

# ELSES ノート：機能編：2017 年 5 月 18 日

ELSES 研究会（文責：星健夫）

対象：本文書作成時の開発版

## 目 次

<b>1</b>	<b>分散メモリ (DST) ワークフロー</b>	<b>4</b>
1.1	概論：ワークフローの選択	4
1.1.1	3つのワークフロー	4
1.1.2	DST ワークフローの現状での制限事項	4
1.1.3	計算規模と計算機環境による分類	5
1.2	DST ワークフローの使い方	5
1.2.1	DST ワークフロー用 sample を用いた 実行	5
1.2.2	設定 XML ファイルでのポイント	7
<b>2</b>	<b>構造 XML ファイルの新機能</b>	<b>9</b>
2.1	elses-xml-generate の新しい機能:atom ID とファイル並列化	9
2.1.1	使用法 1:従来と同じ使い方 (atom ID なし、分割せず)	9
2.1.2	使用法 2:atom ID の付加 (ファイル分割せず)	9
2.1.3	使用法 3:分割ファイルの生成	9
2.2	非並列ファイル読み込み	10
2.2.1	DOM 法 (従来法)	10
2.2.2	SAX 法 (1):原子数指定なし	11
2.2.3	SAX 法 (2):原子数指定あり	11
2.2.4	SAX 法 (3):原子数指定あり, 「root-only」モード	11
2.3	分割 XML ファイルによる並列化ファイル読み込み	12
2.4	分割 XML ファイルによる並列化ファイル書き出し	12
2.5	ELSES 本体による XML ファイルの分割・統合	13
2.6	XML ファイル出力の 3 つのモードと継続計算	14
<b>3</b>	<b>物理量出力の新機能</b>	<b>16</b>
3.1	波動関数 (LCAO 係数) 書き出しにおける新機能	16
3.1.1	連続書き出し機能	16
3.1.2	参考:波動関数 (LCAO 係数) ファイルの仕様	16
3.2	H,S 行列および固有エネルギー書き出しにおける新機能	17
3.2.1	H,S 行列生成のみのモード (matrix generation モード)	19
3.3	局所結合エネルギー (ICOHP) の出力	20
3.4	波動関数グリッドデータ生成ツール:elses-generate-cubefile	21
3.4.1	カットオフの導入	21
3.4.2	符号反転	21
3.4.3	grid region の指定	22
3.4.4	入出力ファイル名のカスタム指定	22
3.5	電荷分布関数の出力	24
3.6	原子基底情報 (basis info) の出力	26

<b>4</b>	<b>マシン環境固有の話題</b>	<b>27</b>
4.1	「京」および FX10 での ELSESES 利用	27
<b>5</b>	<b>sample データ関連の新機能</b>	<b>30</b>
5.1	sample データ検証ツール (shell script) (非 DST ワークフロー)	30
5.2	sample データ検証ツール (shell script) (DST ワークフロー)	32
5.3	band 計算データ検証ツール (shell script)	35
5.3.1	計算手順	35
5.3.2	検証ツール (shell script)	40
<b>6</b>	<b>エラー検出機能</b>	<b>41</b>
6.1	原子構造のチェック (距離が近すぎる原子ペアを検出)	41
<b>7</b>	<b>構造作成関連ツール</b>	<b>43</b>
7.1	大規模系むけ supercell 構造作成ツール	43
7.2	分子凝集系むけ group ID 設定ツール	46
<b>8</b>	<b>van der Waals 力</b>	<b>49</b>
8.1	概要とサンプルデータ	49
8.2	理論の詳細	49
<b>9</b>	<b>Group ID 機能</b>	<b>51</b>
9.1	概要とサンプルデータ	51
9.2	group ID の活用例: group ごとの重心座標の計算	53
<b>10</b>	<b>極限的な大規模計算</b>	<b>55</b>
10.1	Microcell booking の手動設定	55
<b>11</b>	<b>External Library の利用</b>	<b>57</b>
11.1	EigenKernel の利用	57
<b>12</b>	<b>Sample data による case study</b>	<b>60</b>
12.1	Kwon モデルによる Si 結晶: diamond 構造と simple cubic 構造	60
12.1.1	diamond 構造計算	60
12.1.2	simple cubic 構造計算	61
12.2	NRL モデルによる W(タングステン) 結晶: BCC, FCC, SC 構造	63
12.2.1	BCC 構造計算	63
12.2.2	FCC 構造計算	64
12.2.3	SC 構造計算	64
12.3	共役高分子系: poly-(9,9 dioctyl fluorene) (ELSESES sample extra に収録)	65
12.3.1	モノマー	65
12.3.2	ダイマー	66
12.3.3	10 量体 (DST ワークフロー)	67
12.4	大規模アモルファス状共役高分子系の DST ワークフロー	68
12.4.1	約 13 万原子系 (FL-test-07-132864atom.xyz)	68
12.4.2	約 212 万原子系 (FL-test-11-2125824atom.xyz)	70
12.4.3	superccell 作成を含む作業: 約 13 万原子系を例に	74
12.5	理想構造ダイヤモンド計算	77

12.5.1	4096 原子系, 4 ノード計算	77
12.5.2	52 万原子系, 512 ノード計算	78
12.5.3	約 1 億原子系, 2592 ノード計算	80
12.6	行列データ生成用 ICNT 計算	81
12.6.1	ICNT11.2 万原子系 (spd 型)	81
12.7	イリジウムを含む分子: Ir(CO) <sub>3</sub> Cl	83
12.7.1	概要	83
12.7.2	データパッケージの詳細	83
12.8	行列データ生成用 VCNT 計算	86
12.8.1	構造作成	86
12.8.2	行列作成	88
<b>13</b>	<b>雑多な機能</b>	<b>90</b>
13.1	計算途中での終了 (削除項目)	90
13.2	4 倍精度演算テストルーチン	90
13.3	出力ファイルの保存ディレクトリを指定	90
<b>14</b>	<b>ELSEES XML 仕様のバージョン</b>	<b>92</b>
14.1	ELSEES XML Version. 1.0	92
14.2	ELSEES XML Version. 5.01	92
<b>15</b>	<b>ELSEES コードについての雑多な情報</b>	<b>93</b>
15.1	単位換算に用いる物理定数など	93
<b>16</b>	<b>ユーザーからの報告 (質問) と回答</b>	<b>94</b>
16.1	ELSEES 実行前の作業 (初期構造作成など) について	94
16.1.1	原子をファイルに記す順番に制限はありますか?	94
16.1.2	非周期系でも「セル」を設定する必要がありますか?	94
16.1.3	原子を置ける座標に制限はありますか?	94
16.2	ELSEES 実行について	94
16.2.1	実行時エラー: 「ERROR(set_interac_list_proj)」	94
16.2.2	実行時エラー: 「The structure XML file may be wrong」「The element tag is required」	95
16.2.3	実行時エラー: 「ERROR in uSu」	95
16.3	波動関数可視化について	96
16.3.1	VESTA(v.3.2.1) で、波動関数表示がおかしいです	96
16.4	ELSEES 実行後の作業について (ただし波動関数可視化関連をのぞく)	96
16.5	マシン環境に依存した話題	96
16.5.1	「京」でランクディレクトリは使うべきですか?	96
16.5.2	「京」(対話実行) で「cannot execute binary file」と出て動きません	96
16.5.3	gfortran v4.6 未満で make できません	97
<b>A</b>	<b>本ノートの更新記録</b>	<b>98</b>
<b>B</b>	<b>関連事項</b>	<b>102</b>
B.1	平均二乗変位 (MSD) と Participation Ratio (PR)	102
B.1.1	MSD	102
B.1.2	Participation Ratio	102

本文書は主に、QuickStart に記載がない新機能を説明している。一部の仕様は、今後変更される可能性がある。本文書は各種文章を機械的につなぎあわせたものであるため、章によって文体や説明のスタイルが統一されていない場合がある。

## 1 分散メモリ (DST) ワークフロー

本章では、分散メモリ版ワークフロー (以下、DST(distributed memory system) ワークフロー) について解説します。MPI, OpenMP(OMP) といった並列計算手法について、ある程度の知識があることを前提としています。

DST ワークフローとは MPI/OMP 複合並列を念頭においたワークフローで、非 DST ワークフローとは、OMP 並列を念頭においたワークフローです。本文書では「MPI」という言葉は、MPI コマンド (=通信コマンド) に直接関連したところでのみ用います。以下では、DST ワークフローにおいて、MPI での並列単位を「node」、OMP での並列単位を「thread」と呼びます。

DST ワークフローは、論文 [1] に基づきます。DST ワークフローを利用して発表等を行う場合は、すくなくとも論文 [1] を引用してくださいよう、お願い申し上げます。

### 1.1 概論：ワークフローの選択

#### 1.1.1 3つのワークフロー

ELSES には、おおまかに言って、3つのワークフローが存在します。

- (a) 非 DST 版対角化ワークフロー (default)
- (b) 非 DST 版 Krylov ワークフロー (OMP 並列が可能)
- (c) DST 版 Krylov ワークフロー (MPI/OMP 複合並列が可能)

(a) は LAPACK ルーチンにより対角化しますので、計算速度は LAPACK ルーチンの実装に依存します。Intel fortran に付属する Math Kernal Library (MKL) など、OMP 並列されているライブラリーをリンクすることで、OMP 並列されます。(b)(c) は、Krylov ソルバー、より正確には multiple Arnoldi 法 [1] にもとづくワークフローで、オーダー N 計算を実現するものです。(b)(c) は数学的に等価な計算であり、結果は全く一致します<sup>1</sup>。(c) においては、MPI/OMP 複合並列の他、pure MPI 並列も可能です。

#### 1.1.2 DST ワークフローの現状での制限事項

主な制限事項として、以下があります。

- Krylov ソルバーのみ対応。Krylov ソルバーのうち、multiple Arnoldi 法 [1] のみが正式に対応。他 Krylov ソルバーは (実験的に対応しているものもあるが) 状況に応じて正式対応する方針。
- 小さい系 (総原子数が 200 以下程度) では、(詳細設定次第だが) 動かない。このような系は、そもそも高速にならない (次節参照)。
- (総原子数) > (総コア数) のときのみ、対応。それ以外の場合については、状況に応じて対応する方針。
- MD(optimization, dynamics など total energy と force の計算) しかできない。注: DOS, LDOS の計算などは、対応予定。バンド図 (E-k カーブ) 描画は対応予定なし (大規模な計算にはならないので)。
- CSC に非対応 (対応予定)。
- GENO hamiltonian のみに対応 (他 Hamiltonian 対応は未定)。

<sup>1</sup> 和の順序が変わることなどから、丸め誤差が若干入りますが、通常 10 桁程度は一致します。

### 1.1.3 計算規模と計算機環境による分類

並列計算においては、必ずしも並列度数分だけ高速になるわけではありません。おおよそ、以下が目安となります。

- $10^2$  原子程度以下では (a) が良い。逆にいうと高速化する手だてがない。
- Krylov ソルバー ((b)(c)) が威力を発揮するのは、 $10^3$  原子前後程度から。

一方、今日の典型的な計算機環境は、下記のように分類ができます。

- (i) ワークステーション ……  $1\text{--}10^1$  コア程度。共有メモリ (OMP) 計算。PC から研究室レベルのワークステーション。  
(a) or (b)。 (c) のテストも可能。
- (ii) ミドルクラスのスパコン ……  $10^2\text{--}10^3$  コア程度。分散メモリ (MPI/OMP) 計算。国内各所スーパーコンピュータでの通常ジョブ。  
主に (c)。特別な配慮 (下記) は不要な場合が大半。
- (iii) トップクラスのスパコン ……  $10^3\text{--}10^5$  コア以上。分散メモリ (MPI/OMP) 計算。「京」など。  
主に (c)。特別な配慮 (ファイル IO の並列化など) [2] が必要な場合がある。

(a)(b) については、「Quick Start」をご覧ください。以下は (c) について、(i)(ii) を前提に説明します。

## 1.2 DST ワークフローの使い方

以下では、ワークステーション (前節 (i)) またはミドルクラススパコン環境 (前節 (ii)) を念頭において、DST ワークフローの簡単な使い方を説明する。

### 1.2.1 DST ワークフロー用 sample を用いた 実行

MPI 計算 (DST ワークフロー) 向けの sample は、sample\_for\_mpi/ディレクトリのみに含まれている。現状では、液体 Carbon1728 原子系・13824 原子系を入力ファイルとするデータが、以下にある。

```
sample/sample_geno_mpi/C_liq_01728atom_dst
sample/sample_geno_mpi/C_liq_01728atom_dst_opt
sample/sample_geno_mpi/C_liq_01728atom_dst_cell_change_only
sample/sample_geno_mpi/C_liq_13824atom_dst
```

ディレクトリ C\_liq\_01728atom\_dst\_opt は「optimization モード」計算 (QuickStart 参照) であり、ディレクトリ C\_liq\_01728atom\_dst は「cell-change-only モード」計算 (QuickStart 参照) である。他は「ダイナミクスモード」計算である。計算結果ファイルが、それぞれのディレクトリの下のディレクトリ result に格納されている。

以下、液体 Carbon13824 原子系 (sample\_geno\_mpi/C\_liq\_01728atom\_dst) を例にして、標準的な linux での手順を説明する。

1. ソースコードでの準備 …… src/ディレクトリで、

```
> cp code_for_mpi/elses-lib-mpi-wrapper.f90 elses-lib-mpi-wrapper-compile.f90
```

とする。あとは、Makefile.inc を MPI に対応した形に書き換えた上で、ELSES をコンパイルする。

注：非 DST 版に戻すときは、src/ディレクトリで、

```
> cp code_for_mpi/elses-lib-mpi-wrapper-dummy.f90 elses-lib-mpi-wrapper-compile.f90
```

とする。

- OMP 並列度数の設定・・・以下、(MPI 並列度, OMP 並列度)=(2,6) で計算することを想定する。環境変数 OMP\_NUM\_THREADS を通じて、OMP 並列度=6 と設定する。具体的には、bash 等の sh 系なら、

```
> export OMP_NUM_THREADS=6
```

とする。csh 系なら

```
> setenv OMP_NUM_THREADS 6
```

とする。

- 実行・・・実行する：

```
> mpirun -n 2 ../../bin/elses -log_node_number=2 ./config.xml > log.txt &
```

注：intel 系 CPU では、numatcl を頭につけると良い。

注：-log\_node\_number=2 は、logfile を出力する node 数を決める。省略できて、省略時には 8 に設定される。  
-log\_node\_number=2 だと、k=0,1 番目の node に対して logfile が、

```
log-node000000.txt  
log-node000001.txt
```

という名称で、それぞれ出力される。これらを node logfile と呼ぶ。node ごとの経過時間を測定するときなどに、有用である。上記設定値がゼロ以下なら、どの node に対しても logfile が出力される。上記設定値が総ノード数より多い場合は、全ノードに対して logfile が出力される。

- 計算結果の確認・・・生成されたデータファイル Output.txt を、参照データディレクトリの同名ファイルと比較する。内容は QuickStart に記載されている。
- 動作検証・・・意図通りに動作しているのかの検証方法を記す。1 番目の node の node logfile (log-node000001.txt) を例にとる。冒頭は、以下のようにになっているはずである

```
@@ ELSES version 0.03.14  
INFO-MPI: True MPI wrapper  
Date: 2012 11 27; Time: 01 33 47  
INFO-MPI-OMP: P_MPI, P_OMP=          2          6  
INFO-MPI:   mpi_is_active or not :   T  
INFO-MPI:   myrank, nprocs      =          1          2
```

各行の意味を説明する。1 行目はバージョン番号表示である。2 行目が「True MPI wrapper」でない場合は、上記作業 1(ソースコードでの準備)を行っていないことを意味する。3 行目は、実行開始時刻なので、計算内容には無関係である。4 行目は、(MPI 並列度, OMP 並列度)=(2,6)、を意味する。5 行目の最後が「T」(true の意)のときは、システム側で MPI 機能が起動していることを意味する。そうでない場合は、mpirun が正しく起動していない、など、システム側での設定に問題があることになる。6 行目は、自身が k=1 番目の node であること、全体として 2 node であること、を意味する。

6. 経過時間の解析：全体 …… 各ログファイルに対して、下記のように、MD step ごとの経過時間がでる。参照データディレクトリで説明する。

```
> grep MDl log-node000001.txt (下記では最初の4行だけを表記)
elaps-time(befor MDloop)=          0          0.522700          0.000000          0.002308
elaps-time(lap MDloop)=           1          256.892800          0.000746          1.150579
elaps-time(lap MDloop)=           2          256.951000          0.000743          1.797614
elaps-time(lap MDloop)=           3          257.286300          0.000697          0.728417
```

ここで、1行に現れる数字のうち最初の1つは、MD loopのステップ数を表す。「befor MDloop」の行は、MD計算が始まる前の初期処理（入力ファイルの読み込みなど）を意味する。1行に現れる3つの実数値は、1ステップごとに、(a) 総経過時間、(b) MPI 通信時間の総計、(c) ノード間ロードアンバランスからくる待ち時間の総計、を秒単位で表す。注：最初のMD step(MDloop=1)では、種々の処理をするため、後続のstepより時間が余分にかかる。注：default設定では、構造データの書き出しはroot node (k=0番目のノード)のみで行っている。そのため、ロードアンバランスの原因になる。

7. 経過時間の解析：詳細 …… 1ステップ中は、おおまかには、下記手順になる。(a) gKrylov ソルバー (Green 関数の生成) 1 回目 (b) chemical potential 決定 (2 分法) (c) gKrylov ソルバー (Green 関数の生成) 2 回目 (d) エネルギー・電荷分布・force の計算。通常は (a)+(c) が dominant である。(a)(c) は以下の要領で出る。

```
> grep -i time | grep 'MAIN loop' log-node000001.txt (下記では最初の6行だけを表記)
TIME:qm_solver_gkrylov_dst:MAIN loop =          127.47790    1
TIME:qm_solver_gkrylov_dst:MAIN loop =          128.28390    2
TIME:qm_solver_gkrylov_dst:MAIN loop =          127.79620    1
TIME:qm_solver_gkrylov_dst:MAIN loop =          128.38150    2
TIME:qm_solver_gkrylov_dst:MAIN loop =          128.16750    1
TIME:qm_solver_gkrylov_dst:MAIN loop =          128.33680    2
```

末尾の「1」「2」が、それぞれ (a),(c) に対応する。2ステップ目 (MDloop=2) だと

$(a)+(c) = 127.79620 + 128.38150 = 256.178 \text{ sec}$

が全体の時間 (7 に出ている 256.951sec) に対して、ほとんどを占めていることが分かる。

一方 (b) は、以下で出る。

```
> grep -i time log-node000001.txt | grep chem (下記では最初の1行だけを表記)
TIME:qm_solver_gkrylov_dst:chem pot =          0.10040
```

上記ですと、約 0.1 秒であることを意味する。

### 1.2.2 設定 XML ファイルでのポイント

設定 XML ファイル (config.xml) でのポイントは、以下です。

1. calc タグ内に distirbuted タグを、以下のように書く：

```
<distributed mode="on">
  <micro_cell_booking mode="one_cell"></micro_cell_booking>
</distributed>
```

上記で「mode="on"」を「mode="off"」にすると、上記タグは無視されて、従来通りの非 DST 計算になります

2. calc タグ内に solver タグを、以下のように書く：

```
<solver scheme="gKrylov_A">
  <projection> 200 </projection>
  <dimension> 30 </dimension>
  <mode_for_large_memory> 0 </mode_for_large_memory>
  <mArnoldi_q> 15 </mArnoldi_q>
  <inner_cg_loop>
    <max_iteration> 100 </max_iteration>
    <convergence_eps> -8 </convergence_eps>
  </inner_cg_loop>
  <mode_for_suggest_projection> default </mode_for_suggest_projection>
</solver>
```

省略してよいものもありますが、その説明はここではしません。projection タグ・dimension タグ・mArnoldi\_q タグは、それぞれ、論文 [1] での  $N_{PR} \cdot \nu \cdot q$  に相当するものです。一方、max\_iteration タグ・convergence\_eps は、論文 [1] にある「inner CG loop」に関連した設定で、inner CG loop の反復上限値と収束判定条件（上記だと、残差  $\leq 10^{-8}$ ）を定めています。max\_iteration タグ・convergence\_eps は、通常変更する必要はないです。

3. (optional) XML 読み込み方法を DOM 法 (低速) から SAX 法 (高速) へ変えます。具体的には設定 XML ファイルの cluster タグ内の属性を、以下のように設定します：

```
<cluster structure="C_liq_13824atom.xml" parser="sax" tag_dump="off"
  read_mode="default" cutoff_radius="8.0d0" />
```

上記は紙面横幅の都合で 2 行に渡って書いてありますが、ファイルでは 1 行で書いてください。

上記のうち「read\_mode="default"」「tag\_dump="off"」は省略できます。「parser="sax"」を「parser="dom"」とするか、parser 属性を省略すると、DOM 法となります。



## 2 構造 XML ファイルの新機能

### 2.1 elses-xml-generate の新しい機能:atom ID とファイル並列化

ツール「elses-xml-generate」が拡張され、(1) atom ID を付加できる、(2) 分割された構造 XML ファイルが作れる、の2機能が追加された。これら機能は、ファイル IO 並列化のときに使われます。

#### 2.1.1 使用法 1:従来と同じ使い方 (atom ID なし、分割せず)

実行は、従来 (QuickStart 記載) の方法と同様、

```
> elses-xml-generate generate.xml sample.xml
```

または、

```
> elses-xml-generate generate.xml sample.xml -split=0
```

とします。

結果は、従来と同様、sample.xml(構造ファイル)ができます。内容は従来と全く同じです。

他に sample\_basic.xml (構造 basic ファイル) ができます。このファイルは、構造 XML ファイルから原子情報 (atom タグ) をのぞいたもので、unitcell タグと heatbath タグのみを含みます。ファイル IO 並列化のときに利用されることがあります。構造 basic ファイルの名称は、構造ファイル名から派生されます。例えば、構造ファイル名が「abc.xml」なら、構造 basic ファイル名は「abc\_basic.xml」となります。

#### 2.1.2 使用法 2:atom ID の付加 (ファイル分割せず)

実行の際に、

```
> elses-xml-generate generate.xml sample.xml -split=1
```

とします。結果として、前節と同様に、sample.xml (構造ファイル) sample\_basic.xml (構造 basic ファイル) の2ファイルができます。違いとしては、sample.xml での全 atom タグに ID 属性が付加されています。下記に例を示します；

```
<atom element="C" id="1" class="" motion="free">
```

ID 属性の値は、上から順に、1,2,3,...(全原子数)、となっています。ID 属性はファイル IO 並列化の際に参照され、それ以外では無視されます。

#### 2.1.3 使用法 3:分割ファイルの生成

4 ファイルに分割する場合を例に説明します。分割ファイル作成時には、atom ID の付加が必須となります。

実行時に分割数 (今の場合は 4) を、以下の要領で指定します：

```
> elses-xml-generate generate.xml sample.xml -split=4
```

結果として、

```
sample_000000.xml
sample_000001.xml
sample_000002.xml
sample_000003.xml
sample_basic.xml (構造 basic ファイル)
```

ができます。構造 basic ファイルは、前節と同じです。他の 4 ファイルが、分割された構造ファイルです。名称は実行時に指定したファイル名 (この例だと sample.xml) から派生されます。分割された構造ファイルの特徴として、以下のような split タグが入っています：

```
<split file_index="2" number_of_files="4" atoms_in_this_file="3456"
      atom_initial="6913" atom_final="10368" >
```

(注：上記は紙面横幅の都合で 2 行に渡っていますが、実際のファイル記述は 1 行です)。

それぞれの属性値の意味は、

```
file_index 値：ファイル番号 (例えば、sample_000002.xml なら、file_index 値=2)
number_of_files 値：ファイルの分割数
atoms_in_this_file 値：このファイルに入っている原子数
atom_initial 値：最初の atom ID
atom_final 値：最後の atom ID
```

です。N 原子系を、M 分割すると、j 番目 (j=0,1,...) のファイルでは、

```
atom_initial 値 = j * N / M + 1
atom_final 値   = (j+1) * N / M
```

の atom ID をもつ原子が記録されています。

## 2.2 非並列ファイル読み込み

非並列ファイル読み込みにおいても、数種の読み込み方法を実装しています。下記では、液体 Carbon13824 原子系 (sample/C\_liq\_13824atom) を例に説明をします。

### 2.2.1 DOM 法 (従来法)

従来と同様、XML パーサーとして DOM 法を使います。現状では、default の方法となっています。設定 XML ファイルの cluster タグを、下記のように書きます。

```
<cluster structure="C_liq_1728atom.xml" parser="dom" tag_dump="off"
      read_mode="default" cutoff_radius="8.0d0" />
```

(上記で cluster タグは紙面横幅の都合で 2 行に渡って書いてありますが、ファイルでは 1 行で書いてください)

注：このうち「parser="dom"」「tag\_dump="off"」「read\_mode="default"」は、省略できます。

注：DOM 法 (従来法) は古くから ELSES に装備されている機能ですが、大規模系では時間とメモリが膨大に消費され、実質的に機能しなくなります。限界は (ハードウェアによりますが) 数 10 万原子程度のようなようです。

注：ユーザーが DOM 法を使うメリットはありませんので、次節にある「SAX 法」に移行することをおすすめします。将来的には、DOM 法で構造 XML ファイルを読む機能は廃止する予定です。なお、設定 XML ファイルを読み込むときには、DOM 法を引き続き利用する予定です。

### 2.2.2 SAX 法 (1):原子数指定なし

XML パーサーとして高速な SAX 法を使います。設定 XML ファイルの cluster タグを、下記のように書きます。

```
<cluster structure="C_liq_1728atom.xml" parser="sax" tag_dump="off"
  read_mode="default" cutoff_radius="8.0d0" />
```

(上記で cluster タグは紙面横幅の都合で 2 行に渡って書いてありますが、ファイルでは 1 行で書いてください)

注: このうち「tag\_dump="off"」「read\_mode="default"」は、省略できます。

注: SAX 法で動作していることの確認は、verbose モードで、node logfile に

```
@@ structure_load_sax:read_mode= (モード名)
```

とすることで分かります。モード名は「redundant(=default)」「root\_only」(後述)「split」(後述)の3種類のうち、どれかです。

### 2.2.3 SAX 法 (2):原子数指定あり

XML パーサーとして SAX 法を使い、設定 XML ファイルに原子数を指定します。原子数を指定しない場合より、高速に読み込みます。

具体的には、設定 XML ファイルの cluster タグを下記のように書きます。

```
<cluster structure="C_liq_13824atom.xml" parser="sax" tag_dump="off"
  read_mode="default" cutoff_radius="5.0d0" number_of_atoms="13824" />
```

(上記で cluster タグは紙面横幅の都合で 2 行に渡って書いてありますが、ファイルでは 1 行で書いてください)

注: このうち「tag\_dump="off"」「read\_mode="default"」は、省略できます。原子数が正しくないと、エラーになります。

注: 原子数を指定することで高速に読み込む原理を説明します。実は、原子数を指定しない場合は、原子数のカウントアップのため、構造 XML を「2 度読み」しています。設定 XML ファイルで原子数を与えることで、構造 XML の読み込みを 1 度だけにすることができ、これにより高速化します。

注: 原子数が構造 XML ファイルで指定されていることの確認としては、verbose モードで、node logfile に、

```
INFO: number of atoms found in the config file:   = (原子数)
```

