

# プログラムの概略

平成 25 年 4 月 16 日

## 1 歴史

### 1.1 開発に関係した人々

xTAPP は慶應義塾大学の山内淳氏が開発してきた平面波基底第一原理計算プログラム群 TAPP の version 1.0.8 から 2000 年ごろ吉本芳英が分岐させたものである。

TAPP は、平面波基底擬ポテンシャル全エネルギー計算プログラム PPSF を基礎骨格として、東京大学工学部 押山淳氏のグループの東京大学 物性研究所 杉野修氏が開発した共役勾配法による対角化コードを取り込んで、山内氏が ultrasoft 擬ポテンシャルに対応させたプログラムが中心となっている。その経緯は、1990 年頃に塚田 捷 東京大学名誉教授の下、物質・材料研究機構 大野隆央氏、杉野氏、東京大学工学部 渡邊聡氏、山内氏が協力して開発を開始し、基本設計を杉野氏、バンド計算部分、擬ポテンシャル作成部分の coding は、それぞれ主に山内氏、渡邊氏が担当した。

PPSF は、鹿児島大学 永吉秀夫氏が開発した経験的局所擬ポテンシャル計算プログラムを参考に、筑波大学 白石賢二氏が一から coding した行列対角化による semilocal の非経験的擬ポテンシャル用平面波基底全エネルギー計算プログラムで、高度に対称性を利用した効率的アルゴリズムを特長としている。

PPSF/TAPP は、主として上記のメンバーによるコードからなるが、当時から現在まで、東京大学 理学部 物理学科 植村、上村、塚田、青木、常行研究室の関係者が直接間接に大きく寄与している。特に、兵庫県立大 島信幸氏、千葉大学 中山隆史氏 (PPSF/TAPP)、当時大学院生であった影島博之氏、木村栄伸氏、東北大学 赤木和人氏、東京理科大 (当時) 諏訪雄二氏 (TAPP) にはお世話になっている。

xTAPP 一般公開版計画には、東京大学理学系研究科 常行真司氏のご協力をいただいている。2012 年夏から作業が始まった xTAPP 一般公開版の新入力形式のプログラムは常行研究室の吉澤香奈子氏によるものが基礎になっている。また、山内氏、日立製作所 諏訪雄二氏にも作業に加わっていただいた。

### 1.2 開発に関係した資金など

xTAPP には、文部科学省科学研究費補助金「新学術領域」平成 22 年度～平成 26 年度コンピュータによる物質デザイン：複合相関と非平衡ダイナミクス、による成果が含まれている。

## 2 関係するプログラム

擬ポテンシャル作成は日立製作所 諏訪雄二氏が保守しているプログラム群 psv を xTAPP 向けに変更したものが必要であるがこのパッケージには含まれていない。擬ポテンシャルで原子を解

くプログラム `solps`、`solspin` はこのプログラム群に含まれている。

### 3 主な機能

- 平面波基底による第一原理計算
- Ultrasoft pseudo-potential
- semi local および hybrid 交換相関汎関数
- LDA(スピン偏極なし) と LSDA(スピン偏極あり)
- non linear core correction (partial core charge)
- 固有エネルギー、電荷分布、波動関数、原子に働く力、セルのストレス
- 構造最適化。セル固定とセル可変。
- 第一原理分子動力学 (BOMD)。NVE、NPT。
- バンド図、軌道電荷分布の積分値
- total DOS、projected DOS、電荷分布の積分値のエネルギー分解
- $\Gamma$  点計算および反転対称性がある場合の計算に対する計算の最適化
- OpenMP および MPI 並列化
- hybrid 汎関数については GPGPU 対応
- OpenDX[1] での電荷密度、波動関数などの可視化

[1] <http://www.opendx.org>

### 4 プログラムの概略

このパッケージには以下のプログラムが含まれている。

**inipot** 入力データから平面波基底を生成し、擬ポテンシャルを平面波基底で表現するデータを生成する初期化プログラムである。pefcos を除くその他のプログラムはこのプログラムで生成した初期化データを必要としており、最初に動かす必要がある。ただし、データその物は計算条件を固定すれば使いまわしできる。

**cgmrpt** 構造最適化を行うプログラムである。

**vbpef** cgmrpt で計算したローカルポテンシャルと波動関数 (hybrid 汎関数の場合) で決まるハミルトニアンを固定して、逆空間上に設定する  $k$  点の軌跡の上で、固有エネルギー、波動関数、軌道電荷分布の空間積分値、軌道電荷分布そのものを求めるプログラムである。いわゆるバンド図のデータを生成するのはこのプログラムである。

