici dinamiche utilizzando l’output (le funzioni d’onda di KS) generate dal calcolo precedente. L’output di questo calcolo viene rediretto su `mgb2.ph.out`

Se volete, potete trovare l’output di questo calcolo nella directory:

```
/marconi_scratch/userinternal/faffinit/MgB2
```

Descriviamo ora i vari passi da compiere per fare il calcolo che ci interessa:

- 1) Effettuare il login con il proprio username su Marconi. Una volta entrati, creare una propria directory di lavoro nell’area `$CINECA_SCRATCH`:

```
[faffinito.NFAFFINIT204044] ► ssh a08tra77@login.marconi.cineca.it
```

```
[a08tra77@r000u06l01 ~]$ cd $CINECA_SCRATCH
```

```
[a08tra77@r000u06l01 ~]$ mkdir Fabio
```

- 2) Copiare i files necessari allo svolgimento dell’esercizio:

```
[a08tra77@r000u06l01 Fabio]$ cp /marconi_scratch/userinternal/faffinit/graphene/exercise/* .
```

- 3) Creare un file per l’esecuzione in modalità batch tramite SLURM. Si suggerisce di copiare l’esempio mostrato in precedenza, andando poi a modificare le ultime due righe (quelle che contengono i comandi che devono essere eseguiti).

Il primo tipo di conto che dovremo fare sarà quello necessario a generare le funzioni d’onda di Kohn-Sham con un conto SCF. Per questo utilizzeremo un job script, `scf.job`, come il seguente:

```
#!/bin/bash
#SBATCH -N 1 # no. of nodes
#SBATCH --ntasks-per-node=36 # 36 cores on 1 node
#SBATCH --time=1:00:00 # time limits: 1 hour
#SBATCH --error=myJob.err # standard error file
#SBATCH --output=myJob.out # standard output file
#SBATCH --reservation=sap_1105
#SBATCH --account=train_csap2018
#SBATCH --partition=bdw_usr_dbg
#SBATCH --qos qos_resv
```

```
module load autoload
module load profile/phys
module load qe

export OMP_NUM_THREADS=1
srun --mpi=pmi2 pw.x -nk 4 -input gr.scf.in > gr.scf.out
```

Prima di mandare in esecuzione il conto, controlla l'input di Quantum ESPRESSO (gr.scf.in) e cerca di capire cosa stai per calcolare. Per sottomettere in esecuzione questo job, usare il comando:

```
sbatch scf.job
```

L'esecuzione del conto SCF dura pochi secondi, dopo i quali sarà possibile trovare il log di esecuzione completato in gr.scf.out

- 4) A questo punto, avendo generato le funzioni d'onda di Kohn-Sham (dove sono salvate? Verifica!) è possibile andare a fare il conto dei fononi vero e proprio.

Verifica il contenuto dell'input gr.ph.in. Poi crea un job script simile al precedente in cui l'ultima riga sarà sostituita da:

```
srun --mpi=pmi2 ph.x -nk 4 -input gr.ph.in > gr.ph.out
```

Si noti che è opportuno utilizzare lo stesso numero di processori e lo stesso tipo di parametri di parallelizzazione (a meno di usare una serie di trucchi che qui non tratteremo).

Questo tipo di conto è più lungo del precedente e dovrebbe impiegare circa dieci minuti. Per verificare l'avanzamento del calcolo si suggerisce di utilizzare il comando:

```
tail -f gr.ph.out
```

- 5) Controlla i file prodotti da questo conto. Nella stessa directory dell'input troverai i file gr.dyn#, in cui # si riferisce alla #-esima rappresentazione irriducibile del grafene. Il file gr.dyn0 contiene la lista dei punti q inequivalenti.
- 6) Usiamo q2r.x per passare alla rappresentazione in spazio reale e calcolare le costanti di forze interatomiche (che verranno salvate nel file gr.fc)

```
q2r.x < gr.q2r.in > gr.q2r.out
```

- 7) Usiamo matdyn.x per calcolare la dispersione fononica e da lì calcolare la struttura a bande dei fononi. Nel file di input gr.matdyn\_disp.in è contenuta la lista dei punti della zona di Brillouin su cui devono essere disegnate le bande.

```
matdyn.x < gr.matdyn_disp.in > gr.matdyn_disp.out
```

- 8) Usiamo plotband.x per produrre il grafico a bande fononiche:

```
plotband.x < gr.plotband
```

- 9) Usare gnuplot per visualizzare il grafico:

```
gnuplot gr_freq.plot
```

- 10) Usiamo matdyn.x, ma questa volta per ottenere la DOS:

```
matdyn.x < gr.matdyn_dos.in > gr.matdyn_dos.out
```

- 11) Usare gnuplot per visualizzare la DOS ottenuta:

```
> gnuplot
```

```
> p `gr_phdos' u 1:2 w lp
```

Esercizi lasciati alla buona volontà 😊

- 1) Nel job script abbiamo sempre definito la variabile `OMP_NUM_THREADS=1`, il che forza Quantum ESPRESSO a utilizzare una parallelizzazione pura MPI (e cioè no OpenMP). Cosa succede se invece decido di usare `OMP_NUM_THREADS=2`? Controlla cosa dice l'output di Quantum ESPRESSO nelle prime righe...
- 2) Come parametro di parallelizzazione abbiamo sempre usato `nk=4`. Cosa succede quando uso `nk=1`, `nk=2`, `nk=6` (e perché, per esempio, non posso usare `nk=8`?)

Scrivendo `srun --mpi=pmi2` sto usando tutti i processi MPI a mia disposizione (e cioè tutti quelli che SLURM mi ha riservato grazie al suo job script). Però posso decidere di eseguire il mio codice con un numero arbitrario `N` di processi MPI utilizzando la sintassi: `srun --mpi=pmi2 -n N`

- 3) Prova a girare con un numero diverso di processi MPI e cerca di creare un grafico di scalabilità (weak o strong?).
- 4) Prova a usare un tipo di input diverso (i.e. un sistema fisico diverso, con più elettroni, pseudo diversi, etc. etc.) e ripeti lo stesso conto