のように出ていることです。

### 2.2.4 SAX 法 (3):原子数指定あり、「root-only」モード

これは、MPI 並列計算において、ファイル読み込みを「root-only」モードで行う場合に指定します。この場合、下記のような動作になります。

```
root ノード (0 番目のノード)  構造ファイルを読む (原子構造を読み込む)
上記以外のノード             構造 basic ファイルを読む (原子構造は読み込まない)
```

設定 XML ファイルで原子数を指定することは、必須です。また、構造 basic ファイルが存在することも必須です。

具体的には、設定 XML ファイルの cluster タグを下記のように書きます。

```
<cluster structure="C_liq_13824atom.xml" parser="sax" tag_dump="off"
  read_mode="root_only" cutoff_radius="5.0d0" number_of_atoms="13824" />
```

(上記で cluster タグは紙面横幅の都合で 2 行に渡って書いてありますが、ファイルでは 1 行で書いてください)

注：このうち「tag\_dump="off"」は、省略できます。

注：前節のように「read\_mode="default"」とした場合、MPI 並列計算においては、全ノードが構造ファイルを（冗長に）読み込みます。これを「redundant」モードと呼びます。「root-only」モードの方が、「redundant」モードより高速であることが期待されますが、あまり変わらない場合も多いようです。

## 2.3 分割 XML ファイルによる並列化ファイル読み込み

分割された構造 XML ファイルを用意することで、ファイル読み込みを並列化します。下記では、液体 Carbon13824 原子系 (sample/C\_liq\_13824atom) を 4 分割する場合の説明をします。

1. 分割された構造 XML ファイルの用意 …… 前章で説明した方法により、分割された構造 XML ファイルを用意します。(MPI 並列数)=(分割ファイル) とすると、最大の効率が期待できます。
2. 設定 XML ファイルの書き換え …… 設定 XML ファイルで、system タグを下記のように書きます；

```
<system>
  <cluster structure="C_liq_13824atom.xml" parser="sax" tag_dump="off"
    read_mode="split" cutoff_radius="5.0d0" number_of_atoms="13824" />
  <splitted_input_file number_of_files="4" />
  <temperature unit="kelvin"> 6000.0 </temperature>
  <element name="C" model="geno" filename="C.xml" />
</system>
```

注：上記で cluster タグは紙面横幅の都合で 2 行に渡って書いてありますが、ファイルでは 1 行で書いてください。

下記にポイントを書きます:cluster タグの属性として、「parser="sax"」「read\_mode="split"」「number\_of\_atoms="(総原子数)"」は、必須です。下記のように、分割ファイル数を指定するのも必須です。

```
<splitted_input_file number_of_files="(分割ファイル数)" />
```

3. 実行 …… 通常通り、実行します。

```
> mpirun -n 4 ../../bin/elses ./config.xml > log.txt &
```

## 2.4 分割 XML ファイルによる並列化ファイル書き出し

前節の format で、ノードごとに分割 XML ファイルを書き出します。設定 XML ファイルで、restart タグを下記のように書きます。

最低限の設定：

```
<restart filename="restart.xml" interval="2" split="on" />
```

詳しい設定例：

```
<restart filename="restart.xml" interval="2" split="on" number_of_split_files="4" />
```

ここでは後者の例で内容を説明します。split 属性は、分割書き出しの on/off の設定で、非指定は「off」に相当します。number\_of\_split\_files 属性は分割ファイル数で、非指定時は「分割ファイル数=MPI 並列度数」に設定されます。この例だと、

```
restart_000000.xml  
restart_000001.xml  
restart_000002.xml  
restart_000003.xml
```

の 4 つのファイルができます。

注：内部動作は、以下のようになっています。

例 1：(MPI 並列度数)=4, (ファイル分割数)=4

```
0 番目のノード  restart_000000.xml を出力。  
1 番目のノード  restart_000001.xml を出力。  
2 番目のノード  restart_000002.xml を出力。  
3 番目のノード  restart_000003.xml を出力。
```

例 2：(MPI 並列度数)=2, (ファイル分割数)=4

```
0 番目のノード  restart_000000.xml, restart_000002.xml を出力。  
1 番目のノード  restart_000001.xml, restart_000003.xml を出力。
```

例 3：(MPI 並列度数)=8, (ファイル分割数)=4

```
0 番目のノード  restart_000000.xml を出力。  
1 番目のノード  restart_000001.xml を出力。  
2 番目のノード  restart_000002.xml を出力。  
3 番目のノード  restart_000003.xml を出力。  
4 番目以降のノード  何もしない
```

## 2.5 ELSESES 本体による XML ファイルの分割・統合

ELSESES 本体の「File conversion モード」(QuickStart 参照)を用いることで、XML ファイルの(再)分割・統合ができます。具体的には、読み込み(2.3 節)と書き出し(2.4 節)とで分割数を変えます。「File conversion モード」は入力データ(原子構造)をそのまま出力するモードですので、restart 用 XML ファイルとして出力されたファイルが、分割数の異なる XML ファイルになります。

elses-xml-generate では

```
XYZ ファイル  XML ファイル(分割・非分割)
```

を行いますが、上記方法では

```
XML ファイル(分割・非分割)  XML ファイル(分割・非分割)
```

となります。

## 2.6 XML ファイル出力の 3 つのモードと継続計算

XML ファイル出力には 3 種のモードがあり、有限温度ダイナミクス継続計算などに影響する (Ref:ELSES 更新記録:2016/8/15;2011/02/28)。

具体的には、設定 XML ファイルの restart tag に「mode」属性を追加する。「overwrite」「sequential」「append」「default」が指定できる。明示的に「default」と指定したり、mode 属性を指定しない場合は、「overwrite」が仮定される。以下、典型例と共に説明する。

場合 1 … overwrite (上書き) mode

```
<restart filename="restart.xml" interval="10" mode="overwrite" />
```

単一 XML ファイル (restart.xml) に上書きされる。このファイルは、ELSES の入力 XML ファイルとして、そのまま使える。

場合 2 … sequential (連番出力) mode

```
<restart filename="restart.xml" interval="10" mode="sequential" />
```

連番 XML ファイルに記録される。

```
restart_step00000000.xml  
restart_step00000010.xml  
restart_step00000020.xml  
(以下同様)
```

それぞれのファイルは、ELSES の入力 XML ファイルとしてそのまま使える。

場合 3 … append (追記) mode (非推奨)

```
<restart filename="restart.xml" interval="10" mode="append" />
```

単一 XML ファイル (restart.xml) に追記される。これを追記型 XML ファイルと呼称している。ELSES の入力 XML ファイルとして、そのままでは使えない。具体的には、出力タイミング (上記では 10 iteration ごと) で、restart XML ファイルは追記される。たとえば、全体で 30 iteration の計算だったとすると、以下になる；

```
<?xml version="1.0" encoding="UTF-8"?>  
<structure_history>  
<structure name="test" mdstep=" 0"> ~ </structure>  
<structure name="test" mdstep="10"> ~ </structure>  
<structure name="test" mdstep="20"> ~ </structure>  
<structure name="test" mdstep="30"> ~ </structure>  
</structure_history>
```

計算結果を ELSES の入力ファイルにしたい場合は、上記ファイルの一部を手動で編集し、1 つの snapshot のみの XML ファイルにする必要がある。例えば、30 iteration 目の snapshot を使いたい場合は、以下のファイルを作る必要がある。

```
<?xml version="1.0" encoding="UTF-8"?>
<structure name="test" mdstep="30"> ~ ~ ~ </structure>
```

これは、sequential mode で作られるファイルと等価である。手動編集が必要な現状は、不便な仕様と言わざるを得ない。なお、実用的立場からは、追記型 XML ファイルは巨大になる可能性があるので、sequential mode で出力することを推奨する。

注：以前からある「append\_mode」属性は、legacy 属性として、非推奨 (廃止予定)。例えば、

```
<restart filename="restart.xml" interval="10" append_mode="on" />
```

は、「場合 3」のように「mode="append"」と記述すべき。

注：追記型 XML ファイルの「不便な仕様」(上記参照)の根源は、XML ファイルが「root tag は 1 つしか許さない」という仕様であるため。例えば、

```
<?xml version="1.0" encoding="UTF-8"?>
<structure name="test" mdstep=" 0"> ~ </structure>
<structure name="test" mdstep="10"> ~ </structure>
<structure name="test" mdstep="20"> ~ </structure>
<structure name="test" mdstep="30"> ~ </structure>
```

というファイルは、root tag(structure タグ) が 4 つあるので、XML の仕様を満たしていない。ELSES XML ファイルの仕様を決めたとき、このことを気づいていなかった。あとから気づいて、複数の snapshot を格納する場合用に、structure\_history タグを策定したが、XML ファイルの読込ルーチンが未対応となっているのが、現状。

### 3 物理量出力の新機能

#### 3.1 波動関数 (LCAO 係数) 書き出しにおける新機能

##### 3.1.1 連続書き出し機能

波動関数 (LCAO 係数) データの連続保存を可能にしました。(従来は、最後の iteration のみ)。ただし、この機能は GENO 対角化 (非 DST) ワークフローでのみ利用可です。

詳細: wavefunction タグで、下記のように interval を指定します。

```
<wavefunction filename="output_wavefunction.txt" interval="10" />
```

結果として、MD step 数が入ったファイルに保存されます。

```
output_wavefunction_00000000.txt  
output_wavefunction_00000010.txt  
output_wavefunction_00000020.txt
```

ただし、最終 MD step では、(interval の値に関わらず) 必ず書き出されます。

また、interval 属性を省略するか、(interval 値)=0 と指定することで、(従来通り) 最終 MD step のみで保存され、そのときのファイル名は (従来通り) 「output\_wavefunction.txt」になります。

##### 3.1.2 参考:波動関数 (LCAO 係数) ファイルの仕様

(この節の内容は、新しい内容ではない)。wavefunction タグで出力されるファイル (典型名: wavefunction.txt、output\_wavefunction.txt) の仕様を記す。文頭から順番に説明する。

冒頭の 4 行では、計算セルの情報が記されている。例えば、

12	30		
18.89726878	0.00000000	0.00000000	0
0.00000000	18.89726878	0.00000000	0
0.00000000	0.00000000	18.89726878	0

1 行目は、原子数 (上記では 12) ・総基底数 (上記例では 30)。2 行目は、基本ベクトル (a.u. 単位) ・周期境界条件の有無 (1 なら周期。0 なら非周期)。3,4 行目も、同様。

次には、原子 index ごとに、原子基底情報が記されている。sp 基底系と spd 基底系で形式が異なるが、ここでは sp 基底系のみ説明する。1 つの原子に対して、下記 6 行が書かれる。原子数が  $N$  個の場合、全体で  $6N$  行となる。

6					
	4				
2	2	2	2		
0	1	1	1		
1.710E+00	1.625E+00	1.625E+00	1.625E+00		
2.645617628884097		0.0000000000000000		0.0000000000000000	

1 行目は、原子番号を表す (上記例では 6。すなわち、原子が炭素であることを示す)。2 行目は、基底数を表す (上記例では 4)。3 行目は、それぞれの基底に対して主量子数を表す。4 行目は、それぞれの基底に対して角運動量を表す (上記例では s, px, py, pz 基底であることを示す)。5 行目は、それぞれの基底に対して  $\zeta$  を表す。6 行目は、原子座標 (デカルト座標) を表す (単位は a.u. )。基底は、ここに記述される順番に、(基底 index)=1, 2, 3, ...,  $M$ 、となる。



次には、level index ごとに、固有ベクトル情報が記されている。基底数が  $M$  個の場合は、1 つの level に対して  $M$  個のデータが以下のように並ぶ。全体では、 $M^2$  行となる。以下では、level index:  $k=1$ 、 $M=30$  の例を示す。

1	-0.278631060206629	k=	1
2	0.008177411500531	k=	1
3	0.000000414215943	k=	1
4	0.000000000000000	k=	1
5	-0.278630297655781	k=	1
6	0.004088919816535	k=	1
7	0.007081875022805	k=	1
8	0.000000000000000	k=	1
9	-0.278631809651751	k=	1
10	-0.004088447284664	k=	1
11	0.007081848796670	k=	1
12	0.000000000000000	k=	1
13	-0.278631060206629	k=	1
14	-0.008177411500530	k=	1
15	-0.000000414215943	k=	1
16	-0.000000000000000	k=	1
17	-0.278630297655780	k=	1
18	-0.004088919816535	k=	1
19	-0.007081875022805	k=	1
20	0.000000000000000	k=	1
21	-0.278631809651750	k=	1
22	0.004088447284664	k=	1
23	-0.007081848796670	k=	1
24	-0.000000000000000	k=	1
25	-0.035531680133322	k=	1
26	-0.035532238704354	k=	1
27	-0.035531486593656	k=	1
28	-0.035531680133322	k=	1
29	-0.035532238704353	k=	1
30	-0.035531486593656	k=	1

### 3.2 H,S 行列および固有エネルギー書き出しにおける新機能

Hamiltonian, overlap matrices, eigen level の出力。XML ファイルからの制御ができるようにした。Hamiltonian, overlap matrices は、MatrixMarket 形式で出力する。現状では、GENO で、対角化モードだけに対応。

output タグ内に追加する。例：

```
<matrix_hamiltonian filename="output_hamiltonian.txt"
  mode="last" format="MatrixMarket_sym" unit="a.u." />
<matrix_overlap      filename="output_overlap.txt"
  mode="last" format="MatrixMarket_sym" />
<eigen_level         filename="output_eigen_levels.txt" mode="last" format="level_only" unit="a.u." />
```

(1,2 行目、3,4 行目は、実際のファイルファイルでは、それぞれ 1 行で書く)

詳細説明：

(1) Hamiltonian 行列

mode 属性：文字：last, periodic, default(=last), on(=last), off

last:最後の iteration のみ出力

periodic:周期的に出力。周期は interval 属性で指定する。

off:発動せず。

interval 属性：整数

mode 属性="periodic"のときの、周期を指定。

format 属性：文字："MatrixMarket\_sym" "MatrixMarket\_gen"

MatrixMarket\_sym : symmetric 形式 (下三角成分のみ出力)

MatrixMarket\_gen : general 形式。非対角要素  $H(i,j)$  と  $H(j,i)$  を冗長に出力

unit 属性：文字：a.u., eV

append\_mode 属性：on, off

追記モードの on/off

(2) Overlap 行列

unit 属性をのぞいて、(1) と同じ仕様。

(3) eigen level

下記以外は (1) と同じ仕様。

format 属性：文字："level\_only" "full"

level\_only : 1 行には、level index (k)、固有エネルギー ( $E_k$ )、が出力。例：

1	-1.26478100744643384701
2	-0.68245286285899264822
3	-0.61888812899997081018
4	-0.58346440488615181064
5	0.22498890821334280310
6	0.30408092998916885774

full : 1 行には、MD step count、level index (k)、固有エネルギー ( $E_k$ )、occupation ( $f_k$ )、Participation R

0	1	-31.02031481645822000000	1.00000000000000000000	29.772053652065180
0	2	-30.98604566671983000000	1.00000000000000000000	35.528846690558470
0	3	-30.96951847298839000000	1.00000000000000000000	39.711708781053850
0	4	-30.91081194931123000000	1.00000000000000000000	56.594242833921640
0	5	-30.87157234362676000000	1.00000000000000000000	56.669011095984760

### 3.2.1 H,S 行列生成のみのモード (matrix generation モード)

H,S 行列生成のみを行うためのモードとして、「matrix generation モード」がある。非 DST workflow のみに対応。  
calc タグの mode 属性を、以下のように設定すれば良い。

```
<calc mode="matrix_generation">
```

具体例が、

```
sample/sample_geno/C6H6_mat_gene
```

に入っている。C6H6\_opt で最適化された構造に対して、行列 (H,S) を生成する。行列の書き出し形式などについては、[Sec. 3.2](#) を参照。

### 3.3 局所結合エネルギー (ICOHP) の出力

局所結合エネルギー (ICOHP) 値の出力に対応した。出力形式は、連続型 (sequential)Matrix-Market 形式。本データを活かせる可視化ツールは、今のところ、星ら鳥取大グループによる可視化ツール「VisBAR」のみ。

output タグ内に追加する。例：

```
<bond_list filename="output_bond_list.txt" interval="10" />
```

ICOHP は原子ペア  $(I, J)$  に対して定義され  $(B_{IJ})$ 、電子ボルト単位で出力される。ICOHP の総和は、電子構造エネルギーを与える：

$$E_{\text{elec}} \equiv \sum_k f_k \varepsilon_k = \sum_{I,J} B_{IJ} \quad (1)$$

再生されたファイル (上記では output\_bond\_list.txt) の冒頭は、例えば、以下ようになる (128 原子系の場合)。

```
%%MatrixMarket matrix coordinate real general
% step_count=          0
      128      128      2562
      1      1      -46.4565651698
      2      1       1.4091077856
      6      1      -0.0094446481
     35      1     -18.5912248001
     36      1     -14.4578617489
     37      1     -0.0129301047
     65      1      1.0592661125
     66      1      0.8139128057
     67      1     -0.0006922386
     95      1     -0.0107046279
     96      1      0.6785227918
     97      1      0.0054212984
       2      2     -51.3983026791
       1      2      1.4091077856
```

### 3.4 波動関数グリッドデータ生成ツール:elses-generate-cubefile

波動関数グリッドデータ (Gaussian cube 形式) ファイルを生成するツール `elses-generate-cubefile` について、新機能を説明。従来機能は、QuickStart Sec.6.3 を参照。

#### 3.4.1 カットオフの導入

本ツールでは、原子中心の STO 型基底を用いており、その動径成分  $R^{(\text{STO})}(r)$  は

$$R^{(\text{STO})}(r) \equiv \sqrt{\frac{(2\zeta)^{2n+1}}{(2n)!}} r^{n-1} e^{-\zeta r} \quad (2)$$

で与えられる。ただし、原子中心を座標原点 ( $r \equiv 0$ ) とし、 $n, \zeta$  は基底の主量子数と減衰定数をそれぞれ表す。

動径成分関数に cutoff radius ( $r_{\text{cut}}$ ) を代入できるようになった。具体的には、ツールにおける動径関数  $R^{(\text{vis})}(r)$  は、

$$R^{(\text{vis})}(r) \equiv \begin{cases} R^{(\text{STO})}(r) & (r < r_{\text{cut}}) \\ 0 & (r > r_{\text{cut}}) \end{cases} \quad (3)$$

で与えられる。

使い方を説明する。ツール実行時の option として、atomic unit 単位で指定する。たとえば、

```
$ elses-generate-cubefile 15 -cutoff_au=1
```

だと、15 番目の波動関数にたいして、 $r_{\text{cut}} = 1\text{au}$  で、cubefile をつくる。「-cutoff\_au=1」の位置はどこにあっても良い。標準出力に

```
OPTION:The cutoff radius for plot [au] = 1.0000000000000000
```

と出力されるので、確認できる。

図??(a)(b) では、ベンゼン HOMO( $k = 15$  番目) 軌道における、カットオフの有無による波動関数を比較した。

```
sample/sample_geno/C6H6_opt/
```

の出力データを利用している。(a) カットオフなし、(b) カットオフあり ( $r_{\text{cut}} = 1.5\text{au} \approx 0.79\text{\AA}$ )、をそれぞれ可視化している<sup>2</sup>。C-C 間の原子間距離が  $d_{\text{CC}} \approx 1.4\text{\AA}$  であるので、波動関数が「接した球の集まり」のように描かれるのは妥当である。

#### 3.4.2 符号反転

波動関数  $\phi(\mathbf{r})$  は、

$$\phi(\mathbf{r}) \Rightarrow -\phi(\mathbf{r}) \quad (4)$$

と符号反転しても一般性を失わない。

本ツールでは、符号反転してからグリッドデータを生成することができる。ツール実行時の option として、たとえば、

```
$ elses-generate-cubefile 15 -sign_inversion
```

<sup>2</sup> VESTA v.3.1.7 による可視化。等値面レベル= $\pm 0.025$ (VESTA で用いられる単位)。VESTA メニューバーから、Objects Properties Isosurfaces、で設定できる。

と指定する。「-sign\_inversion」の位置はどこにあっても良い。省略形として「-si」でも良い。

図??(a)(c) では、ベンゼン HOMO( $k = 15$  番目) 軌道における、符号反転の有無による波動関数を比較した。

```
sample/sample_genoc/C6H6_opt/
```

の出力データを利用している。

### 3.4.3 grid region の指定

input\_mesh\_grid.txt から、grid region を指定できる。省略時の設定も含めて、以下で説明する。長さの単位は、「internal」「au」が選べる。前者は、LCAO 係数ファイル内の cell size (AX, AY, AZ) で規格化された単位のこと。

- (1) internal 単位でのフル設定例

```
80 80 80 # Number of mesh points
-0.5d0 -0.5d0 -0.5d0 internal # origin of the mesh grid region
1.0d0 1.0d0 1.0d0 internal # sizes of the mesh grid region
```

3つの並びは、 $x$  方向、 $y$  方向、 $z$  方向をあらわす。上記例では、 $\text{origin} = (-0.5, -0.5, -0.5)$  ,  $\text{size} = (1.0, 1.0, 1.0)$ 、を意味する。つまり  $x$  方向には、 $-0.5 < x < 0.5$ (internal 単位) の領域に mesh が設定される。 $y, z$  方向も同様。

- (2) au 単位でのフル設定

```
70 70 70 # Number of mesh points
-10d0 -10d0 -10d0 au # origin of the mesh grid region
25d0 25d0 25d0 au # sizes of the mesh grid region
```

上記例では、 $\text{origin} = (-10, -10, -10)$ 、 $\text{size} = (25, 25, 25)$ 、を意味する。つまり  $x$  方向には、 $-10 < x < 15$ (au 単位) の領域に mesh が設定される。 $y, z$  方向も同様。

- (3) mesh point のみの設定例：

```
70 70 70 # Number of mesh points
```

(1) の値に、grid region が設定される (従来通り)

- (4) input\_mesh\_grid.txt なし

(1) の値に、grid region, mesh point が設定される (従来通り)

### 3.4.4 入出力ファイル名のカスタム指定

LCAO 係数ファイル (入力ファイル)・cube ファイル (出力ファイル)header 部分が、カスタム指定できる。それぞれのデフォルト名は「output\_wavefunction-test.txt」「eigen\_state\_」。

(a) LCAO 係数ファイル名の指定：実行時の option として「-lcao\_coef\_file=(文字列)」で設定可能。例えば LCAO 係数ファイルが「output\_wavefunction-test.txt」だった場合、

```
$ elses-generate-cubefile -lcao_coef_file=output_wavefunction-test.txt
```

とする。この場合は、標準出力ファイルに

```
LCAO coef. file name (custom ) :output_wavefunction-test.txt
```

と出力される。指定がない場合は、「output\_wavefunction.txt」とされる。この場合は、標準出力ファイルに

```
LCAO coef. file name (default) :output_wavefunction.txt
```

と出力される。

(b) cube ファイル名の header 部分の指定：実行時の option として「-cube\_filename\_header=(文字列)」で設定可能。  
例えば、上記文字列部分が「mycube\_」だった場合、

```
$ elses-generate-cubefile -cube_filename_header=mycube_
```

とする。この場合は、標準出力ファイルに

```
cube file name header (custom) :mycube_
```

と出力される。出力される cube ファイルは、例えば、

```
mycube_000001.cube  
mycube_000001.cube
```

となる。指定がない場合は「eigen\_state\_」とされる。この場合は、標準出力ファイルに

```
cube file name header (default) :eigen_state_
```

と出力される。出力される cube ファイルは、従来通り例えば、

```
eigen_state_000001.cube  
eigen_state_000002.cube
```

となる。

### 3.5 電荷分布関数の出力

電荷分布  $n(\mathbf{r})(> 0)$  の cube ファイルを作るために、電荷の LCAO 係数ファイルをつくる。

config.xml の output タグ内に以下のように書く。例：

```
<output>
  <wavefunction_charge filename="output_wfn_charge.txt" />
</output>
```

詳細設定は、wavefunction タグと同様。例として、

```
sample/sample_geno/C6H6_opt/config.xml
```

がある。

LCAO 係数ファイルから cube ファイルを作るときは、elses-generate-cubefile を使う際に「-absolute\_value」をつける。例：

```
$ elses-generate-cubefile -absolute_value -lcao_coef_file=output_wfn_charge.txt -cutoff_au=5
                        -cube_filename_header=wfn_charge
```

注：上記 2 行は、実際には 1 行で書く。

参考 1：ここで出している電荷分布は、軌道ごとの Mulliken 電荷  $q_i$  に対して、

$$n^{(M)}(\mathbf{r}) \equiv \sum_i q_i |\chi_i(\mathbf{r})|^2 \quad (5)$$

で定義している。この積分値

$$\int n^{(M)}(\mathbf{r}) d\mathbf{r} = \sum_i q_i \int |\chi_i(\mathbf{r})|^2 d\mathbf{r} = \sum_i q_i \quad (6)$$

は、正しい全電子数を与える。ただし上記は、LCAO 波動関数の定義とは、完全には一致しない。電荷分布の形状を把握するのは、これで十分と考えられる。また、実際に cubefile に含まれる数値データは

$$\mu^{(v)}(\mathbf{r}) \equiv \sum_i q_i |\chi_i(\mathbf{r})| \quad (7)$$

で定義される。isosurface を作る際に、便利なので。

理論ノート：上記で定義した電荷分布  $n^{(\text{orb}, \text{Mul.})}(\mathbf{r})$  と、LCAO 波動関数から作った電荷分布の違いをまとめておく LCAO 波動関数

$$\phi_k(\mathbf{r}) \equiv \sum_i C_{ik} \chi_i(\mathbf{r}) \quad (8)$$

から電荷分布を作ると、

$$\begin{aligned} n^{(\text{true})}(\mathbf{r}) &\equiv \sum_k f_k |\phi_k(\mathbf{r})|^2 \\ &= \sum_k f_k \sum_{i,j} C_{ik} C_{jk} \chi_j(\mathbf{r}) \chi_i(\mathbf{r}) \\ &= \sum_{i,j} \rho_{ij} \chi_j(\mathbf{r}) \chi_i(\mathbf{r}) \end{aligned} \quad (9)$$



とかける。ただし

$$\rho_{ij} \equiv \sum_k f_k C_{ik} C_{jk} \quad (10)$$

は密度行列。一方全電子数  $N_{elec}$  は (スピン自由度 2 を無視すると)

$$N_{elec} = \text{Tr}[\rho S] = \sum_{i,j} \rho_{ij} S_{ji} \quad (11)$$

である。基底ごとの Mulliken Charge  $q_i$  は、上記で  $j$  の和だけをとったものに相当するので、

$$\begin{aligned} q_i &\equiv \sum_j \rho_{ij} S_{ji} \\ &= \sum_j \rho_{ij} \int \chi_j(\mathbf{r}) \chi_i(\mathbf{r}) d\mathbf{r} \end{aligned} \quad (12)$$

なお、「真の」電荷分布  $n^{(\text{true})}(\mathbf{r})$  を mesh data として生成したければ、mesh 上で  $|\phi_k(\mathbf{r})|^2$  を計算して、 $k$  について和を取れば良い。

### 3.6 原子基底情報 (basis info) の出力

原子基底情報 (basis information) の書き出し。

```
<basis_info filename="output_basis_information.txt" mode="first"/>
```

使い方は、設定 XML ファイルの output タグ内に書く。

- filename : 省略不可。
- mode : 「first」「last」「default(=initial)」が設定可。省略可。

指定されたファイルに basis information を出力する。ここでいう basis information とは、「output\_wavefunction ファイルにおいて、LCAO 係数をのぞいたもの」である。mode が「first」「last」時には、それぞれ、最初・最後の iteration 時に出力される。