**vbstm** cgmrpt で計算したローカルポテンシャルと波動関数 (hybrid 汎関数の場合) から STM 像のシミュレーションを局所状態密度の積分値を用いて行うプログラムである。

**wfn2chg** cgmrpt で計算した波動関数から、電荷分布の空間積分を固有エネルギーで分解したものや projected DOS を計算するプログラムである。

**mdrpt** 第一原理分子動力学 (BOMD) を行うプログラムである。

**pefcos** cgmrpt で計算されているバンドの cos 展開データからバンド図を生成するプログラムである。

**xticonv** 入力ファイルの構造データを可視化ツール用に変換するツール

**strconv** 構造最適化の結果ファイルを可視化ツール用に変換するツール

入力データの詳細については inputformat.tex を参照すること。なお、入出力においてはとくに断らない限り原子単位系が使われる。

また、プログラムの定式化については formalism.tex を参照すること。

## 5 プログラムのコンパイルについて

### 5.1 はじめに

コンパイルのための設定が Makefile に用意されているのは Intel 環境と京コンピュータである。この他の環境で性能がでるようにコンパイルしたい場合には開発者に問い合わせるべきである。他の環境用の設定を編集して作成することは容易ではない。

コンパイルには MPI が必要である。Intel 環境では OpenMPI で開発が行われている。MPI-2 の機能は利用していない。ライブラリとして BLAS と LAPACK を利用する。Intel 環境では FFT を MKL で実行させるべきである。京コンピュータの場合 FFTW を用いるのがよい。

プログラムのコンパイルはソースのある src ディレクトリで行う。

### 5.2 コンパイルのための各種設定

```
Makefile-dist  
config.h-dist  
config90.h-dist
```

からそれぞれ

```
Makefile  
config.h  
config90.h
```

を生成する。config90.h は Fortran 90 用の config.h である。

config.h で設定すべき項目はほとんどないが、最大限保持できる対称操作の数 nsymq は利用目的によっては大きくする必要がある。

Makefile にある、config の例を編集して必要な設定を行い、make する。FFT ルーチン、LAPACK、密行列対角化ルーチン、交換相互作用の計算コードなどの選択を行うこと。

### 5.2.1 FFT

FFT3D\_OBJを選択する。FFTが使用するデータフォーマット (interleave または planar) に合わせて FFTGRID\_H\_SRC を選択しなければならない。fftgrid-scl.h は interleave 用、fftgrid-vec.h は planar 用である。MKL と FFTW は interleave と組み合わせること。

- MKL 用の fft3d\_mkl.o
- FFTW 用の fft3d\_fftw.o
- xTAPP 同梱の pfft3duz.o など

### 5.2.2 密行列対角化ルーチン

EIGSYSTEM\_OBJ を選択する。

- xTAPP 同梱の eigsystm.o
- LAPACK の eigsystm-lapack.o
- ScaLAPACK の eigsystm-scalapack.o

を選択する。最適化されたライブラリがあるならそれを使用する。intel MKL は高速である。

### 5.2.3 交換相互作用の計算コード

交換相互作用の計算コードには GPGPU に対応したものが用意されている。Intel 向けの config の場合、

```
# for NVIDIA CUDA
...
# for AMD GPU
...
# for CPU
...
```

となっている所からそれぞれ、NVIDIA CUDA を使う場合の設定、AMD GPU を使う場合の設定 CPU を使う場合の設定が選択できる。またこれらに合わせて SCGDGOBJ の選択も行うこと。

京コンピュータ向けの config の場合は CPU のみ選択できる。SCGDGOBJ の選択では CPU を選択すること。

## 5.3 MKL の LAPACK

MKL の LAPACK のライブラリのファイル名は composerxe-2011 で -lmkl\_lapack から -lmkl\_lapack95\_lp64 に変更になっているので注意すること。

## 5.4 ビルド

オプションなしで make すると、デフォルトですべてのプログラムが作成される。プログラムは src ディレクトリ中にできるので、必要であればこれを適切な場所にコピーして使う。

## 6 各プログラムの説明

### 6.1 inipot

inipot は入力データから平面波基底を生成し、擬ポテンシャルを平面波基底で表現するデータを生成する初期化プログラムである。pefcos を除くその他のプログラムはこのプログラムで生成した初期化データを必要としており、最初に動かす必要がある。ただし、データその物は計算条件を固定すれば使いまわしできる。固定されるべき条件が固定されていない場合、自動的に後続のプログラムは停止する。固定されるべき条件は、

- 平面波基底の数を決めるカットオフと  $\cos$  展開の係数の数を決めるカットオフ。
- 単位胞の形状（単位胞を構造最適化している場合は基準となる単位胞。）
- 対称性
- サンプル  $k$  点
- 原子種の数と原子の数

