

CDMFT Hubbard : Code de Bumsoo Kyung

Utilisation

1. Mettre les fichiers *matrix1.h*, *nr.h*, *nrutil.h*, *nrutil.cpp* et *cdmft_hubbard_nr2.cpp* (pour l'état normal), *cdmft_hubbard_sc2.cpp* (pour l'état supraconducteur) ou *cdmft_hubbard_scaf.cpp* (pour l'état supraconducteur+antiferromagnétisme) dans un répertoire.
2. Compiler le programme en exécutant la commande suivante à partir du répertoire :
`c++ -O3 -static cdmft_hubbard_nr2.cpp nrutil.cpp -o job.out`
3. Copier le fichier *job.out* dans un nouveau répertoire où se feront les calculs.
4. Ajouter un fichier *input.dat*, *input.txt* et *bqsubmit.dat* pour la soumission des calculs. Voir la documentation bqTools pour de plus amples détails.
5. Exécuter la commande : `bqsubmit &`

Fichiers *input.dat* et *input.txt*

Le fichier *input.dat* contient les paramètres fictifs (*token*) utilisés dans le programme, soit `~~U~~`, `~~CHEPOT~~` et `~~TPRIME~~`. Ils sont disposés en colonnes. Le fichier *input.txt* contient les valeurs de ces paramètres. Ils sont disposés de la même façon. Chaque ligne correspond à un ensemble de paramètres qui seront utilisés lors des calculs. Notons également que les calculs utiliseront autant de nœuds que de lignes dans le fichier de données.

Voici ce qui doit apparaître dans le fichier *input.dat*:

```
~~U~~      ~~CHEPOT~~      ~~TPRIME~~
```

et un exemple de *input.txt*:

```
7.25 2.8 -0.6
7.25 3.2 -0.6
7.25 3.6 -0.6
7.25 4 -0.6
```

Fichier *bqsubmit.dat*

Voici un exemple de fichier *bqsubmit.dat* utilisé lors de calculs:

```
batchName = tprime06a
command = ./job.out
remoteHost = ss3
concurrentJobs = 70
linkFiles = job.out
templateFiles = input.dat
```

```

copyFiles = input.txt
param1 = (U,CHEPOT,TPRIME) = load input.txt
#param1 = for i = 1 to 2 step 1
#param2 = for j = 1 to 15 step 1
#param3 = for k = 0.0 to 2.0 step 1.0
#param4 = i = [0.9 1.0 1.1 1.2 1.3 1.4 1.5]
#param5 = (i,j,k) = [(1,2,3), (4,5,6), (7,8,9), (10,11,12)]

```

Description du modèle

La description du modèle (amas et bain) est figée dans le code même. Pour plus de détails, s'adresser à M. Bumssoo Kyung.

Résultats

Les résultats des calculs se retrouvent dans les répertoires *[batchName]_XXXXX.BQ*, où *[batchName]* est le nom spécifié dans le fichier *bqsubmit.dat* et *XXXXX* est un numéro d'identification unique. Chaque ensemble de valeurs des paramètres spécifiés dans le fichier *input.txt* possède son propre répertoire. Une copie exacte de ces répertoires dont le nom contient les paramètre (e.g. *[batchName]_U8_CHEPOT4_TPRIME0_XXXXX.BQ*) est également créée, mais peut être supprimée lorsque les calculs sont terminés.

Le fichier de résultats se trouve dans *[batchName]_XXXXX.BQ* et se nomme *[batchName]_XXXXX.oXXXXX*. Il peut être ouvert avec tout éditeur de texte. Les résultats importants apparaissent à la fin de ce fichier sous la forme suivante:

```

density = 0.965809

old e1, e2, v1, v2 and func
-0.643364 1.44744 0.524648 0.964474 func = 3.46104
new e1, e2, v1, v2 and func
-0.643361 1.44744 0.524655 0.964474 func = 3.46104

count = 39, convergence_ratio = 1.97309e-06
double_occupancy = 0.0557312
pot_energy = 0.390118
cluster_kin_energy = -0.635417
total_cluster_energy = -0.245299
cluster_kin_energy - mu*n = -3.2431
total_cluster_energy - mu*n = -2.85298
memory for GREEN_FUNCTION_CDMFT = 324.48
chepot0 = -1.1682
chepot_new = 2.7, CHEPOT = 2.7
DENSITY_FROM_GREEN_FUNCTION = 0.995569
double_occupancy = 0.0557312
pot_energy = 0.390118
lattice_kin_energy = -1.07804
total_lattice_energy = -0.687921
lattice_kin_energy - mu*n = -3.68572

```

```

total_lattice_energy - mu*n = -3.29561
Obtain lattice green_function from REAL frequency
chepot0 = -1.1682
chepot_new = 2.7, CHEPOT = 2.7
total_weight = 0.994127
(ikx_fermi,iky_fermi) = (16,64)
(ikx_iky_fermi,ikx_iky_fermi) = (24,24)

```

La première densité affichée est la densité de l'amas par opposition à la densité du réseau infini DENSITY_FROM_GREEN_FUNCTION. Les paramètres de bain e1, e2, v1 et v2 correspondent aux énergies des sites de bain et aux interactions entre l'amas et le bain respectivement.

Paramètres de départ

Si certains calculs ne convergent pas (i.e. si le texte ci-haut est absent du fichier de sortie), on peut modifier légèrement les paramètres de départ e1, e2, v1 et v2 directement dans le code. À cet effet, il suffit d'ouvrir le fichier de code (e.g. *cdmft_hubbard_nr2.cpp* pour l'état normal) et de modifier les lignes suivantes en toute fin de code :

```

if ( option == FIRST )
    {
        xx[ 1 ] = -0.305873;
        xx[ 2 ] =  0.30429;
        xx[ 3 ] =  0.517486;
        xx[ 4 ] =  0.474504;
    }

```

Ici, la première valeur correspond à e1; la seconde, à e2; et ainsi de suite. Notons que ces valeurs apparaissent également au tout début du fichier de résultats *[batchName]_XXXXX.oXXXXX*.

Il ne faut surtout pas oublier de recompiler le code lorsque l'on fait de tels changements

Secteur fixe vs variable

Par défaut, le secteur d'énergie est variable. Or il est possible que certains calculs ne convergent pas car le secteur d'énergie minimale varie d'une itération à l'autre. Cela se voit dans le fichier de sortie à la ligne suivante:

```

minimum energy sector = 6

```

Lorsque cette valeur varie d'une itération à l'autre (ou à chaque 3, 4, 5 itérations), on peut utiliser une version différente du code: *cdmft_hubbard_nr1.cpp*. Il suffit alors de modifier la ligne:

```
const int SECTOR = 4;
```

et d'inscrire la valeur du secteur qui apparaît le plus souvent dans le fichier de sortie.

Il ne faut surtout pas oublier de recompiler le code lorsque l'on fait de tels changements