## 4 マシン環境固有の話題

スパコンなど、マシン環境に固有の話題を載せます。Makefile.inc、job script file の典型的な記載方法などです。ただし、スパコン環境は個体差がある場合もあります。詳細は、それぞれのマニュアル等を参照してください。

### 4.1 「京」およびFX10 での ELSESES 利用

「京」および富士通 FX10 での ELSESES 利用について。

Makefile.inc の典型的な記述は、

```
FC = mpifrtpx
FFLAGS = -Kfast,openmp
LDFFLAGS = $(FFLAGS)
LIBS = -SSL2
```

である。CPU 名を、例えば FX10 の場合なら

```
FFLAGS = -Kfast,openmp,SPARC64IXfx
```

と指名することもできる。以下、分割 XML ファイルを用いた job script ファイルの例を示す。

FX10 における job script ファイルの例 1 ; ランクディレクトリを用いない場合。

```
#!/bin/bash
#PJM -L "rscgrp=short"
#PJM -L "node=8"
#PJM -L "elapsed=2:00:00"
#PJM --mpi "proc=8"
#PJM --stg-transfiles all
#PJM --stgin "./elses /*.xml ./"
#PJM --stgout "*.xyz *.txt restart*.xml ./ "
#PJM --stgout "std-file.* ./out/ "
#PJM -s
#PJM --mail-list test-test@elses.jp
#PJM -m b
#PJM -m e
#
MPI_HOME=/opt/FJSVfxlang/1.2.0
TOFU_LIB=/opt/FJSVpxtof/sparc64fx/lib64
export PATH="${MPI_HOME}/bin:${PATH}"
export LD_LIBRARY_PATH=${MPI_HOME}/lib64:${TOFU_LIB}:${LD_LIBRARY_PATH}
export THREAD_STACK_SIZE=65536
export FLIB_FASTOMP=TRUE
export OMP_NUM_THREADS=16
export PARALLEL=16
mpirexec -of-proc std-file --mca mpi_print_stats 1 lpgparm
-s 32MB -d 32MB -h 32MB -t 32MB -p 32MB ./elses -verbose ./config.xml
```

注：最後の2行は、実際のファイルでは1行で書く。

注：上記における「-of-proc std-file」という指定で、ノードごとの標準出力ファイルが

```
std-file.X (X=0-71)
```

というファイル名で出る。

FX10 における job script ファイルの例 2；ランクディレクトリを用いる場合。

```
#!/bin/bash
#PJM -L "rscunit=unit1"
#PJM -L "rscgrp=large"
#PJM -L "node=72"
#PJM -L "elapsed=24:00:00"
#PJM --mpi "proc=72"
#PJM --stg-transfiles all
#PJM --stgin "rank=* ./elses %r:./"
#PJM --stgin "rank=* ./config.xml %r:./"
#PJM --stgin "rank=* ./C_liq_13824atom_basic.xml %r:./"
#PJM --stgin "rank=* ./C_liq_13824atom_%06r.xml %r:./"
#PJM --stgout "rank=0 ./Output.txt ./out/"
#PJM --stgout "rank=0-7 ./log-node%06r.txt ./out/"
#PJM --stgout "rank=* ./restart_%06r.xml ./out/"
#PJM --stgout "rank=* ./std-file.%r ./out/"
#PJM -s
#
MPI_HOME=/opt/FJSVfxlang/1.2.0
TOFU_LIB=/opt/FJSVpxtof/sparc64fx/lib64
export PATH="${MPI_HOME}/bin:${PATH}"
export LD_LIBRARY_PATH=${MPI_HOME}/lib64:${TOFU_LIB}:${LD_LIBRARY_PATH}
export THREAD_STACK_SIZE=65536
export FLIB_FASTOMP=TRUE
export OMP_NUM_THREADS=16
export PARALLEL=16
mpiexec -of-proc std-file --mca mpi_print_stats 1 lpgparm ./elses -verbose ./config.xml
```

「京」における job script ファイルの例 1；ランクディレクトリを用いない場合。

```
#!/bin/bash -x
#
#PJM --rsc-list "node=8x8x8"
#PJM --rsc-list "elapsed=03:00:00"
#PJM --rsc-list "rscgrp=small"
#PJM --stg-transfiles all
#PJM --stgin "./elses /*.xml ./"
#PJM --stgout "*.txt fort.* ./ stgout=all"
#PJM -s
```

```
#
. /work/system/Env_base
export PARALLEL=8
export OMP_NUM_THREADS=$PARALLEL
mpiexec -n 512 -of-proc std-file --mca mpi_print_stats 1 lpgparm
-s 4MB -d 4MB -h 4MB -t 4MB -p 4MB ./elses ./config.xml
```

(最後の2行は、ファイルでは1行で書いてください)

「京」における job script ファイルの例2；ランクディレクトリを用いる場合。

```
#!/bin/bash -x
#
#PJM --rsc-list "node=8x8x8"
#PJM --rsc-list "elapsed=03:00:00"
#PJM --rsc-list "rscgrp=small"
#PJM --stg-transfiles all
#PJM --mpi "use-rankdir"
#PJM --stgin "rank=* ./elses %r:./"
#PJM --stgin "rank=* ./config.xml %r:./"
#PJM --stgin "rank=* ./FL-test-14-10629120atom-split512_basic.xml %r:./"
#PJM --stgin "rank=* ./FL-test-14-10629120atom-split512_%06r.xml %r:./"
#PJM --stgout "rank=0 ./Output.txt ./out/"
#PJM --stgout "rank=0-7 ./log-node%06r.txt ./out/"
#PJM --stgout "rank=* ./restart_%06r.xml ./out/"
#PJM --stgout "rank=* ./std-file.%r ./out/"
#PJM -s
#
. /work/system/Env_base
export PARALLEL=8
export OMP_NUM_THREADS=$PARALLEL
mpiexec -n 512 -of-proc std-file --mca mpi_print_stats 1 lpgparm
-s 4MB -d 4MB -h 4MB -t 4MB -p 4MB ./elses ./config.xml
```

(最後の2行は、ファイルでは1行で書いてください)

## 5 sample データ関連の新機能

### 5.1 sample データ検証ツール (shell script) (非 DST ワークフロー)

sample データ検証ツール (shell script) を実行する事で、計算結果が検証できる。ELSESES を make した後に、チェックとして行うことを想定している。

問題がなければ、下記のように出力される。

```
-----
ELSESES sample test
-----
TEST for C6H6_opt (GENO)
.....Total Simulation time (sec )      =                0.11080
TEST for C6H6_opt (GENO) in generating cube file
TEST for C6H6_dyn (GENO)
.....Total Simulation time (sec )      =                0.11500
TEST for VCNT_100atom (non-geo)
      (with generation of initial XML file and final eigen values)
.....Total Simulation time (sec )      =               33.59730
.....Checking the resultant eigen levels
TEST for Au_NW_0143atom (non-geo;NRL)
      (with generation of the element file)
.....Total Simulation time (sec )      =                7.21200
ELSESES sample test .... ended successfully, if no numerical data appears above
-----

ELSESES band calculation test : carbon diamond with Cerda spd orbitals
test> ../../../../bin/elseses -band -verbose ./config.xml > logfile.txt
test> ../../../../bin/band > log-band.txt
test> diff EigenEnergy.txt result/EigenEnergy.txt
---> Test ended successfully, if no numerical data appears above
```

計算時間以外に、なんらかの数値データが出た場合は問題があることになる。

詳細：現状では、

```
sample/sample_gen0/C6H6_opt
sample/sample_gen0/C6H6_dyn
sample/sample_non_gen0/Au_NW_0143atom
tool/band/samples/band-diamond-spd
```

のケースを扱う。それぞれ、ベンゼンの構造最適化 (GENO 系)、ベンゼンのダイナミクス (GENO 系)、金ナノワイヤのダイナミクス (NRL 系)、バンド図描画ツールの検証 (Sec. 5.3)、を扱う。具体的には、shell script を実行すると、それぞれのディレクトリで ELSESES を実行し、サブディレクトリにある結果ファイル (result/Output.txt) に含まれるエネルギー値と一致するのかを (diff を取る事で) 検証する。

参考：上記金ナノワイヤ系の詳細は、下記論文を参照：Y. Iguchi, T. Hoshi, T. Fujiwara, Phys. Rev. Lett. 99, 125507 (2007) (<http://dx.doi.org/10.1103/PhysRevLett.99.125507>); T. Hoshi, T. Fujiwara, J. Phys.: Condens. Matter 21, 272201, (2009) (<http://dx.doi.org/10.1088/0953-8984/21/27/272201>).

参考：スクリプト (sample/shell\_scripts/elses\_sample\_test.sh) の全文

```
#!/bin/sh
echo '-----',
echo 'ELSES sample test'
echo '-----',
echo 'TEST for C6H6_opt (GENO)'
cd sample_gen0
cd C6H6_opt
../../shell_scripts/elses_test.sh 0
echo 'TEST for C6H6_opt (GENO) in generating cube file'
../../bin/elses-generate-cubefile 1 3 > log-generete-cube.txt
head -n 100 ./eigen_state_000001.cube > eigen_state_000001.cube.header
head -n 100 ./eigen_state_000002.cube > eigen_state_000002.cube.header
head -n 100 ./eigen_state_000003.cube > eigen_state_000003.cube.header
diff ./eigen_state_000001.cube.header ./result
diff ./eigen_state_000002.cube.header ./result
diff ./eigen_state_000003.cube.header ./result
echo 'TEST for C6H6_dyn (GENO)'
cd ../C6H6_dyn
../../shell_scripts/elses_test.sh 0
echo 'TEST for VCNT_100atom (non-gen0) ',
echo '      (with generation of initial XML file and final eigen values)',
cd ../../sample_non_gen0/CNT100atom_dyn
../../bin/elses-xml-generate generate.xml CNT.xml
../../shell_scripts/elses_test.sh 0
echo 'TEST for Au_NW_0143atom (non-gen0;NRL) ',
echo '      (with generation of the element file)',
cd ../Au_NW_0143atom
../../bin/elses-xml-generate-nrl-param au_par_99.txt au_par_99.xml Au
../../shell_scripts/elses_test.sh 0
echo 'ELSES sample test .... ended successfully, if no numerical data appears above'
echo '-----',
cd ../../../../tool/band/samples/band-diamond-spd
```

注：ベンゼン最適化の計算では、cube file の生成 (elses-generate-cubefile ) コマンドも含まれている (cf. QuickStart Sec. 6.3)。

注：NRL 系金ナノワイヤの計算では、element XML file の生成 (elses-xml-generate-nrl-param ) コマンドも含まれている (cf. QuickStart Sec. 3.3)。

参考：上記ファイルから呼ばれているスクリプト (sample/shell\_scripts/elses\_test.sh) の全文

```
#!/bin/sh

if [ $# -eq 0 ];then
    echo '  USAGE ERROR: The number of nodes is required. See the document.'
    exit 1
fi
```

```

if [ $# -ge 2 ];then
    echo ' USAGE ERROR: Unsupported input is detected. See the document.'
    exit 1
fi

# Check $1 --- string or number
expr 1 + $1 >/dev/null 2>&1
if [ $? -ge 2 ];then
    echo 'USAGE ERROR: The number of nodes is required. See the document.'
    exit 1
fi

if [ $1 -eq 0 ];then
    ../../bin/elses -verbose ./config.xml > logfile.txt
else
    mpirun -n $1 ../../bin/elses -verbose ./config.xml > logfile.txt
fi

grep 'Total Simulation time (sec )' log-node000000.txt
grep 'Energy summary (eV/atom)' Output.txt > data.txt
grep 'Energy summary (eV/atom)' result/Output.txt > data_ref.txt
diff data.txt data_ref.txt
rm data.txt
rm data_ref.txt

```

## 5.2 sample データ検証ツール (shell script) (DST ワークフロー)

DST ワークフロー用の sample データ検証ツール (shell script) を実行する事で、計算結果が検証できる。ELSES を make した後に、チェックとして行うことを想定している。

例 1：標準テストをノード数 = 2 で実行する。

```

$ cd sample
$ ./shell_scripts/elses_sample_test_for_mpi.sh 2

```

注：ノード数指定(上記の「2」)は、省略できない。スレッド数は、本コマンドでは設定されず、環境変数(例:OMP\_NUM\_THREADS)で指定される。

注：以下の sample が実行される。

```

sample/sample_genompi/C_liq_01728atom_dst
sample/sample_genompi/C_liq_01728atom_dst_opt
sample/sample_genompi/C_liq_01728atom_dst_cell_change_only

```

例 1 を実行したときの結果：(2node, 6thread)

```

$ cd sample
$ ./shell_scripts/elses_sample_test_for_mpi.sh 2

```



```

ELSESES sample test for MPI: the number of nodes = 2
  test C_liq_01728atom_dst
.....Total Simulation time (sec )      =          191.78970
  test C_liq_01728atom_dst_opt
.....Total Simulation time (sec )      =          126.19260
  test C_liq_01728atom_dst_cell_change_only
.....Total Simulation time (sec )      =          157.52590
ELSESES sample test .... ended successfully, if no numerical data appears above
-----
Information of the parallelism
-----
INFO-MPI:   mpi_is_active or not :   T
INFO-MPI: True  MPI wrapper
INFO-MPI-OMP: P_MPI, P_OMP=          2          6

```

注：上記で、計算時間以外の数値データが現れていたら、問題があることを意味する。

例 2：大型テストをノード数 = 2 で実行する。

```

$ cd sample
$ ./shell_scripts/elses_sample_test_for_mpi.sh 2 large

```

注：スレッド数は、本コマンドでは設定されず、環境変数 (例：OMP\_NUM\_THREADS) で指定される。

注：以下の sample が実行される。

```

sample/sample_for_mpi/C_liq_01728atom_dst
sample/sample_for_mpi/C_liq_01728atom_dst_opt
sample/sample_for_mpi/C_liq_01728atom_dst_cell_change_only
sample/sample_for_mpi/C_liq_13824atom_dst

```

例 2 を実行したときの結果：(2node, 6thread)

```

$ cd sample
$ ./shell_scripts/elses_sample_test_for_mpi.sh 2 large
ELSESES sample test for MPI: the number of nodes = 2
Large test mode
  test C_liq_01728atom_dst
.....Total Simulation time (sec )      =          197.46070
  test C_liq_01728atom_dst_opt
.....Total Simulation time (sec )      =          125.57180
  test C_liq_01728atom_dst_cell_change_only
.....Total Simulation time (sec )      =          157.17160
  test C_liq_13824atom_dst
.....Total Simulation time (sec )      =          1529.96850
ELSESES sample test .... ended successfully, if no numerical data appears above
-----
Information of the parallelism
-----

```

```
INFO-MPI:  mpi_is_active or not :  T
INFO-MPI: True  MPI wrapper
INFO-MPI-OMP: P_MPI, P_OMP=          2          6
```

注：上記で、計算時間以外の数値データが現れていたら、問題があることを意味する。

参考：スクリプトの全文:elses\_sample\_test\_for\_mpi.sh

```
#!/bin/sh

if [ $# -eq 0 ];then
    echo ' USAGE ERROR: The number of nodes is required. See the document.'
    exit 1
fi

if [ $# -ge 3 ];then
    echo ' USAGE ERROR: Unsupported input is detected. See the document.'
    exit 1
fi

# Check $1 --- string or number
expr 1 + $1 >/dev/null 2>&1
if [ $? -ge 2 ];then
    echo 'USAGE ERROR: The number of nodes is required. See the document.'
    exit 1
fi

cd sample_genompi

echo 'ELSES sample test for MPI: the number of nodes =' $1

if [ $# -eq 2 ];then
    if [ "$2" = large ];then
        echo 'Large test mode'
    else
        echo 'USAGE ERROR: Unsupported input is detected. See the document.'
        exit 1
    fi
fi

echo ' test C_liq_01728atom_dst'
cd C_liq_01728atom_dst
../../shell_scripts/elses_test.sh $1

echo ' test C_liq_01728atom_dst_opt'
cd ../C_liq_01728atom_dst_opt
../../shell_scripts/elses_test.sh $1
```

```

echo ' test C_liq_01728atom_dst_cell_change_only'
cd ../C_liq_01728atom_dst_cell_change_only
../../shell_scripts/elses_test.sh $1

if [ $# -ge 2 ];then
  if [ "$2" = large ];then
    echo ' test C_liq_13824atom_dst'
    cd ../C_liq_13824atom_dst
    ../../shell_scripts/elses_test.sh $1
  else
    echo 'USAGE ERROR: Unsupported input is detected. See the document.'
    exit 1
  fi
fi

echo 'ELSEES sample test .... ended successfully, if no numerical data appears above'
echo '-----'
echo 'Information of the parallelism '
echo '-----'
grep 'MPI' log-node000000.txt | grep 'mpi_is_active'
grep 'MPI' log-node000000.txt | grep 'wrapper'
grep 'INFO-MPI-OMP' Output.txt

```

## 5.3 band 計算データ検証ツール (shell script)

### 5.3.1 計算手順

データ検証自動化ツール作成の基礎として、手動での計算手順を説明する。例として、Cerdá論文<sup>3</sup>にある spd 基底を用いた、ダイヤモンドのバンド図を計算する。

作業ディレクトリは、

```
tool/band/samples/band-diamond-spd/
```

とする。

- STEP(1) : Cerdá論文をみながら、element file を手動作成する。ただし、論文では s,d 軌道の c2 が明示されていないので、QuickStar Sec. 8.5 を参考に c2 を決める。

C\_spd\_Cerda.xml と名付ける。

```

<?xml version="1.0" encoding="UTF-8"?>
<element type="Huckel" name="C">
  <mass> 12.0107d0 </mass>
  <initial_charge> 0.0000 </initial_charge>
  <principal_quantum_number> 4*2 5*3 </principal_quantum_number>

```

<sup>3</sup> J. Cerdá and F. Soria, Phys. Rev. B. 61, 7965 (2000).

```

<initial_occupation> 2.0d0 3*0.6666666666666666d0 5*0.0d0 </initial_occupation>
<initial_diagonal_elements> -22.649d0 3*-14.871d0 5*-3.440d0 </initial_diagonal_elements>
<zeta> 2.125d0 3*1.269d0 5*0.906d0 </zeta>
<zeta2> 25.000d0 3*2.271d0 5*25.00d0 </zeta2>
<c1> 0.691771422701462d0 3*0.177d0 5*0.706787401819841d0 </c1>
<c2> 0.691771422701462d0 3*0.851d0 5*0.706787401819841d0 </c2>
<chemical_hardness> 14.871d0 </chemical_hardness>
<repulsive_rescaling> 9*1.00d0 </repulsive_rescaling>
</element>

```

- STEP(2) :ダイヤモンド構造のユニットセル情報ファイルを手動作成する。

unitcell.xml と名付ける。

```

<?xml version="1.0" encoding="UTF-8"?>
<structure name="sample" mdstep="0">
  <unitcell>
    <vector unit="angstrom"> 0 1.7836 1.7836 </vector>
    <vector unit="angstrom"> 1.7836 0 1.7836 </vector>
    <vector unit="angstrom"> 1.7836 1.7836 0 </vector>
  </unitcell>
  <atom element="C" class="" motion="free">
    <position unit="internal"> 0 0 0 </position>
  </atom>
  <atom element="C" class="" motion="free">
    <position unit="internal"> 0.25 0.25 0.25 </position>
  </atom>
</structure>

```

- STEP(3) :スーパーセルファイル (バンド計算用の補助ファイル) を作成するための、設定ファイルを手動作成する。

supercell.xml と名付ける。

```

<?xml version="1.0" encoding="UTF-8"?>
<structure name="sample" mdstep="0">
  <unitcell>
    <vector unit="angstrom"> 14.2688 0 0 </vector>
    <vector unit="angstrom"> 0 14.2688 0 </vector>
    <vector unit="angstrom"> 0 0 14.2688 </vector>
  </unitcell>
  <heatbath>
    <massperatom unit="a.u."> 0.2500000000000000E+02</massperatom>
    <position unit="a.u."> 0.0000000000000000E+00 </position>
    <velocity unit="a.u."> 0.0000000000000000E+00 </velocity>
  </heatbath>
</structure>

```

- STEP(4) :バンド図描画用の gnuplot 用 script を手動作成する。

GNU-Ekcurve-Diamond.plt と名付ける。

```
set term postscript portrait 'Times-Roman' 12
# set terminal postscript portrait enhanced 'Times-Roman' 12
set output 'Diamond-band.ps'
#set tics out
set title 'Diamond (C) with spd orbitals in Cerda parameters' font "Times Roman,25"
set yrange[ -40.000000: 0.0000]
set xrange[ 0.000000: 4.4238]
set style data lines
##### set size 0.9481, 0.7584
# set size 0.50, 0.385
set grid
set xtics('L' 0.0,'G' 0.80721, 'X' 1.7393,'W' 2.2053,'L' 2.8644,'K' 3.4350,'G' 4.4238)
# set xzeroaxis
set yzeroaxis
set xlabel ' k ' font "Times Roman,20"
set ylabel ' Energy E(k) eV ' font "Times Roman,25" # 1.0,0.0
set nokey
set style line 1 lw 1
plot 'EigenEnergy.txt' using 5:6 w l linestyle 1, \
'EigenEnergy.txt' using 5:7 w l linestyle 1, \
'EigenEnergy.txt' using 5:8 w l linestyle 1, \
'EigenEnergy.txt' using 5:9 w l linestyle 1, \
'EigenEnergy.txt' using 5:10 w l linestyle 1, \
'EigenEnergy.txt' using 5:11 w l linestyle 1, \
'EigenEnergy.txt' using 5:12 w l linestyle 1, \
'EigenEnergy.txt' using 5:13 w l linestyle 1, \
'EigenEnergy.txt' using 5:14 w l linestyle 1, \
'EigenEnergy.txt' using 5:15 w l linestyle 1, \
'EigenEnergy.txt' using 5:16 w l linestyle 1, \
'EigenEnergy.txt' using 5:17 w l linestyle 1, \
'EigenEnergy.txt' using 5:18 w l linestyle 1, \
'EigenEnergy.txt' using 5:19 w l linestyle 1, \
'EigenEnergy.txt' using 5:20 w l linestyle 1, \
'EigenEnergy.txt' using 5:21 w l linestyle 1, \
'EigenEnergy.txt' using 5:22 w l linestyle 1, \
'EigenEnergy.txt' using 5:23 w l linestyle 1
```

注:「set style line 1 lw 1」をいれないと、「linestyle not found」というエラーになった (gnuplot ver.4 利用)。参考 URL: <http://t16web.lanl.gov/Kawano/gnuplot/misc4.html>

- STEP(5) :バンド計算用の補助ファイルを作成する。

バンド計算用の補助ファイルとして、スーパーセル情報ファイル<sup>4</sup>(下記ではファイル名を C\_bulk\_512.xml に設定)・バンド計算用設定ファイル(ファイル名は固定: ForBandCalc.txt)、を生成する。

<sup>4</sup> 計算手順上に必要となるもので、物理的な意味はない。

```
$ ../../../../Perl/mkSupercell.pl unitcell.xml supercell.xml
```

標準出力として、下記が現れる：

```
translation vectors of unitcell (primitive translation vectors)
( 0.000000e+00, 3.370517e+00, 3.370517e+00 )
( 3.370517e+00, 0.000000e+00, 3.370517e+00 )
( 3.370517e+00, 3.370517e+00, 0.000000e+00 )
translation vectors of supercell
( 2.696413e+01, 0.000000e+00, 0.000000e+00 )
( 0.000000e+00, 2.696413e+01, 0.000000e+00 )
( 0.000000e+00, 0.000000e+00, 2.696413e+01 )
lengths of translation vectors of supercell (in atomic unit)
2.69641348735867226e+01, 2.69641348735867226e+01, 2.69641348735867226e+01
VOL(supercell)=1.960467e+04, VOL(unitcell)=7.658073e+01
VOL(supercell)/VOL(unitcell)=2.560000e+02
(Translation vector 1) = (-4, 4, 4) in the unit of primitive translation vectors
(Translation vector 2) = ( 4, -4, 4) in the unit of primitive translation vectors
(Translation vector 3) = ( 4, 4, -4) in the unit of primitive translation vectors
Structure in unitcell
Position      1 = ( 0.000000e+00, 0.000000e+00, 0.000000e+00 ) :      C
Position      2 = ( 2.500000e-01, 2.500000e-01, 2.500000e-01 ) :      C
We assume supercell is rectangular.
The minimal rectangular cell consists of      4 unitcells.
Given supercell consists of      4 x      4 x      4 minimal rectangular cells
(# of Bravais lattice points) = 4
List of Bravais lattice points (in the unit of minimal rectangular cell)
( 0.000000e+00, 0.000000e+00, 0.000000e+00 )
( 5.000000e-01, 5.000000e-01, 0.000000e+00 )
( 5.000000e-01, 0.000000e+00, 5.000000e-01 )
( 0.000000e+00, 5.000000e-01, 5.000000e-01 )
List of all unit-cell origins (in the unit of given supercell)
( 0.000000e+00, 0.000000e+00, 0.000000e+00 )
( 1.250000e-01, 1.250000e-01, 0.000000e+00 )
( 1.250000e-01, 0.000000e+00, 1.250000e-01 )
( 0.000000e+00, 1.250000e-01, 1.250000e-01 )
( 0.000000e+00, 0.000000e+00, 2.500000e-01 )
( 1.250000e-01, 1.250000e-01, 2.500000e-01 )
( 1.250000e-01, 0.000000e+00, 3.750000e-01 )
( 以下略 )
```

生成されたファイル supercell\_new.xml を、C\_bulk\_512.xml と rename する。

```
$ mv supercell_new.xml C_bulk_512.xml
```

以下の2ファイルが生成されたことになる：

```
C_bulk_512.xml
ForBandCalc.txt
```

これらはそれぞれ、スーパーセル情報ファイル、バンド計算用設定ファイル、である。

- STEP(6) :バンド図計算用の ELSESES 設定 XML ファイルを手動作成する。

config.xml と名付ける。

```
<config name="C_bulk_512atom">
<system>
  <cluster structure="C_bulk_512.xml" cutoff_radius="13.0" />
  <boundary x="periodic" y="periodic" z="periodic" />
  <temperature unit="kelvin"> 300 </temperature>
  <element name="C" model="geno" filename="C_spd_Cerda.xml"> </element>
</system>
<calc mode="dynamics">
  <genoOption>
    <CSC_method> ELSTNER </CSC_method> <CSC_max_loop_count> 0 </CSC_max_loop_count>
    <CSC_charge_convergence> 1d-3 </CSC_charge_convergence>
    <CSC_charge_mixing_ratio> 3d-1 </CSC_charge_mixing_ratio>
    <HML_constant_K> T </HML_constant_K>
    <HML_K> 2.8d0 </HML_K>
  </genoOption>
  <dynamics scheme="velocity verlet">
    <delta unit="fsec"> 1.00 </delta>
    <total unit="fsec"> 0.00 </total>
  </dynamics>
</calc>
<output>
  <restart filename="C_bulk_512atom.xml" interval="0" />
  <position filename="C_bulk_512atom.xyz" interval="0" />
</output>
</config>
```

注：重要なデータは、(相互作用 cutoff)=13 au、および、(Hamiltonian の非対角要素を決める K 値)=2.8、のみ。計算モードが「dynamics」となっているがこれはダミーで、実際の計算では無視される。

- STEP(7) :バンド図計算用の補助行列データファイル ( $H$ ,  $S$  行列の基礎データ) を生成する。

```
$ ../../../../bin/elses -band -verbose ./config.xml > & ! log.txt
```