である。

このプログラムによって、平面波基底の数が決定される。このプログラムの入力に使った格子パラメータが基準となる。

このプログラム群では擬ポテンシャルの動径データから擬ポテンシャルの逆空間表示を連続関数としてフィッティングしたものを用いるが、これを inipot は計算している。

inipot は論理機番 10 から入力データを読み込む。MPI 並列実行できるが、MPI プロセスのうち、rank 0 のみが実際の処理を行い、他のプロセスは待っているだけである。

inipot は擬ポテンシャルの入力データを必要とするが、一番目の擬ポテンシャルは論理機番 34 から二番目は論理機番 34+1 からと言うように順に読み込まれる。この他、論理機番 28 から一番目の擬ポテンシャルに対応する原子電荷分布データ、論理機番 28+1 から二番目の擬ポテンシャルに対応する原子電荷分布データと順に原子電荷分布データを順に読み込ませることもできる。ここで読んだ原子電荷分布は初期ローカルポテンシャルを作成する時に使用させることができる。

論理機番 28 などに与えるファイルは、擬ポテンシャルを用いて原子を解くプログラム (solps、solspin) を用いて計算することができる。なお、このファイルには spin 偏極を含めることもでき、初期条件を偏極させたい場合に使用する。

また論理機番 13,14,15,40,22,73,74 に出力を行う。この他、論理機番 21 にも出力を行うが後方互換性のためである。内部のデータは使用されていない。

#### 6.1.1 replica 実行

cgmrpt 同様、replica 実行に対応する。詳細は cgmrpt での説明を参照すること。

### 6.2 cgmrpt

cgmrpt は電子状態計算を行い、必要なら構造最適化を行うプログラムである。

### 6.2.1 電子状態計算

電子状態計算は周期境界条件の元で行われる。そのため適切なサンプル  $k$  点を指定する必要がある。部分並進を含めた対称性が考慮されて既約なサンプル  $k$  点が選出され計算は既約なサンプル  $k$  点のみで行われる。

電子状態計算はいわゆる SCF ループの形式で行われる。1 SCF は固定されたハミルトニアンのもとで Davidson 法によって実行される逐次対角化である。

SCF ループで自己無撞着にされるローカルポテンシャルは Anderson extrapolation 法 [1] によってその収束を加速されている。Anderson extrapolation 法の混合パラメータは 0 から 1 までの実数で 0 に近いほど減速が強くなり SCF の不安定性を抑えられるが収束その物は遅くなる。

軌道への電子の詰め方を決める Fermi 面の smearing のアルゴリズムには、cos 展開、Methfessel-Paxton 法 [2,3]、Fermi 関数の三種類を使える。電子系の設定温度はこのアルゴリズムのパラメータである。また、系の中性状態から追加で電子を加えたり、全スピン偏極値を固定したりできる。

## 6.3 原子構造の最適化

原子構造の構造最適化は共役勾配法を用いて行われる。構造最適化は、原子数  $N$  について  $3N$  次元空間内のベクトルでできる探索方向（共役勾配方向）について次元構造最適化を行う主ステップとその探索方向を更新する副ステップの二重ループで実行されている。

構造最適化においては個別の原子ごとに固定条件を課することができる。指定した系の対称性は自動的に保持される。これらは、計算した力からこれらを破る成分が取り除かれることで実現される。収束判定などの計算はこの処理済みの力について行っている。

また、セル形状の最適化を行えるがこの場合には、電子の運動エネルギーを修正して実質的な基底のカットオフエネルギーを一定に保つ BTP[4] による運動エネルギー補正を使用すべきである。

### 6.3.1 入出力

cgmrpt は論理機番 10 から入力データを読み込む。MPI 並列実行している場合すべての MPI プロセスがこの入力データを読み込む。

cgmrpt は inipot が作成したデータを必要とし、論理機番 13,14,15,40,22,73,74 から読み込む。これら読み込みは rank 0 が担当する。21 からも読み込むがこれは後方互換性のためである。

cgmrpt は初期ローカルポテンシャルと初期波動関数を論理機番 11,95 からそれぞれ読み込むことができる。これら読み込みは rank 0 が担当する。なお平面波基底のカットオフや波動関数のサンプル  $k$  点が異なっても読み込むことができる。また初期ローカルポテンシャルとして、イオンのローカルポテンシャルか、読み込んだ原子の電荷分布の重ね合わせから計算したローカルポテンシャルを使うこともできる。読み込みを行わない場合初期波動関数はランダムに生成される。