補助行列データファイル OverlapAndHamiltonian.txt ができる。

参考：ノードログファイル (less log-node000000.txt) の冒頭は、下記のようにになっている。ここから、(相互作用 cutoff)・(Hamiltonian の非対角要素を決める K 値) が、上記設定値になっていることが確認できる：

```

@@ ELSES version 0.03.16
INFO-MPI: Dummy MPI wrapper
Date: 2013 04 23; Time: 02 45 36
INFO-MPI-OMP: P_MPI, P_OMP=          1          8
  @@ elses_xml_load_config.. is done
    Config file name = ./config.xml
    The verbose level =          1
  ...call config_load
INFO-XML:config name=C_bulk_512atom
INFO-XML-WARNING : NO tag detected : temperature
INFO-XML:XML parser= dom
INFO-XML:structure name = sample
INFO-XML:allocate structure%vatom(structure%natom) at structure_load
INFO-XML:Optional tag detected : cutoff_radius [au] = 13.00000
  INFO-XML:genoOption_load: genoOption%HML_K= 2.8000000E+00
(下略)

```

- STEP(8) :ツール band で、バンドデータ (対称線にそった固有値のデータ) を出す。

```
$ ../../../../bin/band
```

バンドデータファイル EigenEnergy.txt ができる

- STEP(9) :gnuplot を用いて、バンドデータからバンド図を描画する。

gnuplot を起動し、gnuplot のコマンドラインから

```
gnuplot> load "GNU-Ekcurve-Diamond.plt"
```

を実行する。

画像ファイル (PS 形式)Diamond-band.ps ができる。

### 5.3.2 検証ツール (shell script)

ディレクトリ tool/band/samples/band-diamond-spd で、

```
$ ../shell_scripts/elses_test_band.sh
```

を実行することで、動作確認ができる。具体的には、STEP(7)(8) を実行し、結果ファイル (EigenEnergy.txt) を参照ファイル (result/EigenEnergy.txt) と比較している。このテストは、Sec. 5.1 にあげた、統合検証ツールにも含まれている。



## 6 エラー検出機能

(主としてユーザーによる)エラーを検出する機能。

### 6.1 原子構造のチェック (距離が近すぎる原子ペアを検出)

距離が近すぎる原子ペアを検出する機能。入力座標を間違った場合など、距離が unrealistic に近すぎる原子ペアがあると計算が破綻することがある(注: 重なり行列の数値的取り扱いによる破綻、など。物理的観点から言うと、ELSEs では内殻電子を陽に考えてはいないので、原子間距離が近すぎると unphysical なことが起こりうる)。

このような状況が起こるときに備えて、原子間距離が近すぎる原子ペアを検出することができる。設定 XML ファイル (典型名: config.xml) の calc タグ内に、下記の要領で追加する。

```
<calc_check mode="on">
  <short_atom_pair_distance mode="on">
    <warning_level unit="angstrom"> 0.5 </warning_level>
    <abort_level unit="angstrom"> 0.1 </abort_level>
  </short_atom_pair_distance>
</calc_check>
```

上記設定の場合、原子間距離が 0.5 Å 以下の時に、

```
WARNING :Too short atom-pair distance: (原子ペア・原子間距離の情報)
```

というメッセージが標準出力にでる。原子間距離が 0.1 Å 以下の時は、

```
ABORT :Too short atom-pair distance: (原子ペア・原子間距離の情報)
```

というメッセージが、標準出力にでて、強制終了する。

注: 「mode="on"」というところを「mode="off"」とすると、タグが無効(タグを書いていないことと同義)となる。

注: 上記タグは、以下の設定 XML ファイルには入っている:

```
sample/C_liq_01728atom_dst/config.xml
sample/C_liq_13824atom_dst/config.xml
```

参考: abort 時のエラーメッセージの例: 標準出力に、以下のように出る。

```
WARNING :Too short atom-pair distance [au]:      208754      423886      0.8355921536
WARNING :Too short atom-pair distance [au]:      542088      503962      0.6166662834
WARNING :Too short atom-pair distance [au]:      708728      979921      0.7860016666
WARNING :Too short atom-pair distance [au]:      750326      388463      0.6672386352
WARNING :Too short atom-pair distance [au]:      875458      717887      0.9193486732
WARNING :Too short atom-pair distance [au]:      125424      144020      0.7085177206
ABORT :Too short atom-pair distance [au]:      583705      860595      0.0500336984
```

強制終了の直接要因は、上記の最後の行。意味は、「(583705 番目と 860595 番目の原子間距離) = 0.0500336984[a.u.]」という意味。

内部仕様:

config 構造体における、以下の値を利用

```

config%calc%calc_check%set : on/off の設定 (default:false)
config%calc%calc_check%short_atom_pair_distance%set : on/off の設定 (default:false)
config%calc%calc_check%short_atom_pair_distance%warning_level : warning がでるしきい値
config%calc%calc_check%short_atom_pair_distance%abort_level : abort になるしきい値

```

DST ワークフローでの検出ルーチン : set\_projection\_dst\_cell [M\_qm\_domain\_dst]

CallTree 抜粋

```

- qm_engine_dst
|   - calc_kinetic_energy
|   + qm_domain_setting_dst
|   + prep_cohp_dst
|   + initialize_charge
|   + qm_solver_gkrylov_dst
|   - qm_solver_gkrylov_dst
|     |   + proj_init_compat
|     |   - set_dst_atm_list
|     |     |   - dst_get_atm_index_range
|     |     |   - suggest_proj_radius
|     |     + set_projection_dst_cell   ここ！

```

非 DST ワークフローでの検出ルーチン : make\_booking\_list [M\_qm\_domain]

CallTree 抜粋

```

- elses_qm_engine
|   - calc_kinetic_energy
|   - qm_domain_setting
|     |   + qm_domain_ini_setting
|     |   - make_booking_list   ここ！

```

参考 : このエラー検出機能を利用しなくても、「重なり行列の数値的取り扱いによる破綻」で abort する場合は、関連する可能性がある「原子間距離が近い原子ペア」がエラーメッセージに出力される。例えば、標準出力に、以下のように出る。

```

ERROR:(calc_u_su_hu_dstm): <u|S|u>=  -1318830.77492204
ERROR in uSu: atm_index =          193263
ERROR in uSu: orb_index =           1
ERROR in uSu: kr_dim    =           16
ERROR in uSu: <u|S|u>   = -1318830.7749220361
ERROR in uSu: nna_distance=  193263    953986          0.0271031273
Abort in ELSSES : gkrylov_main_dstm [M_la_gkrylov_main_dstm]

```

強制終了の直接要因は、行列計算の破綻。が、最後の行から「(193263 番目と 953986 番目の原子間距離) = 0.0271031273 [au]」であることが分かり、ここから「原子間距離が近すぎるのが、根源的な原因である」ことが推測される。

## 7 構造作成関連ツール

### 7.1 大規模系むけ supercell 構造作成ツール

大規模系 (おおむね数万原子以上) むけ supercell 構造作成用の、実験的ツールとして

```
tool/experimental/mk_supercell_xyz/mk_supercell_xyz.f90
```

がある。<sup>5</sup> 今のところ、ELSES の make 時にはコンパイルされない。

本ツールは、cell(並進対称)ベクトルが直交している場合、つまり

$$\begin{aligned} & (AX, 0, 0)^T \\ & (0, AY, 0)^T \\ & (0, 0, AZ)^T \end{aligned} \tag{13}$$

である場合のみ、対応している。

このツールの入出力ファイルは以下であり、それぞれの例が上記ディレクトリにあがっている。

入力 (a) : original 構造ファイル (cell 情報付き XYZ 形式)

例 : diamond\_in\_cubic\_cell.xyz

ダイヤモンド構造 (8 原子からなる cubic cell) データファイル。

使うときは「structure\_org.xyz」(ファイル名固定) に改名する必要がある。

入力 (b) : supercell の拡大周期情報 (x, y, z 方向の拡大周期を整数値 (NX,NY,NZ) で指定)

例 : input\_supercell\_size.txt

NX=10, NY=40, NZ=80, の場合 (一行「10 40 80」と書いてあるだけ)。

そのまま、使う (ファイル名固定)。

出力 : supercell 構造ファイル (cell 情報付き XYZ 形式)

例 : diamond\_in\_super\_cell.xyz

(上記入力ファイルを使った場合の結果ファイル)

実際の出力ファイルは「structure\_new.xyz」となる (ファイル名固定)。

注 : 「cell 情報付き XYZ 形式ファイル」とは、1 行目に原子数 N と cell 情報 (Angstrom 単位) が 1 行目に、  
N AX AY AZ

と記されているファイル形式である (cf. QuickStart Sec. 3.1.11)。

以下、上記データファイルを用いた作業手順を述べる。

(1) original 構造ファイルを、cell 情報付き XYZ ファイルとして用意する。一般的には、手作業 (または他ソフト) で生成する。上記ディレクトリはサンプルデータとして、

```
diamond_in_cubic_cell.xyz
```

がある。

参考 : 上記サンプルデータを作成する simple なツールとして

```
mk_diamond_cubic_cell.f90
```

<sup>5</sup> Supercell 構造作成ツールとしては tool/Perl/mkSupercell.pl がある (QuickStart Sec. 6.1)。しかし、大規模系では (メモリ・計算時間が巨大になり) 現実的でなくなる。そのような場合でも使えるツールの需要があったので、暫定的に提供。

が、上記ディレクトリにある。コンパイルして実行 (入力ファイルなし) するだけ。

(2) supercell の超周期構造の情報が記載されたファイル

```
input_supercell_size.txt
```

も入力ファイルとして必要。

(3) Supercell 構造作成ツールは、

```
mk_supercell_xyz.f90
```

である。これを単体でコンパイルする。例：

```
ifort mk_supercell_xyz.f90
```

以下では、できた実行ファイルを a.out とする。

(4) 以下のように実行する

```
cp diamond_in_cubic_cell.xyz structure_org.xyz
./a.out
```

(5) 標準出力に、以下ができる。

```
Input file : input_supercell_size.txt
-->  nx, ny, nz  =      10      40      80
Input file : structure_org.xyz
-->   N          =        8
-->  ax,ay,az [A] =      3.5479998469      3.5479998469      3.5479998469
Output file : structure_new.xyz
-->   N          =    256000
-->  ax,ay,az [A] =    35.4799984686    141.9199938742    283.8399877484
```

(6) 実行後、structure\_new.xyz というファイルができています。

冒頭は、下記のようになる：

```
      256000      35.4799984686      141.9199938742      283.8399877484
# generated by mk_supercell_xyz.f90 (build 20130402)
C      0.00000000000000000000      0.00000000000000000000      0.00000000000000000000
C      0.88699996171382866628      0.88699996171382866628      0.88699996171382866628
C      1.77399992342765733255      1.77399992342765733255      0.00000000000000000000
C      2.66099988514148577678      2.66099988514148577678      0.88699996171382866628
C      1.77399992342765733255      0.00000000000000000000      1.77399992342765733255
C      2.66099988514148577678      0.88699996171382866628      2.66099988514148577678
C      0.00000000000000000000      1.77399992342765733255      1.77399992342765733255
C      0.88699996171382866628      2.66099988514148577678      2.66099988514148577678
C      3.54799984685531466511      0.00000000000000000000      0.00000000000000000000
C      4.43499980856914355343      0.88699996171382866628      0.88699996171382866628
```

(7) structure\_new.xyz の cell 情報 (AX, AY, AZ) は引き継がれないので、補助 XML ファイル (典型名: generate.xml) に自分で書き込む (QuickStart Sec. 3.5.2 参照)。上記ディレクトリにある、

```
generate.xml
```

が、該当ファイルになっている。generate.xml の全文は、以下：

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml version="1.0" encoding="UTF-8"?>
<generate name="structure_new">
  <unitcell>
    <vector unit="Angstrom"> 35.4799984686    0.0          0.0          </vector>
    <vector unit="Angstrom">  0.0          141.9199938742    0.0          </vector>
    <vector unit="Angstrom">  0.0          0.0          283.8399877484    </vector>
  </unitcell>
  <cluster structure="structure_new.xyz">
  </cluster>
</generate>
```

(8) elses-xml-generate を使って、XML ファイルを作る：

```
../../../../bin/elses-xml-generate generate.xml structure_new.xml
```

結果として、XML ファイル

```
structure_new.xml
```

が生成される。

## 7.2 分子凝集系むけ group ID 設定ツール

結合で結ばれた原子の集団（分子など）を定義することで、group ID を設定する実験的ツール。現在、周期系には非対応。下記ディレクトリにあり、今のところ、ELSES の make 時にはコンパイルされない。

```
tool/experimental/set_group_id
```

例として、上記ディレクトリにはベンゼンダイマーのデータが入っている。ベンゼン分子が 1 グループとなるため、全体で 2 グループになる。上記ディレクトリのファイル構成は、下記となる：

```
set_group_id.f90
benzene_dimer.xyz
bond_info.txt
```

以下は使い方の説明。

- (1) 原子構造データ (XYZ 形式) を用意する。benzene\_dimer.xyz が例として入っている。
- (2) bond list ファイルを用意する。結合判定をする原子間距離を設定する。bond\_info.txt が例として入っていて、その内容は下記である：

```
3
C H 1.3d0
C C 1.6d0
C O 1.6d0
```

1 行目の「3」は、以下に 3 つの記述があることを意味する。2 行目以降は、結合判定をする距離を Å 単位で設定している。たとえば、C-H なら距離が 1.3 Å 以下なら「結合あり」と判定する。なお、この例（ベンゼンダイマー）では O を含まないので 3 行目は不要。

- (3) compile と実行。

```
$ gfortran set_group_id.f90
$ ./a.out benzene_dimer.xyz bond_info.txt result.txt
```

出力ファイルとして、result.txt ができる。result.txt に各原子ごとの group id が入っている。これが、elses-xml-generate の入力ファイル (input\_group\_id\_setting.txt) になる。result.txt の内容は、下記である：

```
# Sample
# number of atoms, number of groups, maximum number of group atoms
24 2 12
#
1 1
2 1
3 1
4 1
5 1
6 1
7 1
8 1
9 1
```

```
10 1
11 1
12 1
13 2
14 2
15 2
16 2
17 2
18 2
19 2
20 2
21 2
22 2
23 2
24 2
```

なお、実行時の標準出力は、下記である：

```
@@ get_bond_info
total number of bond info = 3
C H 1.3000000000000000
C C 1.6000000000000001
C O 1.6000000000000001
INFO:get_basic_info: filename=benzene_dimer.xyz
noa= 24
INFO:set_hop_mat only for the non-periodic case
i, num. of bonds= 1 H 1
i, num. of bonds= 2 C 3
i, num. of bonds= 3 C 3
i, num. of bonds= 4 C 3
i, num. of bonds= 5 C 3
i, num. of bonds= 6 C 3
i, num. of bonds= 7 C 3
i, num. of bonds= 8 H 1
i, num. of bonds= 9 H 1
i, num. of bonds= 10 H 1
i, num. of bonds= 11 H 1
i, num. of bonds= 12 H 1
i, num. of bonds= 13 H 1
i, num. of bonds= 14 C 3
i, num. of bonds= 15 C 3
i, num. of bonds= 16 C 3
i, num. of bonds= 17 C 3
i, num. of bonds= 18 C 3
i, num. of bonds= 19 C 3
i, num. of bonds= 20 H 1
```

```
i, num. of bonds= 21 H 1
i, num. of bonds= 22 H 1
i, num. of bonds= 23 H 1
i, num. of bonds= 24 H 1
INFO:set_group_id
id, i_seed= 1 1
count= 1 2
count= 1 4
count= 1 8
count= 1 11
count= 1 12
count= 1 12
id, i_seed= 2 13
count= 2 2
count= 2 4
count= 2 8
count= 2 11
count= 2 12
count= 2 12
INFO:save_groupid: filename=result.txt
Number of groups= 2
INFO:mem_group = 1 12
INFO:mem_group = 2 12
```

参考 : CallTree

```
- make_group
- main
- get_info_from_xyz_file
- set_hop_mat
| - set_cutoff
| - get_distance
- set_group_id
| - find_vacant_comp
| - normalize_vect
- save_group_id
```



## 8 van der Waals 力

### 8.1 概要とサンプルデータ

Ortmann (2006)[3] にあるポテンシャルを用いて、van der Waals (vdW) 力を利用することができる。サンプルデータが、下記ディレクトリにある：

```
sample/sample_geno/vdW_C6H6_opt
sample/sample_geno/vdW_C6H6_dimer
```

以後「サンプルディレクトリ」と呼ぶ。前者は、ベンゼンモノマーの構造最適化計算である。後者は、ELSESES の cell-change-only モードを利用し、サンドイッチ型ダイマー (モノマー面に対して垂直に平行移動した 2 つのモノマー)[3] において、面間距離  $D$  を、 $1.40 \text{ \AA} < D < 7.00 \text{ \AA}$  の範囲で変化させている。

### 8.2 理論の詳細

原論文 [3] からの理論式を抜粋する：

$$E_{\text{vdW}} \equiv \frac{1}{2} \sum_{i \neq j}^{\text{atom}} \varepsilon_{ij}^{\text{vdW}} \quad (14)$$

$$\varepsilon_{ij}^{\text{vdW}} \equiv -f_{ij}(R) \frac{C_6^{ij}}{R^6} \quad (15)$$

$$C_6^{ij} \equiv \frac{3}{2} \alpha_i \alpha_j \frac{I_i I_j}{I_i + I_j} \quad (16)$$

$$f_{ij}(R) \equiv 1 - \exp \left( -\lambda \left( \frac{R}{d} \right)^n \right) \quad (17)$$

$$d \equiv r_i^{(\text{cov})} + r_j^{(\text{cov})} \quad (18)$$

元素によらない無次元パラメーターとして  $n, \lambda$  があり、原論文 [3] では  $n = 8, \lambda = 7.5 \times 10^{-4}$  となっている  $n = 8$  は原点付近 ( $R \approx 0$ ) で  $R$  の 2 乗になるように決められている ( $\varepsilon_{ij}^{\text{vdW}} \propto R^2$ )。ELSESES では  $n$  は  $n = 8$  で固定し、 $\lambda$  は設定 XML ファイルで設定可能。上記サンプルディレクトリでは、 $\lambda = 1.4 \times 10^{-5}$  を用いている。具体的には、calc/genoOption タグ内に、以下のように記述する；

```
<van_der_Waals mode="on">
  <vdW_lambda mode="on"> 1.4d-5 </vdW_lambda>
</van_der_Waals>
```

具体例は、上記サンプルディレクトリの config.xml を参照。実際に使われている値は、verbose モードで ELSESES を実行し、node log ファイルの冒頭付近に下記のように記されていることで確認できる：

```
INFO-XML: optional tag: van_der_Waals (vdW) = T
INFO-XML: optional tag: vdW_lambda = 1.4000000000000000E-005
```

具体例は、上記サンプルディレクトリの result/log-node000000.txt を参照。

さらに、各元素ごとに、以下の 3 パラメーターを設定する必要がある。

- ionization energy : 上記式の  $I_i, I_j$
- dipole polarizability : 上記式の  $\alpha_i, \alpha_j$

- covalent radius : 上記式の  $r_i^{(\text{cov})}, r_j^{(\text{cov})}$

ELSES では、原子番号 84(Po) まで default 値が設定されている。85 番以上だと、エラーで abort する。この default 値は、原論文 [3] と同じく、CRC 社刊行の書籍「CRC Handbook of Chemistry and Physics」[4] の値による。実際に使われている値は、verbose モードで ELSES を実行し、node log ファイルの冒頭付近に下記のように記されていることで確認できる：

```
INFO-vdW:element name=C
INFO-vdW:atomic_num=      6
INFO-vdW:ionization_energy unit =   11.260300000000000      eV
INFO-vdW:ionization_energy [au] =   0.413805141924767
INFO-vdW:dipole_polarizability unit =  1.6700000000000000E-024 cm3
INFO-vdW:dipole_polarizability [au] =   11.2697320805955
INFO-vdW:covalent_radius unit =   0.7500000000000000      Angstrom
INFO-vdW:covalent_radius [au] =   1.41729515833077
```

具体例は、上記サンプルディレクトリの result/log-node000000.txt を参照。

上記 3 つ量の default 値は、ソースファイル

```
src/elses-lib-db-ion-ene.f90
src/elses-lib-db-polarization.f90
src/elses-lib-db-covalent-radius.f90
```

に、データベース型ルーチンとしてそれぞれ格納されている。

手動で設定したい場合は、element XML ファイルに書く。書式は、

```
sample/sample_genovdW_C6H6_opt/C_vdW_unused.xml
```

を参照。<sup>6</sup>

Default 値についての、いくつかのコメントを述べる：

- vdW を有効にしたうえで verbose モードで ELSES を実行すると、default 値の一覧表を output\_vdW\_parameter\_db.txt に出力する。
- 以下の元素の polarization 値は、書籍には複数の値が記されている：He, Na, Mg, K, Va, Cu, Zn, Sr, Ag, Cd, In, Sn I, Hg, Tl, Pb。ELSES default 値には書籍で一番上に記載されているデータを収録している。
- データベース型ルーチンとしては、原子番号 86(Rn) まで数値が入っている。ただし、原子番号 85(At, アスタチン) の ionization energy が書籍になかったため、そこだけはダミー値 (-1.0d0) を返すようになっている。ELSES で使う場合は、85 番以降が呼ばれると (データベース型ルーチンを呼ぶ前に) エラーでプログラムがストップするので、このダミー値が計算で使われることはない。
- これらの値は vdW 項のみに使われている。GENO 系での CSC 項 formulation に現れる「Chemical Hardness」は default では ionization energy の値が設定されているが、これらは src/elses-qm-genovdW-Huckel-atom-parameters.f90 で設定されている。

<sup>6</sup> この element XML file は参考のために置かれているだけで、同ディレクトリの計算実行には使われていない。

## 9 Group ID 機能

### 9.1 概要とサンプルデータ

原子をいくつかの「group」に分けて考えることができる。ここで各原子は、ひとつの group のみに属している。group の物理的意味は任意だが、典型的には分子凝集体計算において、各分子を group と見立てる。

以下の条件が揃っている必要がある。

- 構造 XML ファイルに group ID が記述されている (詳細以下)。
- 設定 XML ファイルで、calc タグ内に、

```
<use_group_id mode="on" />
```

の記載がある。注：当該タグの記載がない、または、

```
<use_group_id mode="off" />
```

と記載されると、無効になる。

ベンゼンダイマーの例が sample ディレクトリ

```
sample/sample_genovdW_C6H6_dimer_cco
```

2つの分子に属する各原子がそれぞれ (group\_id:=1,2) となっている。生成された構造 XML ファイルの冒頭 (空行は一部削除してある)

```
<?xml version="1.0" encoding="UTF-8"?>
<structure name="benzene_dimer" mdstep="0">
  <total_number_of_atoms> 24 </total_number_of_atoms>
  <unitcell>
    <vector unit="a.u."> 0.188972687777436E+02 0.000000000000000E+00 0.000000000000000E+00</vector>
    <vector unit="a.u."> 0.000000000000000E+00 0.188972687777436E+02 0.000000000000000E+00</vector>
    <vector unit="a.u."> 0.000000000000000E+00 0.000000000000000E+00 0.188972687777436E+02</vector>
  </unitcell>
  <boundary x="nonperiodic" y="nonperiodic" z="nonperiodic" />
  <heatbath>
    <massperatom unit="a.u."> 0.200000000000000E+02</massperatom>
    <position unit="a.u."> 0.000000000000000E+00 </position>
    <velocity unit="a.u."> 0.000000000000000E+00 </velocity>
  </heatbath>
  <group_id_is_added>
    <number_of_groups> 2 </number_of_groups>
    <max_number_of_group_atoms> 12 </max_number_of_group_atoms>
  </group_id_is_added>
  <atom element="H" id="1" group_id="1" motion="free">
    <position unit="internal"> -0.106834237290000E+00 0.000000000000000E+00 0.000000000000000E+00</position>
  </atom>
  <atom element="C" id="2" group_id="1" motion="free">
```

```
<position unit="internal"> -0.802754360000000E-03  0.000000000000000E+00  0.000000000000000E+00</position>
</atom>
```

group ID 付きの構造 XML ファイルを作る手順を説明する。group ID を指定するための補助ファイルを用意する必要がある。sample ディレクトリにある input\_group\_id\_setting.txt が例である。

```
# C6C6 dimer
# number of atoms, number of groups, maximum number of group atoms
24 2 12
#
1 1
2 1
3 1
4 1
5 1
(中略)
16 2
17 2
18 2
19 2
20 2
21 2
22 2
23 2
24 2
```

上記で「#」から始まる行はコメント行で、何行書いても (ただし上限は 1000 行) 無視される。コメントでない 1 番最初の行

```
24 2 12
```

は、総原子数 (ベンゼン 2 個分で 24 原子)・総 group 数 (ダイマーなので 2)・最大のグループ内原子数 (グループはベンゼン 1 個分 (C6H6) なので 12 個) をそれぞれ意味する。後の行は、各原子について、atom id, group id, がなっている。たとえば、

```
17 2
```

という行は、17 番目の原子 (atom\_id=17) が、2 番目の group に属する (group\_id=2) ことを意味する。上記情報を構造 XML ファイルに反映させる設定が、sample ディレクトリにある generate.xml で例示されている

```
<?xml version="1.0" encoding="UTF-8"?>
<generate name="benzene_dimer">
  <unitcell>
    <vector unit="angstrom"> 10.0 0.0 0.0 </vector>
    <vector unit="angstrom"> 0.0 10.0 0.0 </vector>
    <vector unit="angstrom"> 0.0 0.0 10.0 </vector>
  </unitcell>
```

```

<cluster structure="benzene_dimer.xyz" >
</cluster>
<boundary mode="on" x="nonperiodic" y="nonperiodic" z="nonperiodic" />
<atom_id_is_added mode="on" />
<group_id_is_added mode="on" file="input_group_id_setting.txt" />
<heatbath>
  <massperatom unit="a.u." 20.0d0</massperatom>
</heatbath>
</generate>

```

ここで「mode="on"」とあるところを「mode="off"」とすると、タグが無効になる。つまりタグを書かないのと同義になる。

注：group ID を restart XML に出力させるには、設定 XML ファイル (典型名:config.xml) にて

```

<restart filename="restart.xml" interval="1" append_mode="on" atom_id_is_added="on"/>

```

のように、restart タグに「atom\_id\_is\_added="on"」という属性を付加する必要がある。上記で「off」にすると、属性を書かないことと等価となる。

## 9.2 group ID の活用例：group ごとの重心座標の計算

group ID の活用例として、group ごとの重心座標を計算するルーチン set\_weight\_center\_sub がある (ただし非周期系のみに対応したルーチン)。以下に、本質的部分 (エラーチェックや出力の一部を除いた部分) を付す。

```

subroutine set_weight_center
  use M_config,          only : config          !(unchanged)
!
  implicit none
  integer :: gid, j, jsv
  integer :: n_atom
  integer :: lu
  integer :: ierr
  real(DOUBLE_PRECISION) :: mass_sum, mass
!
  lu=config%calc%distributed%log_unit
  n_atom = config%system%structure%natom
!
  allocate (group_center(3,num_groups),stat=ierr)
  group_center(:,:)=0.0d0
!
  do gid = 1, num_groups
    call set_weight_center_sub(gid,group_center(1:3,gid))
  enddo
!
  do gid = 1, num_groups

```

```

        if (lu > 0) write(lu,'(a,i10,3f10.5)')'gid, group center =', gid, group_center(1:3,gid)
    enddo
!
    end subroutine set_weight_center
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! @ Set the weight center for 'gid'-th group : COMPATIBLE ONLY TO THE NON-PERIODIC CASES
subroutine set_weight_center_sub(gid, group_center_wrk)
    use M_config,          only : config          !(unchanged)
    use M_qm_domain,       only : atm_position, atm_element !(unchanged)
    use elses_mod_mass,    only : awt             !(unchanged)
!
    implicit none
    integer,                intent(in)  :: gid
    real(DOUBLE_PRECISION), intent(out) :: group_center_wrk(3)
    integer :: j, jsv
    integer :: n_atom
    integer :: ierr
    real(DOUBLE_PRECISION) :: mass, mass_sum
!
    n_atom = config%system%structure%natom
!
    mass_sum=0.0d0
    group_center_wrk(1:3)=0.0d0
    do j = 1, num_group_mem(gid)
        jsv=group_mem(j, gid)
        mass=awt(atm_element(jsv))
        mass_sum=mass_sum+mass
        if ( (jsv < 1) .or. (jsv > n_atom) ) then
            write(*,*)'ERROR(ini_group_id):jsv=',jsv
            stop
        endif
        group_center_wrk(1:3)=group_center_wrk(1:3)+mass*atm_position(1:3,jsv)
    enddo
    group_center_wrk(1:3)=group_center_wrk(1:3)/mass_sum
!
end subroutine set_weight_center_sub

```

## 10 極限的な大規模計算

ここでは極限的な（おおむね 100 万原子以上の）大規模計算をやるときの、細かい設定について説明する。

### 10.1 Microcell booking の手動設定

注：この節の説明は、あまり分かりやすすくないので、必要に迫られない限りは、読む必要はない。

DST ワークフローにおいては、各原子の回りに球状の「projection 領域」をとる。例えば、projection=100 とすると、球の半径は、球内に最低 100 原子が入る必要がある。ただし、どれだけの半径をとれば上記が満たされるのかは、分からない。これをどうやって決めるか、というのが、以下の話。

$O(N^2)$  の計算を厭わなければ、全部の原子 pair について距離を計算すれば良い。これを実行するのが「one cell setting」で、config.xml に

```
<micro_cell_booking mode="one_cell">
```

と書けば良い。たとえば、

```
sample/sample_geno_mpi/C_liq_01728atom_dst/config.xml
```

がその例。経験的には、10 万原子くらいまでは、上記で問題ない（全計算時間に占める割合が大きい）。

しかし、100 万原子以上だと、上記の  $O(N^2)$  計算の計算時間が大きくなる場合がある。そこで、以下の「micro cell booking」を行う。まず、periodic cell を直方体状の「micro cell」に分割する。次に、各原子に対して、自分が所属している micro cell の近傍 cell のみの対して、booking を行う。たとえば、distributed タグ内に、下記の要領で書く。

```
<micro_cell_booking mode="default">
  <cell_number_x> 4 </cell_number_x>
  <cell_number_y> 4 </cell_number_y>
  <cell_number_z> 4 </cell_number_z>
  <search_range_x> 1 </search_range_x>
  <search_range_y> 1 </search_range_y>
  <search_range_z> 1 </search_range_z>
</micro_cell_booking>
```

上記の場合は、 $x, y, z$  方向にそれぞれ 4 分割して、micro cell を定義している。micro cell の総数は  $4 \times 4 \times 4 = 64$  となる。search\_range は、どこまでの micro cell を検索対象にするのか、を指定している。上記ではすべて 1 であり、自分の cell に対して、1 つの方向に第一隣接 cell までとっている (-1, 0, +1)。3 方向にそれぞれとるので、 $3^3 = 27$  個の cell をとったことになる。

上記のような手動設定をしたくない場合は、

```
<micro_cell_booking mode="default"></micro_cell_booking>
```

とすれば、自動設定してくれるが、万能ではない。自動設定時には「1 方向の cell 数は 2 のべき乗」になるようにとり、「系全体で密度が均一であること」を仮定して見積もる。疎な領域、密な領域、ができると、破綻することがある。

自動設定において、上記 parameter がいくらに設定されているのかは、標準出力にでる。

```
@@ initial_mcell_booking:myrank,nprocs= 0 1296
noav= 101376000
INFO-CELL:n_cell= 1048576
```

```

INFO-CELL:n_cell_x, ax/(n_cell_x)[au,A]=      64      37.7684357030      19.9861875000
INFO-CELL:n_cell_y, ay/(n_cell_y)[au,A]=      128      20.7719250837      9.9930937500
INFO-CELL:n_cell_z, az/(n_cell_z)[au,A]=      128      30.8261135924      9.9930937500
INFO-CELL:  dn_cell_x =          2
INFO-CELL:  dn_cell_y =          2
INFO-CELL:  dn_cell_z =          2
Check : sum of atoms in cell (should be noav)...OK!
max. number of the atoms in a cell, allocated size = 258 387

```

この場合は、手動で

```

<cell_number_x> 64 </cell_number_x>
<cell_number_y> 128 </cell_number_y>
<cell_number_z> 128 </cell_number_z>
<search_range_x> 2 </search_range_x>
<search_range_y> 2 </search_range_y>
<search_range_z> 2 </search_range_z>

```

と書いたことと等価。総 cell 数は、(総 cell 数) =  $64 \times 128 \times 128 = 1048576$  となる。booking するには、1 方向に second neighbor まで探索している (-2, -1, 0, +1, +2)。つまり、booking の際には、 $5^3 = 125$  個の cell に含まれる原子に対して、search をかける。もし、すべての micro cell に均一に原子が入っているなら、1cell あたり、(全原子数) / (総 cell 数) = 96.6796875 と、約 100 原子が入っていることになる。125 個の cell だと約  $100 \times 125 =$  約 12500 個となるので、「100 原子を含む領域をとる」ためには、十分な「気がする」。

しかし上記はあくまでも「micro cell に均一に原子が入っている」仮定での見積もりであり、疎密があると、うまくいかない場合がある。その場合は、以下のようなメッセージが標準出力にでて止まる：

```

@@ ALLOC_VAL_ELEC:NOS_DEF= 2
ax = 2417.179884991220
ay = 2658.806410709460
az = 3945.742539830720
ntlimit      = -10
r_mem_limit = -1.0000000000000000
ERROR:rcut_book_tmp, max_distance= 41.90947341092205 41.72493324539812
      atm_index= 91375191
length_of_atm_list      = 100
length_of_atm_list_wrk= 3368
      itry= 47
rcut_book_wrk= 12.69984042755214
      n_count= 114

```

この場合は探索する範囲を増やせば（計算時間は増えるかもしれないが）解決する。たとえば、

```

<search_range_x> 3 </search_range_x>
<search_range_y> 3 </search_range_y>
<search_range_z> 3 </search_range_z>

```

とする、など。



## 11 External Library の利用

External Library(外部ライブラリ)を利用するの計算について。「外部ライブラリ」は ELSSES そのものではなく、ELSES に内包される予定はない。汎用な (他ソフトウェアにも接続されることを想定した) 数値計算ライブラリである。

### 11.1 EigenKernel の利用

EigenKernel は、分散並列の対角化ソルバーを提供する。ScaLAPACK, ELPA, EigenExa が使えるが、ここでは ScaLAPACK 利用のみ説明する。

京コンピュータでの利用を想定して、作業手順を記す。

Step (0) : EigenKernel のダウンロード …… 下記 URL から、elses\_ext-lib\_20141214.zip をダウンロードする :

<https://drive.google.com/file/d/0B-65ZAw4AZq0ZnZTWj1Pc3JVS2c/view?usp=sharing>

Step (1) : ELSSES の top directory (ELSES\_history.txt がある directory) に、上記 zip ファイルを置く。さらにこのディレクトリで

```
$ unzip elses_ext-lib_20141214.zip
```

を実行する。

```
ext-lib/
```

directory 以下のファイルが置き換えられる。

Step (2) : 下記を実行する。

```
$ cp -p src/code_for_mpi/elses-lib-mpi-wrapper.f90 src/elses-lib-mpi-wrapper-compile.f90
```

```
$ cp -p src/code_optional/elses-la-eigen-solver-main-true.f90 src/elses-la-eigen-solver-main.f90
```

Step (3) : Makefile.inc を京用にカスタマイズする。下記設定になるようにする。

```
FC = mpifrtpx
FFLAGS = -Kfast,openmp
LDFFLAGS = $(FFLAGS)
LIBS-EXT = -L$(TOPDIR)/ext-lib -lEigenTest
LIBS = -SCALAPACK -SSL2BLAMP
```

Step (4) : ELSSES を make する。

```
$ make
```

Step (5) : EigenKernel 対応 sample data がある

```
sample/sample_w_ext_lib/Polymer_PPE_144atom_dyn_w_EigenKernel
```

に移動し、ELSES を実行する。

設定 XML ファイル (config.xml) の内容は、以下である。

```

<?xml version="1.0" encoding="UTF-8"?>
<config name="Polymer_PPE_144atom_eigen_mpi_20160102">
<system>
  <cluster structure="Polymer_PPE_144atom.xml" />
  <boundary x="nonperiodic" y="nonperiodic" z="nonperiodic" />
  <element name="C" model="geno" filename="C.xml"> </element>
  <element name="H" model="geno" filename="H.xml"> </element>
  <temperature unit="kelvin"> 500 </temperature>
  <heatbath_mass mode="on">
    <mass_per_atom unit="a.u."> 25.0d0 </mass_per_atom>
  </heatbath_mass>
</system>
<calc mode="dynamics">
  <solver scheme="eigen_mpi">
    <eigen_mpi>
      <SEP_solver> scalapack </SEP_solver>
      <GS_transformation> scalapack </GS_transformation>
      <blocksize> 32 </blocksize>
    </eigen_mpi>
  </solver>
  <genoOption>
    <CSC_method> ELSTNER </CSC_method>
    <CSC_max_loop_count> 0 </CSC_max_loop_count>
    <CSC_charge_convergence> 1d-5 </CSC_charge_convergence>
    <CSC_charge_mixing_ratio> 0.1 </CSC_charge_mixing_ratio>
  </genoOption>
  <dynamics scheme="velocity verlet">
    <delta unit="fsec"> 1.00 </delta>
    <total unit="fsec"> 100.00 </total>
  </dynamics>
  <use_integer_elec_num mode="on" />
  <wave_packet mode="off" />
</calc>
<output>
  <restart filename="Polymer_PPE_144atom_restart.xml" interval="10" />
  <position filename="Polymer_PPE_144atom_position.xyz" interval="10" />
  <wavefunction filename="output_wavefunction.txt" interval="10" />
  <eigen_level filename="output_eigen_levels.txt" mode="periodic" interval="10" format="full" unit="eV" />
</output>
</config>

```

ここで、

```

<solver scheme="eigen_mpi">
  <eigen_mpi>

```

```

    <SEP_solver> scalapack </SEP_solver>
    <GS_transformation> scalapack </GS_transformation>
    <blocksize>      32 </blocksize>
  </eigen_mpi>
</solver>

```

の部分、ExternalLibrary 利用を指定している。

なお、上記ディレクトリには、job script ファイルの例として jobscript\_on\_KEI.sh が置いてある。内容は、下記であるので、必要に応じて書き直す：

```

#!/bin/bash
#PJM --rsc-list "rscgrp=small"
#PJM --rsc-list "node=2"
#PJM --rsc-list "elapsed=00:10:00"
#PJM --stg-transfiles all
#PJM --stgin "./elses /*.xml ./"
#PJM --stgout "*.xyz *.txt restart*.xml ./ "
#PJM --stgout "std-file.* ./out/ "
#PJM -s
#PJM --mail-list hoshi@damp.tottori-u.ac.jp
#PJM -m b
#PJM -m e
#
. /work/system/Env_base
export PARALLEL=8
export OMP_NUM_THREADS=$PARALLEL
mpiexec -n 2 -of-proc std-file --mca mpi_print_stats 1 lpgparm -s 32MB -d 32MB -h 32MB -t 32MB -p 32MB ./

```

result ディレクトリ以下には、京上の 2 ノード計算の結果ファイル

```

Output.txt
Polymer_PPE_144atom_position.xyz
output_eigen_levels.txt

```

が置いてある。このうち output\_eigen\_levels.txt には、固有エネルギー・占有数・Participation Ratio が記述されている (Sec. 3.2 参照)。

## 12 Sample data による case study

### 12.1 Kwon モデルによる Si 結晶：diamond 構造と simple cubic 構造

Kwon モデル [5] による Si 結晶において、diamond 構造と simple cubic 構造で体積を変えた計算をする。図 1(a) に ELSES での計算結果をまとめ、(b) には対応する原論文の図を載せる。

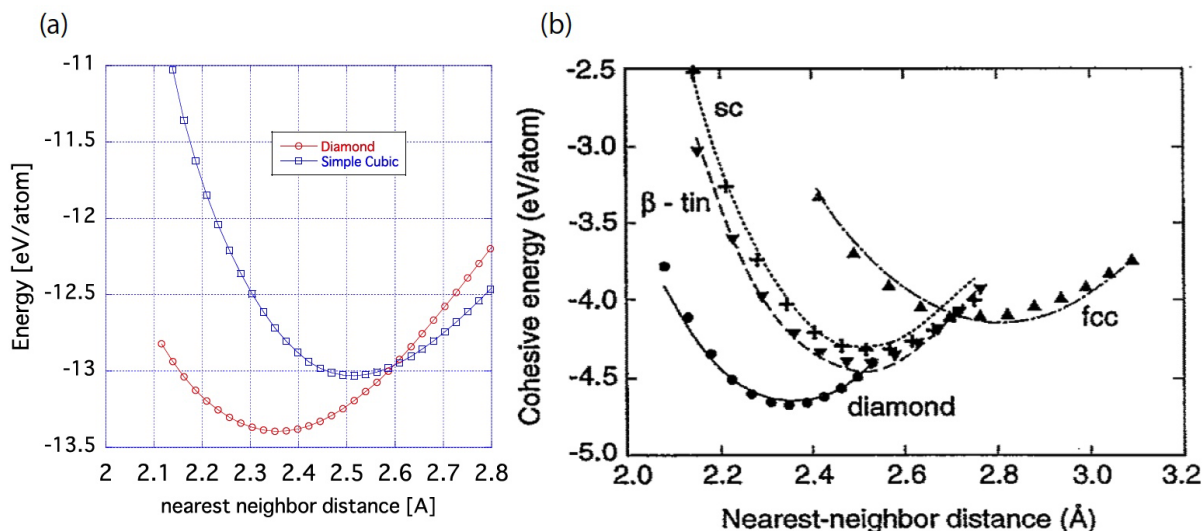


図 1: Kwon モデルによる Si 結晶：diamond 構造と simple cubic 構造; (a) ELSES での計算。(b) 原論文 [5] の図 3。原論文 [5] の caption : Cohesive energy as function of nearest-neighbor distance fore several bulk phases of Si. The points are the LDA calculation and lines are from the present tight binding.

以下、図 1(a) を描く手順を説明する。

#### 12.1.1 diamond 構造計算

パッケージに含まれている、下記 sample data を用いる。

```
sample/sample_non_geno/Si_bulk_512atom_cell_change_only
```

この計算では、512 原子のスーパーセルに対して、cell\_change\_only モード (Quick Start Sec.3.6.2) を用いている。構造 XML ファイルにおける cell length  $L_0$  は  $L_0 \equiv 41.0342853147435$  au であり、nearest neighbor distance  $d_0 = (\sqrt{3}/16)L_0 = 4.442091688588325$  au ( $= 2.350652753492104$  Å) である。cell(格子定数) 変化を設定する補助ファイル input\_cell\_change.txt が用意されており、下記内容となっている。

```
41.0342853147435d0 41.0342853147435d0 41.0342853147435d0
0 0.90 0.90 0.90
1 0.91 0.91 0.91
2 0.92 0.92 0.92
3 0.93 0.93 0.93
4 0.94 0.94 0.94
5 0.95 0.95 0.95
6 0.96 0.96 0.96
7 0.97 0.97 0.97
```

```

8  0.98 0.98 0.98
9  0.99 0.99 0.99
10 1.00 1.00 1.00
11 1.01 1.01 1.01
12 1.02 1.02 1.02
(中略)
28 1.18 1.18 1.18
29 1.19 1.19 1.19
30 1.20 1.20 1.20

```

nearest neighbor distance  $d$  が、 $d:=0.9 d_0$  から  $1.2 d_0$  まで、 $0.01d_0$  刻みで変化する。例えば (step count) = 10 のときが  $d = 1.0d_0$  に相当する。

上記ディレクトリで、下記を実行する。

```
$ ../../bin/elses -verbose ./config.xml > logfile.txt
```

なお、result ディレクトリには一部結果ファイル (Output.txt など) があらかじめ用意されており、計算結果の確認に使える。

各 step count に対して、原子あたりのエネルギーは下記のように与えられる (下記では紙面幅の都合で各行の右側データが省略されている)。

```

$ grep summary Output.txt | grep Energy | grep atom
Energy summary (eV/atom):      0      -12.82093158      -12.82093158      -24.50065459
Energy summary (eV/atom):      1      -12.93880204      -12.93880204      -24.03977513
Energy summary (eV/atom):      2      -13.03834353      -13.03834353      -23.59109129
Energy summary (eV/atom):      3      -13.12538651      -13.12538651      -23.15612302
Energy summary (eV/atom):      4      -13.19560182      -13.19560182      -22.72739995
Energy summary (eV/atom):      5      -13.25443639      -13.25443639      -22.30812835
Energy summary (eV/atom):      6      -13.30259145      -13.30259145      -21.89745887
Energy summary (eV/atom):      7      -13.34043514      -13.34043514      -21.49471533
Energy summary (eV/atom):      8      -13.36814569      -13.36814569      -21.09937337
Energy summary (eV/atom):      9      -13.38582682      -13.38582682      -20.71103784
Energy summary (eV/atom):     10      -13.39358919      -13.39358919      -20.32942075
Energy summary (eV/atom):     11      -13.39159837      -13.39159837      -19.95432136
Energy summary (eV/atom):     12      -13.38009633      -13.38009633      -19.58560932
Energy summary (eV/atom):     13      -13.35940614      -13.35940614      -19.22321110
Energy summary (eV/atom):     14      -13.32992737      -13.32992737      -18.86709912
Energy summary (eV/atom):     15      -13.29212587      -13.29212587      -18.51728154
(略)

```

下記行に現れる自然数は step count(=0,1,2..) であり、最初の実数がエネルギー  $E(\text{eV/atom})$  である。各 step count に対して input\_cell\_change.txt から格子定数の比率を読み取ることで、nearest neighbor distance  $d$  が分かる。結果の  $E(d)$  関数が、図 1 に現れている。

### 12.1.2 simple cubic 構造計算

パッケージに含まれている、下記 sample data を用いる。

```
sample/sample_non_gen0/Si_sc_512atom_cell_change_only
```

原子構造は、diamond 構造と simple cubic 構造とで、nearest neighbor distance の基準値  $d_0$  が共通になるように設定されている。Sec. 12.1.1 と同様に計算を行った場合の結果の  $E(d)$  関数が、図 1 に現れている。

## 12.2 NRL モデルによる W(タングステン) 結晶 : BCC, FCC, SC 構造

(本項目は追記予定) NRL モデルによる W(タングステン) 結晶において、NRL BCC, FCC 構造と simple cubic(SC) 構造で体積を変えた計算をする。

### 12.2.1 BCC 構造計算

パッケージに含まれている、下記 sample data を用いる。

```
sample/sample_non_geno/W_NRL_BCC_250atom_cell_change_only
```

この計算では、250 原子のスーパーセルに対して、cell\_change\_only モード (Quick Start Sec.3.6.2) を用いている。以下の作業を行うのに必要なファイルは、下記だけ。テストを行う際には、下記ファイル以外は別ディレクトリに移動してから行うことを推奨する。

```
W_BCC_250atom.xyz
config.xml
generate.xml
input_cell_change.txt
w_par
```

(1) Element XML file を生成する … コマンドラインから

```
$ ../../../../bin/elses-xml-generate-nrl-param w_par W_nrl.xml W
```

を実行すると、下記ファイルが生成する

```
W_nrl.xml
```

(2) 構造 XML file を生成する … コマンドラインから

```
$ ../../../../bin/elses-xml-generate generate.xml W_BCC_250atom.xml
```

を実行すると、標準出力に

```
INFO:No <unitcell> tag in the XML file
INFO:unitcell info is obtained from the XYZ file: AX [au] =      29.6687119811
INFO:unitcell info is obtained from the XYZ file: AY [au] =      29.6687119811
INFO:unitcell info is obtained from the XYZ file: AZ [au] =      29.6687119811
```

が表示され、下記ファイルが生成する

```
W_BCC_250atom.xml
W_BCC_250atom_basic.xml
```

(3) ELSSES を cell change only モードで計算する … コマンドラインから

```
$ ../../../../bin/elses -verbose config.xml > logfile.txt &
```

を実行すると、各種出力ファイルができる。結果が正しいかどうかは、

```
$ grep 'Energy summary (eV/atom)' Output.txt
```

として出てくる結果と、

```
$ grep 'Energy summary (eV/atom)' result/Output.txt
```

が同一であることで確認できる。

### 12.2.2 FCC 構造計算

パッケージに含まれている、下記 sample data を用いる。

```
sample/sample_non_geno/W_NRL_FCC_500atom_cell_change_only
```

手順は、BCC 構造データと同じ。

### 12.2.3 SC 構造計算

パッケージに含まれている、下記 sample data を用いる。

```
sample/sample_non_geno/W_NRL_SC_125atom_cell_change_only
```

手順は、BCC 構造データと同じ。



## 12.3 共役高分子系:poly-(9,9 dioctyl fluorene) (ELSEES sample extra に収録)

ELSEES sample extra [6] に含まれる、共役高分子関連物質

### 12.3.1 モノマー

ELSEES sample extra [6] の下記ディレクトリに含まれる：

```
PFO_n1_opt/
```

以下のファイルが格納されている：

```
PFO_n1.xyz  
config.xml  
elses_test.sh  
generate.xml  
input_mesh_grid.txt  
result/Output.txt  
result/eigen_state_000079.cube  
result/eigen_state_000080.cube  
result/log-node000000.txt  
result/logfile.txt
```

結果ファイルは、result/ディレクトリに格納されているので、検証に利用できる。

簡単に実行するには、上記ディレクトリにある shell script

```
elses_test.sh
```

を実行すれば良い。ファイルの内容は、

```
#!/bin/sh  
echo 'TEST: elses-xml-generate generate.xml PFO_n1.xml'  
elses-xml-generate generate.xml PFO_n1.xml  
echo 'TEST: elses -verbose ./config.xml > logfile.txt'  
elses -verbose ./config.xml > logfile.txt  
echo 'TEST: elses-generate-cubefile 79 80'  
elses-generate-cubefile 79 80
```

であり、3つの計算を行っている：(1) XYZ ファイルから構造 XML ファイルを生成、(2) ELSEES による構造最適化、(3) HOMO, LUMO 波動関数の cube file を生成。

### 12.3.2 ダイマー

ELSES sample extra [6] の下記ディレクトリに含まれる：

```
PFO_n2_opt/
```

以下のファイルが格納されている：

```
PFO_n2.xyz  
config.xml  
elses_test.sh  
generate.xml  
input_mesh_grid.txt  
result/Output.txt  
result/eigen_state_000157.cube  
result/eigen_state_000158.cube  
result/log-node000000.txt  
result/logfile.txt
```

結果ファイルは、result/ディレクトリに格納されているので、検証に利用できる。

簡単に実行するには、上記ディレクトリにある shell script

```
elses_test.sh
```

を実行すれば良い。ファイルの内容は、

```
#!/bin/sh  
echo 'TEST: elses-xml-generate generate.xml PFO_n2.xml'  
elses-xml-generate generate.xml PFO_n2.xml  
echo 'TEST: elses -verbose ./config.xml > logfile.txt'  
elses -verbose ./config.xml > logfile.txt  
echo 'TEST: elses-generate-cubefile 157 158'  
elses-generate-cubefile 157 158
```

であり、3つの計算を行っている：(1) XYZ ファイルから構造 XML ファイルを生成、(2) ELSES による構造最適化、(3) HOMO, LUMO 波動関数の cube file を生成。

### 12.3.3 10 量体 (DST ワークフロー)

ELSES sample extra [6] の下記ディレクトリに含まれる：

```
PFO_n10_dyn_dst/
```

以下のファイルが格納されている：

```
PFO_n10.xyz  
config.xml  
generate.xml  
result/Output.txt  
result/PFO_n10.xml  
result/PFO_n10_basic.xml  
result/log-node000000.txt  
result/log-node000001.txt  
result/log.txt  
result/output_C.xml  
result/output_H.xml  
result/position.xyz  
result/restart.xml
```

結果ファイルは、result/ディレクトリに格納されているので、検証に利用できる。

簡単に実行する手順を記す。ここでは標準的ワークステーションで、(MPI 並列度数)=2、の場合の説明をする。

STEP1：構造 XML ファイルを作成する

```
elses-xml-generate generate.xml PFO_n10.xml
```

STEP2：ELSES を実行する

```
mpirun -n 2 ./elses -verbose ./config.xml > log.txt
```

## 12.4 大規模アモルファス状共役高分子系の DST ワークフロー

### 12.4.1 約 13 万原子系 (FL-test-07-132864atom.xyz)

STEP1 : ウェブページ

<http://www.damp.tottori-u.ac.jp/~hoshi/elses-data/ELSES-FL-DST-test-201602/>

から以下のファイルをダウンロードする。

```
FL-test-07-132864atom.xyz.gz
config-FL-test-07-132864atom-20160204.xml
generate-FL-test-07-132864atom-20160204.xml
```

さらにファイル展開と名義変更を行う

```
$ gunzip FL-test-07-132864atom.xyz.gz
$ mv config-FL-test-07-132864atom-20160204.xml config.xml
$ mv generate-FL-test-07-132864atom-20160204.xml generate.xml
```

STEP2 : 構造 XML ファイルを作成する :

```
$ elses-xml-generate generate.xml FL-test-07-132864atom-20160204.xml
```

STEP3 : ELSES を実行する :

```
$ elses -verbose config.xml
```

参考 : generate-FL-test-07-132864atom-20160204.xml の内容

```
<?xml version="1.0" encoding="UTF-8"?>
<generate name="FL-test-07-132864atom-20160204">
  <unitcell>
    <vector unit="angstrom"> 107.4256d0    0.0        0.0    </vector>
    <vector unit="angstrom">  0.0        107.4256d0    0.0    </vector>
    <vector unit="angstrom">  0.0          0.0        107.4256d0 </vector>
  </unitcell>
  <cluster structure="FL-test-07-132864atom.xyz" >
  </cluster>
  <boundary mode="on" x="periodic" y="periodic" z="periodic" />
  <atom_id_is_added mode="on" />
  <heatbath>
    <massperatom unit="a.u."> 25.0d0</massperatom>
  </heatbath>
</generate>
```

参考 : config-FL-test-07-132864atom-20160204.xml の内容

```
<?xml version="1.0" encoding="UTF-8"?>
<config name="FL-test-07-132864atom">
<system>
```