構造最適化時の原子構造、全エネルギーなどの情報は rank 0 が標準出力のログに出力する。

cgmrpt は計算終了時にローカルポテンシャルと電荷分布を論理機番 11, 25 にそれぞれ出力する。また論理機番 96 に波動関数を書き出すことができる。このほか論理機番 99 に構造最適化した原子構造、全エネルギー、原子に働く力、ストレス、スピン偏極度など結果の要約をまとめたものを出力する。これら書き出しは rank 0 が担当する。

出力された電荷分布、ローカルポテンシャル、波動関数を OpenDX で読み込める形式に変換するプログラムはそれぞれ rho2dx, lpt2dx, wfn2dx として xTAPP-util に含まれている。

構造最適化の計算の継続を行うために必要なデータは、論理機番 70,71,72 に rank 0 から随時書き出されている。計算が時間切れとなって終了した場合、次の計算実行時に継続を指示すると、これらデータとローカルポテンシャル、波動関数から構造最適化が継続される。

この他、cgmrpt は 51 に固有エネルギーの cos 展開データを記録する。この記録は pefcos のためのものである。

### 6.3.2 論理機番 99 のフォーマット

```
&struct_data
lattice_factor = ...
lattice_list = ...
total_energy = ...
stress_tensor = ...
fermi_energy = ...
number_element = ...
number_atom = ...
spin_polarization = ...
abs_spin_polarization = ...
/

# valence_charge, nucleous_charge
zo zn
...

# atom_kind, atom_position by cell coordinate
atom_kind, pos_a, pos_b, pos_c
...

# force by Cartesian coordinate
force_x force_y force_z
...
```

となっている。入力ファイルフォーマットと同じキーワードは同じ意味を持つ。その他のキーワードは

**total\_energy** 全エネルギー  $E$

**stress\_tensor** ストレステンソル

$$-\frac{\partial E}{\partial \epsilon_\nu}; \quad \nu \text{ は } xx, yy, zz, yz, zx, xy \text{ の順}$$

**fermi\_energy** フェルミエネルギー。それぞれのスピン成分ごとに個数を決めるモードがあるので二つある。

**spin\_polarization** スピン偏極

$$\int_{\Omega} (\rho_2(\mathbf{r}) - \rho_1(\mathbf{r})) d\mathbf{r}$$

**abs\_spin\_polarization** 絶対スピン偏極 (アンチフェロの時に値が0にならないもの。)

$$\int_{\Omega} |\rho_2(\mathbf{r}) - \rho_1(\mathbf{r})| d\mathbf{r}$$

**force\_x, force\_y, force\_z** 原子に働く力 (デカルト座標)

である。

### 6.3.3 replica 実行

計算機システムによっては、入力パラメータを多数用意してそれを並列実行させることが困難なものがある。このようなシステムで複数入力パラメータでの並列実行を行うために replica 実行機能が用意されている。レプリカ実行を行うためには通常の入力データを複数用意し、cgmrpt 自体にはそれらを複数並列実行するための特別な入力を与える。

入力の書式は

```
REPLICA
color(1) key(1)
color(2) key(2)
...
color(nproc) key(nproc)
dirprefix ifname ofname
```

となっている。ここで `color(i)` と `key(i)` は、`MPI_COMM_WORLD` で `rank=i` の MPI process を所属させる新しいコミュニケータのカラー番号と新しいコミュニケータでの rank を決めるためのキーである。詳しくは `MPI_COMM_SPLIT` のマニュアルを参照すること。同一のカラーの rank の MPI process は同一の新しいコミュニケータを構成し、その中で一つの入力ファイルが実行される。実行される入力ファイルは、cgmrpt が実行されているディレクトリ以下に、`dirprefix` に続けてカラー番号を先頭に 0 を補って 5 桁に揃えた数字をつなげた名前のディレクトリの中の `ifname` の名前のファイルである。そしてプログラムの出力はこのディレクトリの `ofname` の名前のファイルに格納される。

なお、これらのディレクトリは `inipot` が生成するデータファイルをそれぞれ含んでいなければならない。`inipot` も replica 実行できるので、`replica` 実行でそれらを作成させるのが良い。

例えば、

```
REPLICA
0 0
1 0
p. si.cg cgmrpt.log
```

の場合、`rank 0` と `rank 1` の MPI process がそれぞれディレクトリ `p.00000` と `p.00001` でその中の `si.cg` を入力ファイルとして実行される。出力はこれらディレクトリの中の `cgmrpt.log` に出力される。

- [1] Donald G. Anderson, J. Assoc. Computing Machinery, 12, 547 (1965)
- [2] M. Methfessel and A.T. Paxton, PRB 40 (1989) 3616
- [3] G. Kresse and J. Furthmueller, Computational Materials Science 6 (1996) 15
- [4] M. Bernasconi et al., J. Phys. Chem. Solids **36** 501-505 (1995).

## 6.4 vbpef と pefocs

vbpef は cgmrpt で計算したローカルポテンシャルと波動関数 (hybrid 汎関数の場合) で決まるハミルトニアンを固定して、逆空間上に設定する  $k$  点の軌跡の上で、固有エネルギー、波動関数、軌道電荷分布の空間積分値、軌道電荷分布そのものを求めるプログラムである。いわゆるバンド図のデータを生成するのはこのプログラムである。

一方で pefocs はバンド図のデータだけを cgmrpt で計算した固有エネルギーの cos 展開から計算するプログラムである。そのため cos 展開の対象となったバンドの範囲だけ計算できる。

$k$  点の軌跡は複数区間のつながった線分で設定する。また軌道電荷分布の積分は指示した点を中心とする球状、またはセルの軸に平行になる平板状のいずれかで行える。

計算したバンドの固有エネルギーデータは論理機番 50 に出力される。これを xTAPP-util の vbpef2gp-ksa で処理するとバンド図を描ける。

計算した軌道電荷の積分データは論理機番 51 に出力される。計算した  $k$  点の波動関数データは 58, 計算した軌道電荷データは 57 にそれぞれ出力される。xTAPP-util 中の wfk2dx および rok2dx で 58 および 57 を OpenDX 向きのデータ形式に変更できる。

vbpef は inipot が作成したデータを必要とし、論理機番 15,40,22,73,74 から読み込む。これら読み込みは rank 0 が担当する。

TAPP にあった vbwfn の機能は vbpef に統合されているので、必要ならこちらを使うこと。

## 6.5 vbstm

cgmrpt で計算したローカルポテンシャルと波動関数 (hybrid 汎関数の場合) から STM 像のシミュレーションを局所状態密度をフェルミエネルギーからバイアス電圧まで積分して行うプログラムである。

フェルミエネルギーは、入力として与えなければならない。

計算した STM 像は論理機番 80 に出力される。この出力ファイルを OpenDX で読み込める形式に変換するプログラム stm2dx が xTAPP-util に入っている。

mdrpt は inipot が作成したデータを必要とし、論理機番 13,14,15,40,22,73,74 から読み込む。これら読み込みは rank 0 が担当する。21 からも読み込むがこれは後方互換性のためである。

## 6.6 wfn2chg

cgmrpt で計算した波動関数から、電荷分布の空間積分を固有エネルギーで分解したものや projected DOS を計算するプログラムである。

このプログラムは実際に MPI 並列動作するものではないが mpirun で起動する必要がある。

電荷分布の空間積分は指示した点を中心とする球状、またはセルの軸に平行になる平板状のいずれかで行える。

projected DOS の計算に必要な動径波動関数は論理機番 18 から読み込む。このファイルは

```
nsmp1 wavil
r(1) wav(1)
r(2) wav(2)
...
r(nsmp1) wav(nsmp1)
```

の形式のもので、主量子数  $n$ 、軌道角運動量  $l$  の動径波動関数を  $R(n, l; r)$  とするとき、1 から  $nsmp1$  までのログメッシュ  $r(1:nsmp1)=r$  上の角運動量  $wavi1=l+1$ 、の動径波動関数  $wav(1:nsmp1)=rR(n, l; r)$  を指定するものである。これは擬ポテンシャルで原子を解くプログラム (solps) で作ることができる。

計算した積分値や projected DOS は 55 に出力される。このデータを処理してエネルギーの関数としてのグラフを描くプログラムが xTAPP-util にあり、積分値用が wfchg2pdos、projected DOS 用が ltzpdos である。これらのプログラムは今のところ、これらに与えるローレンチアンとの畳み込みによってエネルギー依存性の計算を行っている。なお、ltzpdos の出力には total DOS も含まれている。

projected DOS の角運動量は  $wavi1 = 3(l = 2; d \text{ 軌道})$  まで指定可能である。また出力における磁気角運動量の並び順はカーテシアン座標を用いて、 $p$  軌道の場合

$$x, y, z$$

$d$  軌道の場合、

$$xy, yz, 3z^2 - 1, zx, x^2 - y^2$$

の順となっている。

また、cgmrpt で計算した  $k$  点の軌道電荷を出力させることも出来、これは論理機番 57 に出力される。このファイルは vbpef と同様に xTAPP-util の rok2dx で OpenDX 用のフォーマットに変換できる。

wfh2chg は inipot が作成したデータを必要とし、論理機番 13,14,15,40,22,73,74 から読み込む。21 からも読み込むがこれは後方互換性のためである。