```

<cluster structure="FL-test-07-132864atom.xml" parser="sax" cutoff_radius="5.0" />
<temperature unit="kelvin"> 600.0 </temperature>
<element name="C" model="geno" filename="C.xml"> </element>
<element name="H" model="geno" filename="H.xml"> </element>
</system>
<calc mode="optimization">
  <distributed mode="on">
    <global_density_matrix_is_off />
    <global_hamiltonian_matrix_is_off />
    <micro_cell_booking mode="default"></micro_cell_booking>
  </distributed>
  <optimization>
    <sd_ratio> 0.2 </sd_ratio> <max_num_iter> 10 </max_num_iter>
  </optimization>
  <dynamics scheme="velocity verlet">
    <delta unit="fsec"> 3.00 </delta>
    <total unit="fsec"> 15.00 </total>
  </dynamics>
  <genoOption>
    <CSC_method> ELSTNER </CSC_method>
    <CSC_max_loop_count> 0 </CSC_max_loop_count>
    <CSC_charge_convergence> 1d-4 </CSC_charge_convergence>
    <CSC_charge_mixing_ratio> 0.1 </CSC_charge_mixing_ratio>
  </genoOption>
  <solver scheme="gKrylov_A">
    <projection> 100 </projection>
    <dimension> 30 </dimension>
    <mode_for_large_memory> 0 </mode_for_large_memory>
    <mArnoldi_q> 15 </mArnoldi_q>
    <inner_cg_loop>
      <max_iteration> 100 </max_iteration>
      <convergence_eps> -8 </convergence_eps>
    </inner_cg_loop>
    <mode_for_suggest_projection> default </mode_for_suggest_projection>
  </solver>
</calc>
<output>
  <restart filename="restart.xml" interval="0" append_mode="on"/>
  <position filename="position.xyz" interval="0" cell_info="on" />
</output>
</config>

```

## 12.4.2 約 212 万原子系 (FL-test-11-2125824atom.xyz)

ここでは京上の  $P = 64$  ノード計算を想定し、 $s = 64$  分割した構造 XML ファイルを使う。

### STEP1: ウェブページ

<http://www.damp.tottori-u.ac.jp/~hoshi/elses-data/ELSES-FL-DST-test-201602/>  
から以下のファイルをダウンロードする。

```
FL-test-11-2125824atom.xyz.gz
config-FL-test-11-2125824atom-20160210.xml
generate-FL-test-11-2125824atom-20160210.xml
job-script-kei-FL-test-11-2125824atom-20160210.sh
```

さらにファイル展開と名義変更を行う

```
$ gunzip FL-test-11-2125824atom.xyz.gz
$ mv config-FL-test-11-2125824atom-20160210.xml config.xml
$ mv generate-FL-test-11-2125824atom-20160210.xml generate.xml
$ mv job-script-kei-FL-test-11-2125824atom-20160210.sh job.sh
```

STEP2:  $s=64$  分割された構造 XML ファイルを作成する:

```
$ elses-xml-generate generate.xml FL-test-11-2125824atom.xml -split=64
```

実行した際の標準出力の冒頭は、以下になる:

```
The number of the splitted file=          64
elses-xml-generate
INFO:XYZ file name=FL-test-11-2125824atom.xyz
INFO:get_basic_info: filename=FL-test-11-2125824atom.xyz
INFO:<unitcell> tag was found in the XML file and will be used for the unit cell info.
INFO:The value is specified in the XML file : heatbath%massperatom =
    25.00000000000000
INFO: <atom_id_is_added> tag is detected
generate_output:filename=FL-test-11-2125824atom_000000.xml
generate_output :atom_ini, atom_fin, atom_num =          1      33216      33216
generate_output:filename=FL-test-11-2125824atom_000001.xml
generate_output :atom_ini, atom_fin, atom_num =      33217      66432      33216
generate_output:filename=FL-test-11-2125824atom_000002.xml
generate_output :atom_ini, atom_fin, atom_num =      66433      99648      33216
(以下略)
```

全ノードに必要な「basic ファイル」

```
FL-test-11-2125824atom_basic.xml
```

その他、「分割原子データファイル」

```
FL-test-11-2125824atom_000000.xml
FL-test-11-2125824atom_000001.xml
```

```
....  
FL-test-11-2125824atom_000063.xml
```

が生成される。「basic ファイル」は、simulation cell size など、系全体の情報が記された小さいファイルである。一方、通し番号がついたファイルには、原子データ (原子座標など) が分割されて格納されている。例えば、標準出力から分かる通り、FL-test-11-2125824atom\_000000.xml には、原子 index  $I = 1, 2, 3, \dots, 33216$  の原子データが格納されている。注：分割数を変えたいときは、上記の-split=64 の値を変更する。

STEP3: 京上でファイルを用意する。ユーザーディレクトリには、basic ファイル、分割原子データファイル ( $s=64$  個)、設定 XML ファイル (config.xml)、job ファイル (job.sh) が必要となる。

STEP4: staging check をする

```
$ pjstgchk job.sh
```

「0 error(s).」と出れば、オッケー。

STEP5: job を submit する:

```
$ pjsb job.sh
```

注：各ノードでは、実行ファイル (elses) の他、basic ファイル、分割原子データファイルの 1 つ、設定 XML ファイル (config.xml)、が stage-in される。たとえば、0 番目のノードでは、

```
FL-test-11-2125824atom_basic.xml  
FL-test-11-2125824atom_000000.xml  
config.xml
```

である。

STEP6: job 終了後、経過時間などを見る (Sec. 1.2.1 を参照)。本設定では、出力される構造 XML ファイル (restart XML ファイル) も分割されている (Sec. 1.2.1 を参照)。

参考: generate-FL-test-11-2125824atom-20160210.xml の内容。

```
<?xml version="1.0" encoding="UTF-8"?>  
<generate name="FL-test-11-2125824atom-20160210">  
  <unitcell>  
    <vector unit="angstrom"> 429.7024d0    0.0        0.0    </vector>  
    <vector unit="angstrom">   0.0        214.8512d0   0.0    </vector>  
    <vector unit="angstrom">   0.0          0.0        214.8512d0 </vector>  
  </unitcell>  
  <cluster structure="FL-test-11-2125824atom.xyz" >  
  </cluster>  
  <boundary mode="on" x="periodic" y="periodic" z="periodic" />  
  <atom_id_is_added mode="on" />  
  <heatbath>
```

```
<massperatom unit="a.u."> 25.0d0</massperatom>
</heatbath>
</generate>
```

注：分割数を変える際、本ファイルを変更する必要はない。

参考：config-FL-test-11-2125824atom-20160210.xml の内容。

```
<?xml version="1.0" encoding="UTF-8"?>
<config name="FL-test-11-2125824atom-20160210">
<system>
  <cluster structure="FL-test-11-2125824atom.xml"
    parser="sax" cutoff_radius="5.0" read_mode="split" number_of_atoms="2125824" />
  <splitted_input_file number_of_files="64" />
  <temperature unit="kelvin"> 600.0 </temperature>
  <element name="C" model="geno" filename="C.xml"> </element>
  <element name="H" model="geno" filename="H.xml"> </element>
</system>
<calc mode="optimization">
  <distributed mode="on">
    <global_density_matrix_is_off />
    <global_hamiltonian_matrix_is_off />
    <micro_cell_booking mode="default"></micro_cell_booking>
  </distributed>
  <optimization>
    <sd_ratio> 0.2 </sd_ratio> <max_num_iter> 5 </max_num_iter>
  </optimization>
  <dynamics scheme="velocity verlet">
    <delta unit="fsec"> 3.00 </delta>
    <total unit="fsec"> 15.00 </total>
  </dynamics>
  <genoOption>
    <CSC_method> ELSTNER </CSC_method>
    <CSC_max_loop_count> 0 </CSC_max_loop_count>
    <CSC_charge_convergence> 1d-4 </CSC_charge_convergence>
    <CSC_charge_mixing_ratio> 0.1 </CSC_charge_mixing_ratio>
  </genoOption>
  <solver scheme="gKrylov_A">
    <projection> 100 </projection>
    <dimension> 30 </dimension>
    <mode_for_large_memory> 0 </mode_for_large_memory>
    <mArnoldi_q> 15 </mArnoldi_q>
    <inner_cg_loop>
      <max_iteration> 100 </max_iteration>
      <convergence_eps> -8 </convergence_eps>
    </inner_cg_loop>
  </solver>
</calc>
```



```

    <mode_for_suggest_projection> default </mode_for_suggest_projection>
  </solver>
</calc>
<output>
  <restart filename="restart.xml" interval="2" append_mode="on" split_mode="on" />
</output>
</config>

```

注：分割数（＝ノード数）を変えるときは、上記ファイルの「splitted\_input\_file number\_of\_files="64"」の箇所を変更する。注：上記で parser=で始まる行は、元のファイルでは前行の続き。

参考：job-script-kei-FL-test-11-2125824atom-20160210.sh の内容。

```

#!/bin/bash -x
#
#PJM --rsc-list "node=64"
#PJM --rsc-list "elapsed=00:30:00"
#PJM --rsc-list "rscgrp=small"
#PJM --stg-transfiles all
#PJM --mpi "use-rankdir"
#PJM --stgin "rank=* ./elses %r:./"
#PJM --stgin "rank=* ./config.xml %r:./"
#PJM --stgin "rank=* ./FL-test-11-2125824atom_basic.xml %r:./"
#PJM --stgin "rank=* ./FL-test-11-2125824atom_%06r.xml %r:./"
#PJM --stgout "rank=0 ./Output.txt ./out/"
#PJM --stgout "rank=0-63 ./log-node%06r.txt ./out/"
#PJM --stgout "rank=* ./restart_%06r.xml ./out/"
#PJM --stgout "rank=* ./std-file.%r ./out/"
#PJM -s
#
. /work/system/Env_base
export PARALLEL=8
export OMP_NUM_THREADS=$PARALLEL
mpiexec -n 64 -of-proc std-file --mca mpi_print_stats 1
      lpgparm -s 4MB -d 4MB -h 4MB -t 4MB -p 4MB ./elses ./config.xml

```

注：分割数（＝ノード数）を変えるときは、上記ファイルの「node=64」「rank=0-63」「mpiexec -n 64」の箇所を変更する。注：上記で lpgparm で始まる行は、元のファイルでは前行の続き。

### 12.4.3 supercell 作成を含む作業：約 13 万原子系を例に

supercell 作成を含んだ作業。約 13 万原子系を例に説明する。

STEP0：supercell を作る：元となる xyz ファイル FL-test-01-2076atom-w-cell-info.xyz は、2076 原子系である。supercell の拡張は  $(n_x, n_y, n_z) = (4, 4, 4)$  で、 $2,076 \times 4 \times 4 \times 4 = 132,864$  原子系の supercell が生成される。ELSES パッケージ内の tool/experimental/mk\_supercell\_xyz で以下の作業をする。まず compile をする (以下は gfortran の例)

```
$ gfortran mk_supercell_xyz.f90
```

次に、以下のように実行する。

```
$ ./a.out -nx=4 -ny=4 -nz=4 -filename_input=FL-test-01-2076atom-w-cell-info.xyz  
-filename_output=FL-test-07-132864atom-w-cell-info.xyz
```

(注：上記 2 行は、実際には 1 行で実行する)。出力ファイル FL-test-07-132864atom-w-cell-info.xyz の冒頭は、以下になっている；

	132864	107.4256000000	107.4256000000	107.4256000000			
# generated by mk_supercell_xyz.f90(build 20160215) : nx, ny, nz =					4	4	4
H	5.40968805600000024469	-3.4369591999999999206	-5.37305380100000018473				
C	5.68181185500000029975	-2.41769831700000015218	-5.06208091099999979434				
C	4.67832137999999986278	-1.52045449899999995935	-4.67092103600000019270				
H	3.62555972799999981504	-1.83708786800000001271	-4.684928292000000038270				
C	7.02432938399999962087	-2.03108669800000019023	-5.080289148000000030993				
C	7.38285363500000002546	-0.76542660100000003975	-4.68411288899999966873				
C	8.74589733000000002505	-0.09005534299999999603	-4.71527065799999967055				
C	9.776168781000000086249	-0.82877041600000000976	-3.83202868100000015872				

STEP1：分割 XML ファイルを作る：同ディレクトリにある generate-general-20160215.xml を使って、4 分割 XML ファイルを作る：

```
$ ./elses-xml-generate -xyz_file=FL-test-07-132864atom-w-cell-info.xyz  
-setting_file=generate-general-20160215.xml -output_file=FL-test-07-132864atom.xml -split=4
```

(注：上記 2 行は、実際には 1 行で実行する)。

STEP2：ELSES 実行のため、ウェブページ

<http://www.damp.tottori-u.ac.jp/~hoshi/elses-data/ELSES-FL-DST-test-201602/>  
から、ファイル

```
config-FL-test-07-132864atom-cco-20160215.xml
```

をダウンロードし、改名する：

```
$ mv config-FL-test-07-132864atom-cco-20160215.xml config.xml
```

STEP3：ELSES を実行する：

```
$ else -verbose config.xml > log.txt
```

計算時間の確認法は、Sec. 1.2.1 を参照。計算結果の確認としては、以下のようにする：

```
$ grep 'Energy summary (eV/atom)' Output.txt
Energy summary (eV/atom):      0      -41.61246139      -41.61246139      -42.79258908
Energy summary (eV/atom):      1      -41.61410629      -41.61410629      -42.78655933
Energy summary (eV/atom):      2      -41.61509479      -41.61509479      -42.77992572
Energy summary (eV/atom):      3      -41.61583082      -41.61583082      -42.77309126
Energy summary (eV/atom):      4      -41.61737044      -41.61737044      -42.76710714
```

(注:上記では、各行の右端は省略されている)。

参考：generate-general-20160215.xml の内容：

```
<?xml version="1.0" encoding="UTF-8"?>
<generate name="generate-xml-general-20160215">
  <boundary mode="on" x="periodic" y="periodic" z="periodic" />
  <atom_id_is_added mode="on" />
  <heatbath>
    <massperatom unit="a.u." 25.0d0</massperatom>
  </heatbath>
</generate>
```

参考：config-FL-test-07-132864atom-cco-20160215.xml の内容：

```
<?xml version="1.0" encoding="UTF-8"?>
<config name="FL-test-07-132864atom-cco-20160215">
<system>
  <cluster structure="FL-test-07-132864atom.xml" parser="sax"
    cutoff_radius="5.0" read_mode="split" number_of_atoms="132864" />
  <splitted_input_file number_of_files="4"/>
  <temperature unit="kelvin"> 600.0 </temperature>
  <element name="C" model="geno" filename="C.xml"> </element>
  <element name="H" model="geno" filename="H.xml"> </element>
  <use_matom />
</system>
<calc mode="cell_change_only">
  <distributed mode="on">
    <global_density_matrix_is_off />
    <global_hamiltonian_matrix_is_off />
    <micro_cell_booking mode="default"></micro_cell_booking>
  </distributed>
  <optimization>
    <sd_ratio> 0.2 </sd_ratio> <max_num_iter> 5 </max_num_iter>
  </optimization>
  <dynamics scheme="velocity verlet">
```

```

    <delta unit="fsec"> 3.00 </delta>
    <total unit="fsec"> 15.00 </total>
</dynamics>
<cell_change scheme="from_file" filename="input_cell_change_default_20160213.txt">
    <max_num_iter> 5 </max_num_iter>
</cell_change>
<genoOption>
    <CSC_method> ELSTNER </CSC_method>
    <CSC_max_loop_count> 0 </CSC_max_loop_count>
    <CSC_charge_convergence> 1d-4 </CSC_charge_convergence>
    <CSC_charge_mixing_ratio> 0.1 </CSC_charge_mixing_ratio>
</genoOption>
<solver scheme="gKrylov_A">
    <projection> 100 </projection>
    <dimension> 30 </dimension>
    <mode_for_large_memory> 0 </mode_for_large_memory>
    <mArnoldi_q> 15 </mArnoldi_q>
    <inner_cg_loop>
        <max_iteration> 100 </max_iteration>
        <convergence_eps> -8 </convergence_eps>
    </inner_cg_loop>
    <mode_for_suggest_projection> default </mode_for_suggest_projection>
</solver>
</calc>
<output>
    <restart filename="restart.xml" interval="2" append_mode="on" split_mode="on" />
</output>
</config>

```

(注:「cutoff\_radius=」から始まる行は、実際には前行からの継続行)。

参考: 同様の手順で、様々な supercell での計算が可能となる。原子数  $N$  は、 $N = 2076n_xn_yn_z$ 。これまで作成してきたファイルのは、下記:

- $(n_x, n_y, n_z) = (4, 4, 4)$ : 約 13 万原子 ( $N = 132,864$  原子)  
xyz ファイル名: FL-test-07-132864atom-w-cell-info.xyz
- $(n_x, n_y, n_z) = (16, 8, 8)$ : 約 200 万原子 ( $N = 2,125,824$  原子)  
xyz ファイル名: FL-test-11-2125824atom-w-cell-info.xyz
- $(n_x, n_y, n_z) = (20, 16, 16)$ : 約 1000 万原子 ( $N = 10,629,120$  原子)  
xyz ファイル名: FL-test-14-10629120atom-w-cell-info.xyz
- $(n_x, n_y, n_z) = (28, 30, 30)$ : 約 500 万原子 ( $N = 52,315,200$  原子)  
xyz ファイル名: FL-test-16-52315200atom-w-cell-info.xyz
- $(n_x, n_y, n_z) = (38, 36, 36)$ : 約 1 億原子 ( $N = 102,238,848$  原子)  
xyz ファイル名: FL-test-18-102238848atom-w-cell-info.xyz

## 12.5 理想構造ダイヤモンド計算

京コンピュータ上のベンチマークテストを想定して、理想構造ダイヤモンドの supercell について、cell-change-only(CCO) 計算を行う。Supercell の元ファイルは、cell info 付き xyz ファイル

```
tool/experimental/mk_supercell_xyz/diamond_in_cubic_cell.xyz
```

を用いる。8 原子を含む、cubic cell 構造のデータである。Supercell size( $n_x, n_y, n_z$ ) なら、原子数  $N$  は  $N = 8n_x n_y n_z$  となる。

### 12.5.1 4096 原子系, 4 ノード計算

Supercell size ( $n_x, n_y, n_z$ ) = (8, 8, 8) で、総原子数は  $N = 8n_x n_y n_z = 8^4 = 4096$ 。(1 ノードあたりの原子数)=4096/4 = 1024, (1 コアあたりの原子数)=1024/8 = 128。

STEP0: supercell の xyz ファイルを作る: ELSSES パッケージ内のディレクトリ tool/experimental/mk\_supercell\_xyz で、以下を行う。

```
$ gfortran mk_supercell_xyz.f90
$ ./a.out -nx=8 -ny=8 -nz=8 -filename_input=diamond_in_cubic_cell.xyz
                        -filename_output=DIA-04-4096atom-w-cell-info.xyz
```

注: 上記で最終行は、実際には前行の継続行。

STEP1: 分割構造 XML ファイルを作る:

```
./elses-xml-generate -xyz_file=DIA-04-4096atom-w-cell-info.xyz
                    -setting_file=generate-general-20160215.xml -output_file=DIA-04-4096atom.xml -split=4
```

注: 上記で最終行は、実際には前行の継続行。

STEP2: 必要ファイルの準備:

```
http://www.damp.tottori-u.ac.jp/~hoshi/elses-data/ELSES-TEST-KEI-201602/
```

より、下記ファイルをダウンロードする:

```
input_cell_change_default_20160213.txt
config-FL-test-02-4152atom-cco-20160222.xml
```

```
$ ./elses -verbose ./config-DIA-04-4096atom-cco-20160224.xml > log.txt
```

STEP3: 結果のチェック:

```
$ grep 'Energy summary (eV/atom)' Output.txt
Energy summary (eV/atom):          0      -73.38813555
Energy summary (eV/atom):          1      -73.38873927
Energy summary (eV/atom):          2      -73.38920701
Energy summary (eV/atom):          3      -73.38482610
Energy summary (eV/atom):          4      -73.38659617
```

注：上記では、行の右端を省略している。

```
$ grep MD1 log-node000000.txt
elaps-time(befor MDloop)=          0          0.225300          0.000000          0.000000
elaps-time(lap MDloop)=            1          27.915000          0.000000          0.000000
elaps-time(lap MDloop)=            2          27.519300          0.000000          0.000000
elaps-time(lap MDloop)=            3          27.722200          0.000000          0.000000
elaps-time(lap MDloop)=            4          27.802700          0.000000          0.000000
elaps-time(lap MDloop)=            5          27.892500          0.000000          0.000000
```

注：上記では、行の右端を省略している。

### 12.5.2 52 万原子系, 512 ノード計算

Supercell size  $(n_x, n_y, n_z) = (64, 32, 32)$  で、総原子数は  $N = 8n_x n_y n_z = 8 \times 64 \times 32 \times 32 = 524,288$  (約 52 万)。想定するノード構造は  $8 \times 8 \times 8 (= 512)$ 。(1 ノードあたりの原子数)  $= 524288 / 512 = 1024$ , (1 コアあたりの原子数)  $= 1024 / 8 = 128$ 。

STEP0: supercell の xyz ファイルを作る: ELSSES パッケージ内のディレクトリ `tool/experimental/mk_supercell_xyz` で、以下を行う。

```
$ gfortran mk_supercell_xyz.f90
$ ./a.out -nx=8 -ny=8 -nz=8 -filename_input=diamond_in_cubic_cell.xyz
                    -filename_output=DIA-04-4096atom-w-cell-info.xyz
```

注：上記で最終行は、実際には前行の継続行。

STEP1: 分割構造 XML ファイルを作る:

```
$ ./elses-xml-generate -xyz_file=DIA-04-4096atom-w-cell-info.xyz
                    -setting_file=generate-general-20160215.xml -output_file=DIA-04-4096atom.xml -split=4
```

注：上記で最終行は、実際には前行の継続行。

STEP2: 必要ファイルの準備:

```
http://www.damp.tottori-u.ac.jp/~hoshi/elses-data/ELSES-TEST-KEI-201602/
```

より、下記ファイルをダウンロードする:

```
input_cell_change_default_20160213.txt
config-DIA-04-4096atom-cco-20160224.xml
```

STEP3: ELSSES の実行:

```
$ ./elses -verbose ./config-DIA-04-4096atom-cco-20160224.xml > log.txt
```

参考: config-DIA-04-4096atom-cco-20160224.xml の内容:

```

/bin/bash -x
#
#PJM --rsc-list "node=512"
#PJM --rsc-list "elapsed=00:10:00"
#PJM --rsc-list "rscgrp=small"
#PJM --stg-transfiles all
#PJM --mpi "use-rankdir"
#PJM --stgin "rank=* ./elses %r:./"
#PJM --stgin "rank=* ./config.xml %r:./"
#PJM --stgin "rank=* ./input_cell_change_default_20160213.txt %r:./"
#PJM --stgin "rank=* ./in/DIA-06-524288atom_basic.xml %r:./"
#PJM --stgin "rank=* ./in/DIA-06-524288atom_%06r.xml %r:./"
#PJM --stgout "rank=0 ./Output.txt ./out/"
#PJM --stgout "rank=0-511 ./log-node%06r.txt ./out/"
#PJM --stgout "rank=* ./std-file.%r ./out/"
#PJM -s
#
. /work/system/Env_base
export PARALLEL=8
export OMP_NUM_THREADS=$PARALLEL
mpiexec -n 512 -of-proc std-file --mca mpi_print_stats 1 lpgparm
-s 4MB -d 4MB -h 4MB -t 4MB -p 4MB ./elses -log_node_number=512 ./config.xml

```

注：上記での最後2行は、実際には1行。

STEP3：結果のチェック：

```

$ grep 'Energy summary (eV/atom):' Output.txt
Energy summary (eV/atom):      0      -73.39696901
Energy summary (eV/atom):      1      -73.39963789
Energy summary (eV/atom):      2      -73.40147656
Energy summary (eV/atom):      3      -73.40064076
Energy summary (eV/atom):      4      -73.40069250

```

注：上記では、行の右端を省略している。

```

$ grep MD1 log-node000000.txt
elaps-time(befor MDloop)=      0      64.416000      0.225866      3.491013
elaps-time(lap MDloop)=      1      31.670000      0.031947      0.476263
elaps-time(lap MDloop)=      2      31.626000      0.031986      0.506478
elaps-time(lap MDloop)=      3      31.661000      0.031350      0.547604
elaps-time(lap MDloop)=      4      31.543000      0.030841      0.411449
elaps-time(lap MDloop)=      5      31.427000      0.030963      0.349882
$ grep MD1 log-node000001.txt
elaps-time(befor MDloop)=      0      64.396000      0.211260      2.941086

```

elaps-time(lap	MDloop)=	1	31.674000	0.031968	0.250488
elaps-time(lap	MDloop)=	2	31.629000	0.032083	0.324329
elaps-time(lap	MDloop)=	3	31.665000	0.031644	0.349734
elaps-time(lap	MDloop)=	4	31.545000	0.031217	0.275699
elaps-time(lap	MDloop)=	5	31.430000	0.031370	0.181272

### 12.5.3 約 1 億原子系, 2592 ノード計算

フルノード計算までの強スケーリングテストを想定した系。1 ノードあたりが  $\text{supercellsize}(n_x^{(0)}, n_y^{(0)}, n_z^{(0)}) = (8, 5, 4)$  に対応する。(原子数  $N_0 = 8n_x^{(0)}n_y^{(0)}n_z^{(0)} = 8 \times 8 \times 5 \times 4 = 1280$ )。フルノード計算を想定すると、上記の 1280 原子 cell を、フルノード構造  $(n_x^{(\text{node})}, n_y^{(\text{node})}, n_z^{(\text{node})}) = (32, 48, 54)$  に合わせてつくる。結果として、8-atom cubic cell に対しては、supercell 構造は  $(n_x, n_y, n_z) = (n_x^{(0)}n_x^{(\text{node})}, n_y^{(0)}n_y^{(\text{node})}, n_z^{(0)}n_z^{(\text{node})}) = (8 \times 32, 5 \times 48, 4 \times 54) = (256, 240, 216)$  となる。原子数は、 $N = 8 \times 256 \times 240 \times 216 = 106,168,320$ (約 1 億)。

以下、 $54 \times 48 = 2592$  ノード (フルノードの 32 分の 1) で計算を行う。前節にならって作業を行う。以下、要点だけ記す。

STEP0 : supercell の xyz ファイルを作る:

```
$ ./a.out -nx=256 -ny=240 -nz=216 -filename_input=diamond_in_cubic_cell.xyz
      -filename_output=DIA-16-106168320atom-w-cell-info.xyz
```

注 : 上記 2 行は、実際には 1 行。

STEP1 : 分割構造 XML ファイルを作る:

```
$ ./elses-xml-generate -xyz_file=DIA-16-106168320atom-w-cell-info.xyz
      -setting_file=generate-general-20160215.xml -output_file=DIA-16-106168320atom.xml -split=2592
```

注 : 上記で最終行は、実際には前行の継続行。



## 12.6 行列データ生成用 ICNT 計算

### 12.6.1 ICNT11.2 万原子系 (spd 型)

STEP1 : ウェブページ

<http://www.damp.tottori-u.ac.jp/~hoshi/elses-data/ELSES-ICNT-112000atom-MatGene/>  
から以下のファイルをダウンロードする。

```
C_spd_Cerda.xml
config-ICNT-112000atom-MatGene-spd-20160208.xml
ICNT112000atom.xml.gz
```

さらにファイル展開と symbolic link する

```
$ gunzip ICNT112000atom.xml.gz
$ ln -s config-ICNT-112000atom-MatGene-spd-20160208.xml config.xml
```

STEP2 : ELSES を実行する :

```
$ elses -verbose config.xml
```

$N = 112,000$  原子系であり spd 型モデルであるので、基底数  $M$  は  $M = 1,008,000$ 。

注 : 現状 (記事執筆当時:v.0.06.01) では、90GB 程度消費する。

参考 : config-ICNT-112000atom-MatGene-spd-20160208.xml の内容