## 6.7 mdrpt

mdrpt は第一原理分子動力学 (BOMD) を行うプログラムである。mdrpt は cgmrpt と同様の電子状態計算を行って原子に働く力を計算する。

速度ベレ法による NVE アンサンブル、および Stern のアルゴリズム [1] による NPT アンサンブルが扱える。なお、barostat が止まるパラメータを設定することで NVT アンサンブルにすることもできる。

分子動力学の原子位置などの記録は、論理機番 16 にバイナリで出力される。出力は rank 0 が行う。このファイルを読み取ってアスキー形式で出力するプログラム scanmdlog が xTAPP-util に入っている。

初速度は設定することもできるが、設定温度に合うようなランダムな初速を与えることもできる。

MD ステップ毎に波動関数とローカルポテンシャルを次のステップに補外させて収束を加速することができる。[2,3]

mdrpt は inipot が作成したデータを必要とし、論理機番 13,14,15,40,22,73,74 から読み込む。これら読み込みは rank 0 が担当する。21 からも読み込むがこれは後方互換性のためである。

mdrpt は初期ローカルポテンシャルと初期波動関数を論理機番 11,95 からそれぞれ読み込むことができる。これら読み込みは rank 0 が担当する。なお平面波基底のカットオフや波動関数のサンプル  $k$  点が異なっても読み込むことができる。また初期ローカルポテンシャルとして、イオンのローカルポテンシャルか、読み込んである原子の電荷分布の重ね合わせから計算したローカルポテンシャルを使うこともできる。読み込みを行わない場合初期波動関数はランダムに生成される。

mdrpt は計算終了時にローカルポテンシャルと電荷分布を論理機番 11, 25 にそれぞれ出力する。また論理機番 96 に波動関数を書き出すことができる。これら書き出しは rank 0 が担当する。

分子動力学が設定ステップまで進まずに終了した場合、論理機番 75 に継続実行のためのデータが rank 0 によって書き出される。次の実行で継続を指示するとこのデータが rank 0 から読み取られて継続実行が始まる。この際、ローカルポテンシャルと波動関数の読み込みも同時に必要である。

[1] H.A. Stern, J. Compt. Chem, Vol. 25, No. 5, 749-761 (2004)

[2] T.A. Arias, M.C. Payne, J.D. Joannopoulos, Phys. Rev. B 45 (1992) 1538

[3] Dario Alfe, Comp. Phys. Comm., 118 (1999) 31

## 6.8 xticonv

xticonv は入力ファイルの中の原子構造データを可視化ツールのために変換するプログラムである。対応フォーマットは CIF(<http://www.iucr.org/resources/cif>) と XYZ である。プログラムは

```
$ xticonv [fmt] [file name]
```

として起動する。ここで [fmt] は

**cif:** CIF 形式

**xyz:** XYZ 形式

**xyzpr:** XYZ 形式であるが、セルの境界付近 (セルの各周期の方向に 0.1 a.u. まで) に位置する原子を重複して出力

であり、[file name] は変換したい入力ファイル名である。変換結果は標準出力に出てくる。

## 6.9 strconv

strconv は cgmrpt が論理機番 99 に出力する構造最適化した原子構造を可視化ツールのために変換するプログラムである。対応フォーマットは CIF(<http://www.iucr.org/resources/cif>) と XYZ である。プログラムは

```
$ strconv [fmt] [file name]
```

として起動する。ここで [fmt] は

**cif:** CIF 形式

**xyz:** XYZ 形式

**xyzpr:** XYZ 形式であるが、セルの境界付近 (セルの各周期の方向に 0.1 a.u. まで) に位置する原子を重複して出力

であり、[file name] は変換したい入力ファイル名である。変換結果は標準出力に出てくる。

## 7 擬ポテンシャル

TAPP-1.0.8 で用いられていた形式の擬ポテンシャルのフォーマットを用いるがヘッダは異なっており、

```
#xPSV-1
# ATOM NAME: {atom_name}
# {comment}
# ...
xcname
zo zn
...
```

で始まるようになっている。ここで{atom\_name}は原子名を表す文字列、{comment}はその他のコメントである。{atom\_name}はログ中で出力される。xcname は交換相関ポテンシャルのタイプを表す。zo はこの擬ポテンシャルのイオンの価数であり、zn はこの擬ポテンシャルの原子番号である。なお、zn は省略可能である。元々のフォーマットでは zn は存在していない。

TAPP-3.0 系列のフォーマットと TAPP-1.0.8 のフォーマットでは initial spin のデータが増設された違いがある。このデータは xTAPP では使えないので削除する必要がある。