```
<?xml version="1.0" encoding="UTF-8"?>
<config name="CNT">
<system>
  <cluster structure="ICNT112000atom.xml" parser="sax" cutoff_radius="13.0" />
  <boundary x="periodic" y="periodic" z="periodic" />
  <temperature unit="kelvin"> 500 </temperature>
  <element name="C" model="geno" filename="C_spd_Cerda.xml"> </element>
</system>
<calc mode="matrix_generation">
  <dynamics scheme="velocity verlet">
    <delta unit="fsec"> 1.00 </delta>
    <total unit="fsec"> 100.00 </total>
  </dynamics>
  <solver scheme="krylov"> </solver>
  <use_integer_elec_num mode="on" />
</calc>
<output>
  <restart filename="CNT-restart.xml" />
  <position filename="CNT-position.xyz" interval="4" />
  <matrix_hamiltonian filename="output_hamiltonian_au.txt"
    mode="last" format="MatrixMarket_sym" unit="a.u."/>
  <matrix_overlap filename="output_overlap.txt" mode="last" format="MatrixMarket_sym"/>
</output>
</config>
```

```
</output>  
</config>
```

実際のファイルでは、mode="last" から始まる行は前行の続き。

## 12.7 イリジウムを含む分子：Ir(CO)<sub>3</sub>Cl

データパッケージ：ELSES\_IrCO3Cl\_20160925\_hoshi\_wfn.zip (78MB)：

<https://drive.google.com/file/d/0B-65ZAw4AZqOTW1JREVKU0VPWlE/view?usp=sharing>

### 12.7.1 概要

論文 [12] のフォロー。イリジウムを含む分子 Ir(CO)<sub>3</sub>Cl の構造最適化と、基底ごとのマリケン電荷出力。Ir は工業的に重要な貴金属であり、通常  $(5d)^7(6s)^2$  と書かれる。周期律表では「Ir Pt Au」と、Pt, Au とならんでいる。

図 2 に、結果をまとめた。図 2(a)(b) は、論文 [12] の図。「十字形」の分子であり、中央にあるのが Ir 原子。(c) は ELSES による HOMO 軌道 ( $k = 23$ )。ほとんど pure な  $d(3z^2 - r^2)$  波動関数。マリケン電荷解析によると、Ir の d 軌道の電子数は、 $n_d = 7.16$ 。para-spin 計算なので、 $n_d \rightarrow 10$  で fully occupied となる。各軌道ごとのマリケン電荷は、 $n_{xy} = 1.33, n_{yz} = 1.51, n_{zx} = 1.40, n_{x^2-y^2} = 0.98, n_{3z^2-r^2} = 1.93$ 。占有数が低い  $d(x^2 - y^2)$  軌道に着目し、同軌道が大きく寄与している固有状態を描画したのが、図 2(d), (e), (f)。結合性軌道・反結合性軌道に分裂し、結合軌道のみが occupy されている。つまり、この軌道が結合に寄与していることが分かる。分子が「十字形」になっているのは、この軌道のおかげ。

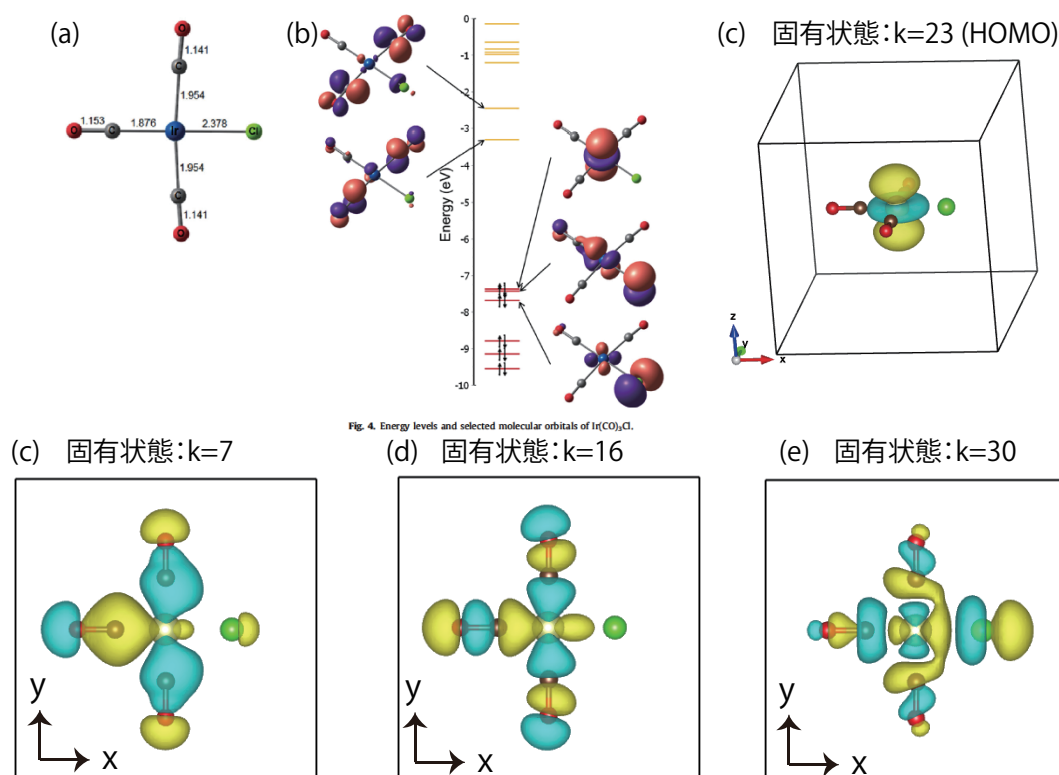


図 2: Ir(CO)<sub>3</sub>Cl 系。(a) 第一原理による最適化構造 (論文 [12] の図)。(b) 第一原理による固有エネルギー・波動関数 (論文 [12] の図)。(c) ELSES 計算による、HOMO 軌道。ELSES データ内で、 $k = 23$  番目の固有状態。ほとんど pure な  $d(3z^2 - r^2)$  波動関数。(d) (e) (f)  $d(x^2 - y^2)$  が大きく寄与している固有状態。 $k=7$ (d),  $k=16$ (e),  $k=30$ (f)。

### 12.7.2 データパッケージの詳細

入出力ファイルの入っているがある。自分で計算してみたい場合は、パッケージ内に OOREADME.txt というファイルが入っているので、その内容に従う。ただし、ELSES は 2016 年 9 月 25 版 (commit:be85533) 以降を使う必要がある。TB

parameter は、論文 [12] にあるものを採用した。

解析の対象となる原子・基底ごとのマリケン電荷は、出力ファイル output\_atom\_charge.txt に、構造最適化後のデータが保存されている。このファイルの全文は、下記である；

8									
mdstep=		129							
1	Ir	9.00000	-0.0525927458	0.72536	1.06344	7.15861	0.42280	0.47107	
2	Cl	7.00000	0.2348485887	1.83465	5.40020	0.00000	1.59566	1.98998	
3	C	4.00000	-0.6489086985	1.21939	2.13170	0.00000	0.71946	0.61512	
4	O	6.00000	0.8353024973	1.62483	5.21047	0.00000	1.71946	1.72542	
5	C	4.00000	-0.7783638975	1.23010	1.99154	0.00000	0.57663	0.77316	
6	O	6.00000	0.5940390766	1.59110	5.00294	0.00000	1.63905	1.73750	
7	C	4.00000	-0.7783638975	1.23010	1.99154	0.00000	0.57663	0.77316	
8	O	6.00000	0.5940390766	1.59110	5.00294	0.00000	1.63905	1.73750	

3 行目からあとが、各原子ごとの情報がでている。ただし、実際の出力は 1 行はもっとながく、上記表記では行の右端が切れている。

「Ir」の文字列が入った行をすべて書くと、以下である：

1	Ir	9.00000	-0.0525927458	0.72536	1.06344	7.15861	0.42280	0.47107	0.16957
		1.32592	1.51456	1.40292	0.98470	1.93051			
8	129	0.00000000	0.00000000						

数字の意味は順番に、(atom index)=1、Ir 原子、(中性原子の価電子数  $n_0$ )=9、(マリケン電荷  $n$  と  $n_0$  の差  $(n - n_0)$ )=-0.053、s 電子のマリケン電荷数=0.725、p 電子のマリケン電荷数=1.06、d 電子のマリケン電荷数=7.16、px 電子のマリケン電荷数=0.42、py 電子のマリケン電荷数=0.47、pz 電子のマリケン電荷数=0.17、d(xy) 電子のマリケン電荷数=1.33、d(yz) 電子のマリケン電荷数=1.51、d(zx) 電子のマリケン電荷数=1.40、d( $x^2 - y^2$ ) 電子のマリケン電荷数=0.98、d( $3z^2 - r^2$ ) 電子のマリケン電荷数=1.93、ハミルトニアン相互作用半径内の原子数 = 8、step count=129 (最後の 2 つの「0」は意味なし)。

参考：00README.txt の全文

```

Ir(CO)3Cl calculation with the TB parameters
in Y. Tsuji et al., Polyhedron 103, pp. 141-180 (2016).
(T. Hoshi, 25. Sep. 2016)
ELSES: ELSES-v0.06.04-20160925-be85533-dev.zip
(0) Initial files
test_IrCO3Cl.xyz
generate.xml
config.xml
Ir_20160803_Hoshi.xml
Cl_20160803_Hoshi.xml
C_20160803_Hoshi.xml
O_20160803_Hoshi.xml
(1) Generate the structure XML file
$ elsas-xml-generate generate.xml test_IrCO3Cl.xml > log-generate-xml.txt
(2) Structure optimization by ELSES
$ elsas -verbose ./config.xml > log.txt &

```

(3) Generate the cube files for visualizaiton of the wavefunctions.

```
$ elses-generate-cube > log-cube.txt
```

参考：config.xml の全文

```
<?xml version="1.0" encoding="UTF-8"?>
<config name="test_IrCO3Cl">
  <system>
    <cluster structure="test_IrCO3Cl.xml" />
    <boundary x="nonperiodic" y="nonperiodic" z="nonperiodic" />
    <element name="Ir" model="geno" filename="Ir_20160803_Hoshi.xml" filecheck="on" />
    <element name="C" model="geno" filename="C_20160803_Hoshi.xml" filecheck="on" />
    <element name="O" model="geno" filename="O_20160803_Hoshi.xml" filecheck="on" />
    <element name="Cl" model="geno" filename="Cl_20160803_Hoshi.xml" filecheck="on" />
  </system>
  <calc mode="optimization">
    <solver scheme="eigen"> </solver>
    <genoOption>
      <CSC_method> ELSTNER </CSC_method>
      <CSC_max_loop_count> 0 </CSC_max_loop_count>
      <CSC_charge_convergence> 1d-6 </CSC_charge_convergence>
      <CSC_charge_mixing_ratio> 0.10 </CSC_charge_mixing_ratio>
    </genoOption>
    <optimization>
      <sd_ratio> 0.2 </sd_ratio>
      <max_num_iter> 2000 </max_num_iter>
      <convergence_criteria mode="energy_per_atom" unit="eV"> 1.0d-5 </convergence_criteria>
    </optimization>
    <use_integer_elec_num mode="on"/>
    <calc_check mode="on">
      <short_atom_pair_distance mode="on">
        <warning_level unit="angstrom"> 0.5 </warning_level>
        <abort_level unit="angstrom"> 0.1 </abort_level>
      </short_atom_pair_distance>
    </calc_check>
  </calc>
  <output>
    <restart filename="restart.xml" interval="1" />
    <position filename="position.xyz" interval="1" />
    <wavefunction filename="output_wavefunction.txt"/>
    <wavefunction_charge filename="output_wfn_charge.txt" />
    <atom_charge filename="output_atom_charge.txt" mode="last" />
    <eigen_level filename="output_eigen_levels.txt" mode="last" format="level_only" unit="eV"/>
  </output>
</config>
```

## 12.8 行列データ生成用 VCNT 計算

### 12.8.1 構造作成

ELSES matrix library にある「VCNT」シリーズの行列データ生成例。熱揺らぎ (thermally vibrated) している single-wall carbon nanotube (VCNT)。DST workflow(MPI 計算) には非対応。

データパッケージは、以下：

```
elses_data_20161003_VCNT_11200atoms_mat_gene.zip  
https://drive.google.com/file/d/0B-65ZAw4AZq0ckYwbUpuZktqRlk/view?usp=sharing
```

パッケージ内のディレクトリ CNT\_dyn で下記作業をする。

(Step 1) Copy the files

```
mk_supercell_xyz.f90  
CNT_w_cell_info.xyz
```

from the directory of the ELSES package: tool/experimental/mk\_supercell\_xyz/.

(Step 2) Compile the super cell maker;For example.

```
$ gfortran mk_supercell_xyz.f90
```

(Step 3) Generate the supercell by the supercell size of (nx,ny,nz)=(1,1,112);

```
$ ./a.out -nx=1 -ny=1 -nz=112 -filename_input=CNT_w_cell_info.xyz [CONTINUED LINE]  
-filename_output=CNT_sucercell_11200atom.xyz -comment=CNT_supercell_11200atom
```

The standard output is as follows;

```
@@@ mk_supecell_xyz.f90 (build 20160215)  
NOTE: This code uses GET_COMMAND_ARGUMENT defined in Fortran2003  
@@@ Read the paramters from the argument  
INFO:supercell sizes   =           1           1           112  
INFO:filename input    = CNT_w_cell_info.xyz  
INFO:filename output   = CNT_sucercell_11200atom.xyz  
INFO:comment line      = CNT_supercell_11200atom  
@@@ Input file : structure_org.xyz  
-->      N              =           100  
-->  ax,ay,az [A] =      10.0000000000      10.0000000000      12.2900000000  
@@@ Output file : structure_new.xyz  
-->      N              =          11200  
-->  ax,ay,az [A] =      10.0000000000      10.0000000000      1376.4800000000  
@@@ Generate the output file  
.... mk_supecell_xyz.f90 (build 20160215) ended without error
```

(Step 4) Check the generated supercell CNT file of CNT\_sucercell\_11200atom.xyz

```
$ head CNT_sucercell_11200atom.xyz  
      11200      10.0000000000      10.0000000000      1376.4800000000
```

CNT_supercell_11200atom			
C	-1.37999999999999989342	3.10000000000000008882	-4.91000000000000014211
C	-0.69999999999999995559	3.31999999999999984013	-3.68000000000000015987
C	-3.37000000000000010658	-0.3499999999999997780	-4.91000000000000014211
C	-3.37000000000000010658	0.3499999999999997780	-3.68000000000000015987
C	-2.9399999999999994671	1.6999999999999995559	-3.68000000000000015987
C	-2.5200000000000001776	2.2700000000000001776	-4.91000000000000014211
C	-0.69999999999999995559	-3.31999999999999984013	-4.91000000000000014211
C	-1.37999999999999989342	-3.10000000000000008882	-3.68000000000000015987

(Step 5) Write the generate XML file

```
$ cat generate.xml
<?xml version="1.0" encoding="UTF-8"?>
<generate name="CNT_supercell_11200atom">
</generate>
```

(Step 6) Generate the structure XML file

```
$ elses-xml-generate -xyz_file=CNT_sucercell_11200atom.xyz [CONTINUED LINE]
-setting_file=generate.xml -output_file=CNT_supercell_11200atom.xml
@@@ get_info_from_command_argument
INFO:filename of the xyz file      =CNT_sucercell_11200atom.xyz
INFO:filename of the setting file =generate.xml
INFO:filename of the output  file =CNT_supercell_11200atom.xml
INFO:filename of the fixed atom list file =
INFO: the single (non-split) file will be generated
@@@ elses-xml-generate
INFO:XYZ file name=CNT_sucercell_11200atom.xyz
INFO:<cluster> not found in the setting XML file
INFO:get_basic_info: filename=CNT_sucercell_11200atom.xyz
INFO:No <unitcell> tag was found in the XML file
INFO:The data in the XYZ file will be used for the unit cell info, if exist.
INFO:No value is specified in the XML file : heatbath%massperatom
INFO:The default value is set              : heatbath%massperatom = 25.000000000000000
INFO:set_fixed_atom ... is ignored
.... elses-xml-generate ends without error
```

(Step 7) Write the config XML file

```
<config name="CNT_supercell_11200atom">
<system>
  <cluster structure="CNT_supercell_11200atom.xml" />
  <boundary x="periodic" y="periodic" z="periodic" />
  <temperature unit="kelvin"> 500 </temperature>
  <element name="C" model="Xu.1992" filename="C.xml"> </element>
</system>
```

```

<calc mode="dynamics">
  <dynamics scheme="velocity verlet">
    <delta unit="fsec"> 1.00 </delta>
    <total unit="fsec"> 100.00 </total>
  </dynamics>
  <solver scheme="krylov">
    <projection> 100 </projection>
    <dimension> 30 </dimension>
  </solver>
  <use_integer_elec_num mode="on" />
</calc>
<output>
  <restart filename="CNT-restart.xml" interval="1" />
  <position filename="CNT-position.xyz" interval="1" cell_info="on" />
</output>
</config>

```

(Step 8) Run ELSES for the MD simulation

```

elses -verbose ./config_md.xml > log.txt &

```

The resultant structure XML file of CNT-restart.xml is generated.

## 12.8.2 行列作成

パッケージ内のディレクトリ CNT\_mat\_gene で下記作業をする。上記作業での CNT-restart.xml を利用する。

(Step 9) Write the config XML file

```

$ cat config_mat_gene.xml
<?xml version="1.0" encoding="UTF-8"?>
<config name="CNT">
  <system>
    <cluster structure="CNT-restart.xml" />
    <boundary x="periodic" y="periodic" z="periodic" />
    <element name="C" model="geno" filename="C_spd_Cerda.xml"> </element>
  </system>
  <calc mode="matrix_generation">
    <solver scheme="krylov"> </solver>
    <use_integer_elec_num mode="on" />
  </calc>
  <output>
    <restart filename="CNT-restart-end.xml" />
    <matrix_hamiltonian filename="output_hamiltonian_au.txt" mode="last" [CONTINUED LINE]
      format="MatrixMarket_sym" unit="a.u." />
    <matrix_overlap filename="output_overlap.txt" mode="last" format="MatrixMarket_sym" />
  </output>
</config>

```



(Step 10) Run ELSEES for the matrix generation procedure

```
elses -verbose ./config_md.xml > log.txt &
```

The Hamiltonian and overlap matrices are generated:

```
output_hamiltonian_au.txt  
output_overlap.txt
```

Note: the cutoff radius is set to be  $r_{\text{cut}} = 5 \text{ \AA}$ , because the minimum cell length is  $L_x = L_y = 10 \text{ \AA}$ .

## 13 雑多な機能

### 13.1 計算途中での終了 (削除項目)

[2014/5/28] QuickStart に記載されたので、この項目は削除。

### 13.2 4倍精度演算テストルーチン

[2015年02月26日追記] Build 20150226以降、4倍精度テストルーチン機能は default では off にしてある (Ref. 更新記録 2015年02月26日)。

(更新記録 2013/05/28, 2013/11/22(2) の内容が基礎) 起動時に、4倍精度演算テストルーチンが走る。4倍精度演算がコンパイラで正しく扱われるかどうかのチェックだけであり、シミュレーション自体には4倍精度演算は用いられていない。

該当ファイルは、elses-lib-quad-precision.f90 であるが、旧コンパイラ (たとえば gfortran v.4.6 未満) ではコンパイルができない。対処法は Sec. 16.5.3 を参照。

詳細：初期設定 (MD loop 前) 中にテストをする。テストには、

$$\pi = 4 \tan^{-1}(1) \quad (19)$$

をつかった。verbose モードの場合、下記内容が log node 冒頭付近に出力される。

```
@@ Quad precision test for pi (= 4.0 * atan(1.0))
Double precision : pi = 3.1415926535897931159979634685441851620000
Quad precision : pi = 3.1415926535897932384626433832795027970000
Reference value : pi = 3.141592653589793238462643383279502884197169399375
(eye guide)          123456789012345678901234567890123456789012345678
INFO:Quad precision test ... OK
```

verbose モードでない場合は、最終行だけ出力される。「Reference value」は文献にある厳密値を、文字列として貼っている。

補足：4倍精度の表現は以下のようにしている：

```
integer, parameter :: QUAD_PRECISION = selected_real_kind(p=30)
real(QUAD_PRECISION) :: value_quad
value_quad = 4.0_QUAD_PRECISION * atan(1.0_QUAD_PRECISION)
```

補足：上記「selected\_real\_kind」の参考 URL

<http://www.nag-j.co.jp/nagfor/faq.htm#faq46>

[http://gcc.gnu.org/onlinedocs/gfortran/SELECTED\\_005fREAL\\_005fKIND.html](http://gcc.gnu.org/onlinedocs/gfortran/SELECTED_005fREAL_005fKIND.html)

### 13.3 出力ファイルの保存ディレクトリを指定

コマンドライン引数 output\_dir を指定することで、出力ファイルの保存ディレクトリを指定する事ができる。例：

```
./elses -output_dir=result01 ./config.xml
```

実行ディレクトリの下に、下記のように出力ファイルが保存される：

```
result01/Output.txt  
result01/log-node000000.txt  
(以下略)
```

注：相対パスで指定する。絶対パスでは、指定しない。

注：当該ディレクトリ (上記例だと result01) があらかじめ存在している必要がある。

## 14 ELSESES XML 仕様のバージョン

ELSESES の XML 仕様のバージョン履歴。XML 仕様は、原則的に下位互換性を持つように拡張されている。が、自由度が高すぎると混乱を招くこともある。そのため互換性のない変更をする場合があり、バージョン番号を変えることにしている。

設定 XML ファイルの、config タグ内の `elses_xml_version` 属性で設定。例：

```
<config name="C6H6_geno_opt_20150121" elses_xml_version="5.01">
```

上記例は、

```
sample/sample_geno/C6H6_opt/config.xml
```

にある。

`elses_xml_version="current"` とすると、最新仕様になる。

注：内部仕様：(ELSESES XML version) は、

```
config%elses_xml_version| (実数)
```

に格納される。「current」のときの値は、subroutine `config_load` の

```
real(8), parameter :: elses_xml_version_current=5.01d0
```

で設定される。

### 14.1 ELSESES XML Version. 1.0

ELSESES v0.05.00 まで。最初の仕様。明示的なバージョン指定がない場合は、これになる。

### 14.2 ELSESES XML Version. 5.01

ELSESES v0.05.01 で導入 (更新記録:2015/01/21)。設定 XML で `boundary tag` を要求する。従来の仕様では、`boundary tag` が無いときは、周期系を仮定していた。しかし、ユーザーが間違える (意図せずに周期系計算になっている) 場合が多いので、明示的な指定を要求することにした。

## 15 ELSESES コードについての雑多な情報

### 15.1 単位換算に用いる物理定数など

ELSESES では、default で原子単位 (atomic unit, au) が用いられる。これは、電子質量  $m_e$ ・プランク定数  $\hbar$ ・電子電荷量の絶対値  $e$  を 1 とした単位系である ( $m_e = \hbar = e = 1$ )。単位換算にもちいられる物理定数などは、`src/elses-lib-phys-const.f90` にまとめられている。以下、よく使われるものをあげておく。ここでは長さ・エネルギー・時間の 1 au を、それぞれ、 $a_B$ (ボーア半径)・Ha(ハートリー)・ $\tau_{au}$ 、と書いている。

- 長さ :  $1 \text{ au} \equiv 1 a_B = 0.529177 \text{ \AA}$
- エネルギー (ハートリーと電子ボルト) :  $1 \text{ au} \equiv 1 \text{ Ha} = 2 \times 13.6058 \text{ eV} (= 27.2116 \text{ eV})$
- 時間 :  $1 \text{ fs} = 124.06948 / 3.0 \tau_{au} (\approx 41.3565 \tau_{au})$
- エネルギー (電子ボルトとケルビン) :  $1 \text{ eV} = 1.60217733 / 1.380658 \times 10^4 \text{ K} (\approx 11604 \text{ K})$

## 16 ユーザーからの報告 (質問) と回答

### 16.1 ELSSES 実行前の作業 (初期構造作成など) について

#### 16.1.1 原子をファイルに記す順番に制限はありますか？

報告:初期構造を XYZ ファイルで作る際に、原子を記す順番に制限はありますか?たとえば、sample/sample\_geno/C6H6\_dyn (ベンゼン) の例では C が 6 つ、H が 6 つの順に並んで記述されています。これをランダムに並べ変えても、問題ないでしょうか。

回答:問題ありません。原子を記す順番には、任意です。

#### 16.1.2 非周期系でも「セル」を設定する必要がありますか？

報告:非周期系でも「セル」を設定する必要がありますか?他に、非周期系計算で注意すべきことはありますか？

回答:「セル」を設定する必要は、あります。これはソフトウェアとしての仕様によるもので、物理的な意味はありません。非周期系計算で注意すべき他点としては、設定 XML ファイル (典型名:config.xml) に

```
<boundary x="nonperiodic" y="nonperiodic" z="nonperiodic"/>
```

と、非周期系であることを明示的に指定する必要があります

#### 16.1.3 原子を置ける座標に制限はありますか？

報告:原子を置ける座標に制限はありますか? 1 次元の例でいうと、 $x = \pm 0.1\text{au}$  に原子を置く場合と、 $x = 0.3\text{au}$ ,  $x = 0.5\text{au}$  に置く場合は、どちらも許されますか？

回答:はい、許されます。ELSSES を実行する上で、座標の絶対値を意識する必要はありません。ただし、cube ファイルを作って波動関数を可視化するには、グリッドを作る領域を座標範囲で指定する必要があるため、その時には座標を意識する必要があります (詳細は Sec. 3.4.3 を参照)。

### 16.2 ELSSES 実行について

#### 16.2.1 実行時エラー:「ERROR(set\_interac\_list\_proj)」

報告: Krylov 法を用いた計算で、下記のような実行時エラーがでて止まる。

```
ERROR(set_interac_list_proj):jj,m_int=          1017          1521
```

注: 2 つの数字 (1017 1521) は、場合による。

回答:手法の詳細設定が正しくない。具体的には、(a) Hamiltonian における interaction radius ・ (b) Krylov 法における real space projection radius は、(a)<(b) でなければならないが、そうっていない。2 つの数字 (1017 1521) は、それぞれ、(b) の領域に入る原子数、(a) の領域に入る原子数、をさし、 $1017 < 1521$  が (b)<(a) を意味している。対応として、(a)(b) をチェックする必要がある。このあたりは完全に自動化されていないところなので、応相談。(b) を大きくするには、設定 XML ファイルの projection タグ値を大きくとる。

### 16.2.2 実行時エラー: 「The structure XML file may be wrong」「The element tag is required」

報告 1: 下記のような実行時エラーがでて止まる (ただし数字部分は場合による)。

```
ERROR*struc_load_sax:after xml_parse_c:nelement, counter_element=      1      2
ERROR:The structure XML file may be wrong.
```

報告 2: 下記のような実行時エラーがでて止まる (ただし数字部分は場合による)。

```
@@ set_structure_data
ERROR:j,jsei(j)=      1      0
#ELSES: Stop by error:
#This may be due to mismatch
# between the configuration XML file
# and the structure XML file
# The element tag is required
# for each atom species
# in the configuration file.
```

回答: 設定 XML ファイルには、各原子種について element tag が必要だが、不足している。例えば、炭素・水素からなる sample/sample\_geno/C6H6\_dyn/config.xml 系では、system タグ下に以下がある

```
<element name="C" model="geno" filename="C.xml"> </element>
<element name="H" model="geno" filename="H.xml"> </element>
```

### 16.2.3 実行時エラー: 「ERROR in uSu」

報告: Krylov 法を用いた計算で、下記のようなエラーが標準出力にでて止まる。

```
ERROR:(calc_u_su_hu_dstm): <u|S|u>= -3.2102553865969337
ERROR in uSu: atm_index =      3423
ERROR in uSu: orb_index =      2
ERROR in uSu: kr_dim =      16
ERROR in uSu: <u|S|u> = -3.2102553866
ERROR in uSu: nna_distance=      3423      3486      7.1895468316
```

注: 上に現れる数字は、場合による。

短い回答: 現象としては、cutoff radius が小さいことが原因なので、cutoff radius を大きくすれば良い。ただしエラーがおきる遠因は、TB parameter が適切でない場合があるので、その可能性も検討した方が良い。

長い回答: 数理的に解説する。一般化固有値問題  $Hy = \varepsilon Sy$  を解くことが数理的基本であるが、重なり行列  $S$  は正定値 (固有値が全て正) である必要がある。 $S$  の各成分は局在基底同士の積分

$$S_{ij} = \int \chi_i(\mathbf{r}) \chi_j(\mathbf{r}) d\mathbf{r} \quad (20)$$

で定義されており、厳密に計算している限りは  $S$  は正定値である。数学的にいうと「各成分が線形独立なベクトルの内積で定義された行列は、正定値である」という定理に基づく。ただし「局在」基底といっても tail を引くので、 $S$  を厳密に定義すると密行列 ( $S_{ij} \neq 0$ ) になる。cutoff を導入すると「ある程度以上離れた原子 pair に対しては、重なり行列を無視する」ことになる。つまり「小さい成分は 0 とみなす」ことを行い、 $S$  を疎行列として扱う。cutoff が小さいと「無理

やり疎行列にしている」ことになり、結果として正定値が崩れることがある。このような場合、計算が（数理的に）破綻する。報告にある、

ERROR in uSu: <u S u> = -3.2102553866
---------------------------------------

というのは、あるベクトル  $u$  に対して、 $u^T S u$  が負 ( $u^T S u \approx -3.21$ ) になっている、ということを意味する。それは  $S$  の正定値性が壊れていることに他ならない。重なり行列は原子構造によって変わるので、有限温度 MD の途中でこのような現象が起こることもある。cutoff をあげることで、正定値性を回復できるので、エラーは避けられる。ただし、ただしエラーがおきる遠因は、TB parameter が適切でない場合があるので、その可能性も検討した方が良い。

## 16.3 波動関数可視化について

### 16.3.1 VESTA(v.3.2.1) で、波動関数表示がおかしいです

報告：VESTA 最新版 (v.3.2.1) で波動関数を表示しようとすると、 $x, y, z$  座標軸を表現する矢印が 2 つ出る、など、表示がおかしいです。(2015 年 1 月 13 日)

回答：これは ELSESES 側の問題ではなくて、VESTA 側の事情と思われます。VESTA v.3.1.8 より cube 形式ファイルの解釈を変えたようです。さしあたって v.3.1.7 に戻すと、問題はなくなるようです。

注：VESTA change log に以下記述があります：「2014 年 1 月 27 日 ver. 3.1.8:Gaussian cube 形式では、構造モデルに周期境界を適用せず、クラスターモデルとして読み込むよう変更」。cube 形式はもともと周期系と非周期系を区別する定義がなく、根源的な問題の 1 つと言えましょう。VESTA のサイトの「Forum」2014 年 4 月 29 日における、VESTA 作者の書込みも参照してください。

## 16.4 ELSESES 実行後の作業について (ただし波動関数可視化関連をのぞく)

項目準備中。

## 16.5 マシン環境に依存した話題

### 16.5.1 「京」でランクディレクトリは使うべきですか？

報告：「京」で rank directory 機能 (Sec. 4.1 参照) は、使うべきですか？ 計算の規模で使い分けるべきですか？ (2015 年 1 月 20 日)

回答：rank directory を使わないと、全ノードで共有されたディスクにデータが乗ります。この場合、非常に大きな入力ファイル（例えば GB オーダー）に対して、非常に多数のノード（例えば 1 万ノード）で同時に読み込みをするような job で、abort したり、hang したり（計算がいつまでたっても終わらない）、といったことが起こりました。しかも、必ず起こるとは限りません。これについては、状況を直接知る術がないらしく、help に聞いてもはっきりした答えが返ってこなくて対処療法という感じです。上記のような規模の計算にならない範囲では、rank directory なしで使っても大丈夫かと思います。が、ELSESES は rank directory 利用にきちんと対応していますので、rank directory を使っても損はないと思います。

### 16.5.2 「京」(対話実行) で「cannot execute binary file」と出て動きません

報告：「京」のログインノード上で対話実行として実行しようとしたところ、「cannot execute binary file」と出て動きません (2015 年 1 月 20 日)。

回答：「京」に限らず多くのスパコンでは、ログインノードと計算ノードは、ハード・OS が異なります。計算ノード用バイナリはログインノードで実行できず、その逆もしかりです。通常行っているコンパイル作業は、「計算ノード用バイ



ナリをログインノード上で生成する」ということで、これを「クロスコンパイル」と言います。対処法はマシンによって異なりますので、取り扱い説明書で「対話 (interactive) 処理」「クロスコンパイラ」などと書かれているところを参照すると良いです。

### 16.5.3 gfortran v4.6 未満で make できません

報告：gfortran v4.6 未満で make できません。下記のようになります (2015 年 1 月 20 日)；

```
gfortran -c -O2 -fopenmp -I../src/Nr1 -I../xmlf90-1.2g-elses/strings -I../xmlf90-1.2g-elses/sax
-I../xmlf90-1.2g-elses/dom -I../src/elses-lib-quad-precision.f90 -o elses-lib-quad-precision.o
elses-lib-quad-precision.f90:25.23:

real(QUAD_PRECISION) :: value_quad
1
Error: Kind -1 not supported for type REAL at (1)
elses-lib-quad-precision.f90:37.37:

value_quad = 4.0_QUAD_PRECISION * atan(1.0_QUAD_PRECISION)
1
Error: Missing kind-parameter at (1)
elses-lib-quad-precision.f90:42.73:

write(log_unit,'(a,f45.40)') ' Quad precision : pi = ',value_quad
1
Error: Symbol 'value_quad' at (1) has no IMPLICIT type
make[1]: *** [elses-lib-quad-precision.o] エラー 1
make[1]: ディレクトリ '/home/staff/work/elses-v0.04.04/src' から出ます
make: *** [elses] エラー 2
```

[2015 年 02 月 26 日追記] Build 20150226 以降、4 倍精度テストルーチン機能は default では off にしてあるため、本現象は起こらない (Ref. 更新記録 2015 年 02 月 26 日)。

回答：gfortran v4.6 未満は 4 倍精度演算に対応しておらず、4 倍精度演算テストルーチン (Sec. 13.2 参照) がコンパイルできません。このルーチンは 4 倍精度演算がコンパイラで正しく扱われるかどうかのチェックだけであり、シミュレーション自体には 4 倍精度演算は用いられていません。したがって、該当ルーチンをダミールーチンに置き換えれば make できます。具体的には、

```
$ cp src/elses-lib-quad-precision.f90 src/elses-lib-quad-precision.f90.org
$ cp src/code_optional/elses-lib-quad-precision-dummy.f90 src/elses-lib-quad-precision.f90
```

を実行してから make してください (上記で 1 行目は original file を保存しているだけで不要です)。

## A 本ノートの更新記録

- 2017 年 5 月 18 日 … Sec. 9 (group ID 機能) に記述追加。
- 2017 年 1 月 19 日 … Sec. 3.6 (原子基底情報 (basis info) の出力) を追加。
- 2016 年 11 月 9 日 … Sec. 16.2.3 (実行時エラー: 「ERROR in uSu」) を追加。
- 2016 年 10 月 3 日 … Sec. 12.8 (行列データ生成用 VCNT 計算) を追加。
- 2016 年 9 月 25 日 … Sec. 12.7 (イリジウムを含む分子: Ir(CO)<sub>3</sub>Cl) を追加。
- 2016 年 8 月 16 日 … Sec. 2.6 (XML ファイル出力の 3 つのモードと継続計算) を刷新。2016 年 8 月 15 日更新を反映。
- 2016 年 8 月 14 日 … Sec. 2.6 (出力 XML ファイルを用いた継続計算) を追加。
- 2016 年 7 月 10 日 … Sec. 14 (ELSE XML 仕様のバージョン) を追加。
- 2016 年 7 月 7 日 … Sec. 9 (Group ID 機能) に、「group ID を restart XML に出力させるには」を追加。
- 2016 年 3 月 22 日 … Sec. 10.1 (Microcell booking の手動設定) 追加。
- 2016 年 3 月 10 日 … Sec. 13.3 (出力ファイルの保存ディレクトリを指定) 追加。Sec. 3.5 (波動関数電荷の出力) 追加。
- 2016 年 2 月 25 日 … Sec. 12.5 (理想構造ダイヤモンド計算) 追加。
- 2016 年 2 月 15 日 … Sec. 12.4.3 (superccell 作成を含む作業: 約 13 万原子系を例に) 追加。
- 2016 年 2 月 10 日 … Sec.12.4.2 (約 212 万原子系, FL-test-11-2125824atom.xyz) 追加。
- 2016 年 2 月 8 日 … Sec.12.4.1 (大規模アモルファス状共役高分子系の DST ワークフロー; 約 13 万原子系) 追加。
- 2016 年 2 月 3 日 … Sec.12.6.1 (行列データ生成用 CNT 計算; ICNT11.2 万原子系 (spd 型)) 追加。
- 2016 年 2 月 3 日 … Sec.16.2.2 (実行時エラー: 「The structure XML file may be wrong」「The element tag is required」) 追加。
- 2016 年 1 月 19 日 … Sec. B.1(MSD と PR) を追加。
- 2016 年 1 月 2 日 … Sec. 11(External Library の利用), Sec. 11.1(EigenKernel の利用) を追加。
- 2016 年 1 月 2 日 … Sec. 3.2(H,S 行列および固有エネルギー書き出しにおける新機能) に ParticipationRatio 出力を追加。
- 2015 年 12 月 17 日 … Sec. 9(Group ID 機能) を追加。
- 2015 年 12 月 16 日 … Sec. 16.2.1 (実行時エラー: 「ERROR(set\_interac\_list\_proj)」) を追加。
- 2015 年 02 月 26 日 … Sec.3.2.1(H,S 行列生成のみのモード (matrix generation モード)) を追加。
- 2015 年 02 月 26 日 … Sec.7.2 (分子凝集系向け group ID 設定ツール) を追加。Sec. 13.2 (4 倍精度演算テストルーチン)・Sec. 16.5.3 (gfortran v4.6 未満で make できません) に追記。
- 2015 年 01 月 20 日 … Sec. 13.2 (4 倍精度演算テストルーチン) を追加。Sec. 16.5.1 (「京」でランクディレクトリは使うべきですか?) を追加。Sec. 16.5.2 (スパコン上 (対話実行) で「cannot execute binary file」と出て動きません) を追加。Sec. 16.5.3 (gfortran v4.6 未満で make できません) を追加。その他、誤字などへの微修正。

- 2015 年 01 月 12 日 … Sec. 16 (ユーザーからの質問と回答) を追加。Sec. 12.3.3 (10 量体 (DST ワークフロー) ) を追加。
- 2015 年 01 月 02 日 … Sec. 3.4.4 (入出力ファイル名のカスタム指定) の追加
- 2014 年 12 月 19 日 … Sec. 7.1(大規模系むけ supercell 構造作成ツール) の微修正。
- 2014 年 9 月 11 日 … Sec. 3.4.3(grid region の指定) を追加。
- 2014 年 9 月 10 日 … Sec. 15.1(単位換算に用いる物理定数など) を追加。
- 2014 年 8 月 28 日 … Sec.3.4(波動関数グリッドデータ出力ツール:elses-generate-cubefile) を追加。
- 2014 年 5 月 28 日 … Sec.10.1(計算途中での終了) は、QuickStart に記載されたので、この項目は実質削除。
- 2014 年 5 月 1 日 … Sec. 12.3 を更新。
- 2014 年 4 月 25 日 … Sec. 12.3 (共役高分子系:poly-(9,9 dioctyl fluorene)) を追加。
- 2014 年 3 月 20 日 …「Sec. 4.1 「京」および FX10 での ELSSES 利用」を更新。
- 2014 年 3 月 18 日 …「Sec. 10.1 計算途中での終了」を追加。
- 2014 年 3 月 3 日 …「Sec. 8 van der Waals 力」に大幅書き換え (default 値が設定されるように)。
- 2014 年 2 月 08 日 …「Sec. 9.2 NRL モデルによる W 結晶」を追加。
- 2014 年 1 月 15 日 …「Sec. 9.1 Kwon モデルによる Si 結晶」を追加。
- 2013 年 12 月 25 日 …「Sec. 8 van der Waals 力」に加筆。
- 2013 年 12 月 20 日 … v. 0.03.20RC2(build 20131220) を対象に。「Sec. 8 van der Waals 力」に加筆。データ整理記号: 20131220-vdW\_C6H6\_dimer
- 2013 年 12 月 19 日 … v. 0.03.20RC(build 20131217) を対象に。「Sec. 8 van der Waals 力」を追加。
- 2013 年 6 月 11 日 … v. 0.03.17RC(build 20130611) を対象に。「5.1 sample データ検証ツール (shell script) (非 DST ワークフロー)」を更新。cube file 生成を追加。
- 2013 年 4 月 25 日 … v. 0.03.16RC(build 20130425) を対象に。「5.3 band 計算データ検証ツール (shell script) 」を新設。「5.1 sample データ検証ツール (shell script) (非 DST ワークフロー)」を更新。band 図描画の検証を追加 (cf. ELSSES 更新記録 2013 年 4 月 25 日)。
- 2013 年 4 月 14 日 … v. 0.03.16RC(build 20130414) を対象に。「5.1 sample データ検証ツール (shell script) (非 DST ワークフロー)」を更新。NRL 系金ナノワイヤを追加 (cf. ELSSES 更新記録 2013 年 4 月 14 日)。
- 2013 年 4 月 13 日 … v. 0.03.16RC(build 20130413) を対象に。
- 2013 年 4 月 2 日 … v. 0.03.16(開発者版: build 20130402) を対象に。「7.1 大規模系むけ supercell 構造作成ツール」。
- 2013 年 3 月 27 日 … v. 0.03.16(開発者版: build 20130327) を対象に。「5.1 sample データ検証ツール (shell script) (非 DST ワークフロー)」 「5.2 sample データ検証ツール (shell script) (DST ワークフロー)」を書き換え。
- 2013 年 3 月 23 日 … v. 0.03.16(開発者版: build 20130322r3) を対象に。sample ディレクトリの整備。「1.2.1 DST ワークフロー用 sample を用いた 実行」 「5.2 sample データ検証ツール (shell script) (DST ワークフロー)」を書き換え。

- 2013 年 3 月 22 日 … v. 0.03.16(開発者版 : build 20130322) を対象に。sample ディレクトリの整備。「1.2.1 DST ワークフロー用 sample を用いた 実行」「5.2 sample データ検証ツール (shell script) (DST ワークフロー)」を書き換え。
- 2013 年 3 月 21 日 … v. 0.03.16(開発者版 : build 20130321r2) を対象に。「1.2.1 DST ワークフロー用 sample を用いた 実行」「5.2 sample データ検証ツール (shell script) (DST ワークフロー)」を書き換え。
- 2013 年 3 月 16 日 rev2 版 … v. 0.03.16(開発者版 : build 20130316r2) を対象に。「5.2 sample データ検証ツール (shell script) (DST ワークフロー)」を書き換え。
- 2013 年 3 月 5 日版 … v. 0.03.16(開発者版 : build 20130305) を対象に。DST ワークフロー用 sample データを新しいディレクトリ「sample/sample\_for\_mpi」以下に移動したことをうけ、「1.2.1 DST ワークフロー用 sample を用いた実行」を書き換え。
- 2013 年 3 月 4 日版 … v. 0.03.16(開発者版 : build 20130304) を対象に。DST ワークフロー用 sample データを新しいディレクトリ「sample\_for\_mpi」以下に移動したことをうけ、「1.2.1 DST ワークフロー用 sample を用いた実行」を書き換え。
- 2013 年 3 月 3 日版 … v. 0.03.16(開発者版 : build 20130303) を対象に。「6.1 原子構造のチェック (距離が近すぎる原子ペアを検出)」に追記。
- 2013 年 2 月 26 日版 … v. 0.03.16(開発者版 : build 20130225rev2) を対象に。「エラー検出機能 : 原子構造のチェック (原子間距離が近すぎる原子ペアを検出)」を追加。細かい修正。
- 2013 年 2 月 25 日版 … v. 0.03.15(正規版) を対象に。細かい修正。
- 2013 年 2 月 19 日版 …「sample データ検証ツール (shell script)」を追加。この機能は、v. 0.03.15(RC) には入っていない (更新記録 2/19 を参照)。次期バージョンに入る予定。
- 2013 年 2 月 4 日版 …「ELSESES メモ : 機能編」から「ELSESES ノート : 機能編」に改題。v. 0.03.15(RC) を対象に。論文リスト更新。
- 2013 年 1 月 17 日版 … v. 0.03.14(正規版) を対象に。Sec. 2.2.4 の細かい修正。
- 2013 年 1 月 14 日版 … ICOHP 出力の説明。
- 2013 年 1 月 7 日版 … 文献更新。細かい修正。
- 2012 年 12 月 13 日版 …「参考:波動関数 (LCAO 係数) ファイルの仕様」「H,S 行列および固有エネルギー書き出しにおける新機能」を追加。細かい修正。
- 2012 年 12 月 09 日版 …「「京」および FX10 での ELSESES 利用」を更新。細かい修正。
- 2012 年 12 月 08 日版 …「ELSESES 本体による XML ファイルの分割・統合」を追加。細かい修正。
- 2012 年 12 月 07 日版 …「波動関数 (LCAO 係数) 書き出しにおける新機能」を追加。細かい修正。
- 2012 年 12 月 06 日版 …「「京」および FX10 での ELSESES 利用」を更新。細かい修正。
- 2012 年 12 月 5 日版 …「並列化ファイル書き出し」を追加。細かい修正。
- 2012 年 12 月 4 日版 …「「京」および FX10 での ELSESES 利用」を追加。細かい修正。
- 2012 年 11 月 29 日版 …「elses-xml-generate の新しい機能」「ファイル読み込みの新機能」を追加。細かい修正。
- 2012 年 11 月 27 日版 … v. 0.03.14(RC1) を対象に。

- 2012 年 11 月 22 日版 rev2 … v. 0.03.14(開発者版) 対応を追加。指摘を受けての、誤字等の修正。
- 2012 年 11 月 22 日版 … 初稿。v. 0.03.13 対応。「DST ワークフロー」追加。

## B 関連事項

### B.1 平均二乗変位 (MSD) と Participation Ratio (PR)

本説では波動関数の空間的広がりを表す概念である、平均二乗変位 (MSD) と Participation Ratio (PR) を説明する。

#### B.1.1 MSD

MSD は、統計における分散に他ならず、非周期系でしか定義できない。まずは、本来の定義を説明する；規格化された波動関数  $\phi(\mathbf{r})$  に対して、電荷分布を  $\rho(\mathbf{r}) \equiv |\phi(\mathbf{r})|^2$  で定義する。重心  $\mathbf{r}_c^{(\text{org})}$  と MSD 値  $M^{(\text{org})}$  はそれぞれ、

$$\mathbf{r}_c^{(\text{org})} \equiv \int \rho(\mathbf{r}) \mathbf{r} d\mathbf{r} \quad (1)$$

$$M^{(\text{org})} \equiv \int \rho(\mathbf{r}) |\mathbf{r} - \mathbf{r}_c|^2 d\mathbf{r} \quad (2)$$

で定義する。

一方、ELSES での MSD は、Mulliken 電荷を原子に局在した点電荷と仮定することで定義している；LCAO 係数から原子ごとに Mulliken 電荷  $\{q_I\}$  を計算する ( $I$  は原子 index,  $\sum_I q_I = 1$ )。  $I$  番目の原子の座標ベクトルを  $\mathbf{R}_I$  とすると、重心  $\mathbf{r}_c$  と MSD 値  $M$  はそれぞれ、

$$\mathbf{r}_c \equiv \sum_I q_I \mathbf{R}_I \quad (3)$$

$$M \equiv \sum_I q_I |\mathbf{R}_I - \mathbf{r}_c|^2 \quad (4)$$

で定義する。

MSD の性質をまとめる；

- 非周期系でしか定義できない。
- (長さ)<sup>2</sup> の次元を持つ。
- 波動関数が 1 原子に完全に局在している場合 (局在の limit)、ELSES での MSD 値は  $M = 0$  になる<sup>7</sup>。
- $x, y, z$  方向の MSD 値など、異方性が議論できる。

#### B.1.2 Participation Ratio

Participation Ratio は、量子局在 (Anderson 局在) で議論されてきた量である。[7, 8, 9, 10, 11] まずは、本来の定義を説明する；規格化された波動関数  $\phi(\mathbf{r})$  に対して、PR 値  $P^{(\text{org})}$  は 4 乗ノルムの逆数

$$P^{(\text{org})} \equiv \left\{ \int |\phi(\mathbf{r})|^4 d\mathbf{r} \right\}^{-1} \quad (5)$$

として定義される。PR の逆数 ( $1/P^{(\text{org})}$ ) は、Inverse PR (IPR) と呼ばれる。例として、閉領域  $D$  で一様であり、それ以外の領域では 0 となる場合を考える。閉領域  $D$  の体積を  $\Omega$  とすると、波動関数  $\phi(\mathbf{r})$  は

$$\phi(\mathbf{r}) \equiv \begin{cases} \frac{1}{\sqrt{\Omega}} & (\mathbf{r} \in D) \\ 0 & (\text{otherwise}) \end{cases} \quad (6)$$

<sup>7</sup> これは、Mulliken 電荷を点電荷と近似したことによる artifact である。実際には、原子基底の空間的な広がりがあるため、MSD 値は有限になる。

となる。対応する PR は

$$P^{(\text{org})} = \left\{ \int_D |\phi(\mathbf{r})|^4 d\mathbf{r} \right\}^{-1} = \left\{ \left( \frac{1}{\sqrt{\Omega}} \right)^4 \int_D d\mathbf{r} \right\}^{-1} = \left\{ \frac{1}{\Omega^2} \Omega \right\}^{-1} = \Omega \quad (7)$$

となる。つまり PR は、波動関数が存在する領域の体積 ( $P^{(\text{org})} = \Omega$ ) を与えている。

ELSES における PR は、LCAO 係数ベクトル  $\mathbf{v} \equiv (v_1, v_2, \dots, v_M)^T$  から、

$$P \equiv \left\{ \sum_i |v_i|^4 \right\}^{-1} \quad (8)$$

と定義される。例として、最初の  $m$  成分のみで一定値であり、その他の成分で 0 になる場合を考える；

$$\mathbf{v} \equiv \left( \frac{1}{\sqrt{m}}, \frac{1}{\sqrt{m}}, \dots, \frac{1}{\sqrt{m}}, 0_{m+1}, 0_{m+2}, \dots, 0_{M-1}, 0_M \right)^T. \quad (9)$$

対応する PR は

$$P \equiv \left\{ \sum_i |v_i|^4 \right\}^{-1} = \left\{ \left( \frac{1}{\sqrt{m}} \right)^4 \sum_{i=1}^m 1 \right\}^{-1} = \left\{ \frac{1}{m^2} m \right\}^{-1} = m \quad (10)$$

つまり PR は、LCAO 係数ベクトル  $\mathbf{v}$  の非ゼロ成分の個数を与えている。

(ELSES における) PR の性質をまとめる；

- 周期系でも非周期系でも定義できる。
- 無次元量。
- 波動関数が 1 基底に完全に局在している場合 (局在の limit)、PR 値は  $M = 1$  になる
- 逆数 (IPR) は 4 乗ノルムそのものであり、軌道ごと・原子ごとに分解できる (例：s 軌道のための IPR)。

## 参考文献

- [1] T. Hoshi, S. Yamamoto, T. Fujiwara, T. Sogabe, S.-L. Zhang, ‘An order-N electronic structure theory with generalized eigenvalue equations and its application to a ten-million-atom system’, J. Phys.: Condens. Matter **24**, 165502, 5pp (2012)  
<http://dx.doi.org/10.1088/0953-8984/24/16/165502> (無料で PDF がダウンロード可)
- [2] T. Hoshi, Y. Akiyama, T. Tanaka and T. Ohno, ‘Ten-million-atom electronic structure calculations on the K computer with a massively parallel order-N theory’, J. Phys. Soc. Jpn. **82**, 023710, 4pp (2013)  
<http://dx.doi.org/10.7566/JPSJ.82.023710> (無料で PDF がダウンロード可)
- [3] F. Ortmann, F. Bechstedt, W. G. Schmidt, ‘Semiempirical van der Waals correction to the density functional description of solids and molecular structures’, Phys. Rev. B **73**, 205101(2006).  
<http://dx.doi.org/10.1103/PhysRevB.73.205101>
- [4] CRC Handbook of Chemistry and Physics, 94th Edition, Ed. William M. Haynes, CRC Press, Boca Raton (2013) (ISBN:978-1466571143)
- [5] I. Kwon, R. Biswas, C. Z. Wang, K. M. Ho, and C. M. Soukoulis, ‘Transferable tight-binding models for silicon’, Phys. Rev. B **49**, 7242 (1994).  
[http://prb.aps.org/abstract/PRB/v49/i11/p7242\\_1](http://prb.aps.org/abstract/PRB/v49/i11/p7242_1)
- [6] ELSEs sample extra: ファイルサイズの都合でパッケージに入っていない sample data 集。現状では下記に用意されている (ファイル内容は同一, 14MB) :  
[http://www.damp.tottori-u.ac.jp/~hoshi/info/elses\\_sample\\_extra.tgz.zip](http://www.damp.tottori-u.ac.jp/~hoshi/info/elses_sample_extra.tgz.zip)  
[http://www.damp.tottori-u.ac.jp/~hoshi/info/elses\\_sample\\_extra\\_20150112.tgz.zip](http://www.damp.tottori-u.ac.jp/~hoshi/info/elses_sample_extra_20150112.tgz.zip)  
注 : 上記は暗号化 zip ファイルであり、暗号解除のパスワードは「ELSEs2007」。
- [7] R. J. Bell and P. Dean, ‘Atomic vibrations in vitreous silica’, Discuss. Faraday Soc. **50**, 55-61 (1970);  
<http://dx.doi.org/10.1039/DF9705000055>
- [8] R. J. Bell, ‘The dynamics of disordered lattices’, Rep. Prog. Phys. **35**, 1315 (1972).
- [9] D. J. Thouless, ‘Electrons in disordered systems and the theory of localization’, Phys. Rep. **13**, 94 (1974).
- [10] F. Wegner, ‘Inverse participation ratio in  $2 + \varepsilon$  dimensions’, Z. Physik B **36**, 209-214 (1980).
- [11] Takeo Fujiwara, Takashi Mitsui, and Susumu Yamamoto, ‘Scaling properties of wave functions and transport coefficients in quasicrystals’, Phys. Rev. B **53**, R2910 (1996).
- [12] Yuta Tsuji, Roald Hoffmann, Joel S. Miller, ‘Revisiting Ir(CO)<sub>3</sub>Cl’, Polyhedron **103**, 141-149 (2016).