## 8 原子の電荷分布

28 から始まる論理機番から入力する原子電荷分布は

```
nsmpl
r(1) rho1(1) rho2(1)
r(2) rho1(2) rho2(2)
...
r(nsmpl) rho1(nsmpl) rho2(nsmpl)
```

の書式で与える。ここで nsmpl は動径のログメッシュの数であり、r はログメッシュ、rho1、rho2 は電荷の動径分布である。rho2 がない場合にはスピン偏極なし、ある場合にはスピン偏極ありとして扱われそれぞれ成分 1、2 となる。このデータは solps か solspin を用いて計算することができる。

## 9 ログの見かたについて

### 9.1 実行条件の確認

ログの先頭部分には入力ファイル形式と同じ形式で BEGIN INPUT から END INPUT まで実行に使われたパラメータが出力されている。入力で省略したものにプログラムが当てはめた値もここで確認できる。

### 9.2 SCF の収束の確認

SCF の収束はログの SQUARE ERROR = で確認する。この定義はデフォルトでは

$$\sum_{\sigma} \frac{1}{\Omega} \int_{\Omega} |V_{\sigma,new}(r) - V_{\sigma,old}(r)|^2 dr = \sum_{\sigma,G} \left( \frac{1}{\Omega} |V_{\sigma,new}(G) - V_{\sigma,old}(G)| \right)^2$$

である。この定義は収束度がシステムサイズに依存しにくくなるので良い。

しかし、TAPP-3.0との互換性を指定している場合は、これをさらに $2 * nvxyz$ で割ったものが定義となる。ここで $nvxyz$ はローカルポテンシャルの逆空間での全メッシュ数である。

なお、この収束の定義では非占有状態の収束の度合いを知ることは出来ない。このためのちの計算の都合で非占有状態を収束させたいときには別途 IKS, EIGEN, RESID, SIN, EPS を調べる必要がある。これは SCF 一回につき、一塊のログが  $k$  点の数だけ出るものであるが、EIGEN が一回前の SCF との固有値の変化の絶対値を表している。これが十分に小さくなるようにしなければならない。なお、非占有状態を収束させるためには SCF を回すことではなく対角化ルーチンを回すことが必要である。そのため非占有状態を特に収束させたいときには SQUARE ERROR = が十分に小さくなった状態で davidson\_number\_diag を大きめに取って対角化を十分に行わなければならない。

### 9.3 全エネルギー

SCF の収束の過程で出力される ETEIG はバンドエネルギーから簡易に計算した変分量になっていない全エネルギーである。ただし hybrid 計算の場合にはこの計算が低コストでできないため正しく行っていない。ETEIG は力の計算が終わった後に出力される変分量になっている全エネルギー”TOTAL ENERGY”に近い値になるものである。

### 9.4 構造最適化と力の残差

構造最適化の過程で収束を見るには MAX FORCE を調べれば良い。これは原子に働く力のうち最大のもの [hartree/bohr] である。セル形状を最適化している場合、これに加えて MAX STRESS を調べる。これは、

$$\frac{1}{N_{atom}} \left( \frac{dE}{d\epsilon_\nu} - p_{ext} V \right); \quad \text{for } \nu = xx, yy, zz$$
$$\frac{1}{N_{atom}} \left( \frac{dE}{d\epsilon_\nu} \right); \quad \text{for } \nu = yz, zx, xy$$

の最大である。

### 9.5 実行プロファイルの確認

SCF が終了する時と、力の計算が終了するときにそこまでの実行プロファイルが profile: time count に続いて出力される。最初の数字が経過時間の合計、次の数字が実行回数である。

### 9.6 全サンプル $k$ 点数とサンプル $k$ 点の位置の確認

inipot が全サンプル  $k$  点を決定している。このログのうち NI=で現れているのが全サンプル  $k$  点の数である。なおこの数は設定した  $k$  点メッシュより多くなることがある。これは設定した  $k$  点メッシュに対称性でつながっている  $k$  点も数えるためである。また、この NI=の直後にサンプル  $k$  点の番号とその位置の対応が一行に 2 セットずつ出ている。1 セットの数字のうちもっとも右側のものは、サンプルした  $k$  点が対称操作を有効活用する割合を示しており 1 が最大である。位置は逆格子の基本ベクトル単位で表示されている。

## 9.7 平面波基底の数の確認

inipot のログにサンプル  $k$  点の番号ごとの平面波基底の数が  $iks, nkm =$  で記録されている。

## 9.8 各種メッシュ

**電荷分布:** inipot のログに NRX, NRY, NRZ で現れている。それぞれの 2 倍が実際に使われるメッシュ数である。NRXYZ は総メッシュ数である。

**波動関数の実空間メッシュ:** inipot のログに nwx, nwy, nwz で現れている。それぞれの 2 倍が実際に使われるメッシュ数である。

## 10 プログラムの運用について

### 10.1 計算条件の改良

同じ計算条件ではない波動関数、ローカルポテンシャルを読むことが出るので計算条件の逐次改良が可能である。

### 10.2 メモリ

最近のハードウェアはメモリが多く搭載されているため、波動関数などのデータはすべてメモリ上に載せられている。オリジナルの TAPP のように途中でディスクに置くことはできない。

### 10.3 ファイル I/O

通常 Fortran の処理系は環境変数をセットすることであらかじめ論理機番を特定のファイルに結びつけておくことができる。この機能を用いて入出力をセットアップするスクリプトを用意してその中から各プログラムを呼び出すようにするのが良いだろう。

ただし、gfortran(version 4.4.5) ではこの機能は存在しないようである。fort.\*のファイルがそのまま各論理機番に対応してしまうので、あらかじめ fort.\*を目的のファイルへのシンボリックリンクとすることで対応しなければならない。

また MPI 並列実行している場合、ログを標準出力に出力するのは rank 0 だけである。その他の I/O も基本的に rank 0 が担当するようにしてある。

### 10.4 MPI と環境変数と論理機番

論理機番を環境変数を使ってプログラムに渡す場合、MPI の実装によっては mpirun の引数の中に記述するなど特定の方法を取らない限りプログラムに環境変数を伝達できないことがあるので、注意が必要である。

### 10.5 波動関数の分散

波動関数の分散は平面波基底で行っており、FFT が必要になるときだけデータをバンド分散に持ち替えている。ローカルポテンシャル、電荷分布などは各 MPI プロセスがコピーを持っている。

## 10.6 hybrid 並列

大規模な系の場合、分散できないメモリ量が増大するため、hybrid 並列を用いて 1 MPI プロセスあたりのメモリ量を増やすことが必須になる。

また計算機システムによっては hybrid 並列の方が通信などが高性能になるものもある。また、バンド × バンド程度の大きさの密行列の固有値問題を解く時、この規模の問題が shared memory 並列では性能が出るが、MPI 並列では性能がでないこともある。ここ部分が高並列では主要な律速の一つなので、高速化が重要である。

hybrid 実行する場合、1 MPI process が 1 つの uniform memory access の単位、通常は 1 chip、を構成するのが通常良い。ただし、非常に高並列になる場合にはもっと大きい単位をとると効果ができる場合がある。またこの場合には、FFT ルーチン、LAPACK、密行列対角化ルーチンが shared memory 並列となるようライブラリなどを選択すること。

### 10.6.1 OpenMP

hybrid 並列を OpenMP によって行っている場合、OpenMP のスレッドに割り当てるスタックの大きさを系の大きさに応じて大きくする必要があることがある。Intel Fortran ではある。この設定は環境変数 OMP\_STACKSIZE で行うことができ、OMP\_STACKSIZE=256MB と指定すれば 256MB が確保される。なお、各 MPI プロセスに環境変数を渡す方法は MPI の実装によって異なるので注意すること。OpenMPI の場合は mpirun にオプション -x OMP\_STACKSIZE を渡すことで行える。

## 10.7 FFTW

FFTW を使っている場合、wisdom はプログラムが実行されるディレクトリにファイル xtapp\_fftw\_wisdom.dat があればそこから読み出される。このファイルはすべての MPI プロセスがアクセスする。またプログラム終了時には wisdom が xtapp\_fftw\_wisdom.new.dat として記録される。

FFTW の初期化ではデフォルトでは詳細にもっとも良い条件を出すように設定してある。そのため第一回目の FFT の初期化には時間がかかることがある。

## 10.8 処理系による namelist の違い

namelist 中の文字列を引用符でくくらないとならない処理系があるので注意が必要である。gfortran はそのような例である。

## 10.9 プログラムのテスト例

パッケージ xTAPP-test にはプログラムのテスト例が集められている。この例は同時に実行例としても扱える。Makefile の先頭を調節して make することでテスト例を実行できる。この中にはプログラムの実行スクリプトも含まれており、Fortran の論理機番の割り当ての処理の参考にできる。

## 10.10 OpenDX による可視化の例

OpenDX による可視化プログラムの例が xTAPP-test に含まれている。Makefile の TARGET\_TOOL で定義されているターゲットを make すると OpenDX 用のデータができるので、これを opendx ディレクトリにある OpenDX のネットワークの例を用いて可視化する。この例の実行には、OpenDX の追加モジュールの CMSP が必要である。これは、Octopus[1] の web ページから入手できる。

[1] <http://www.tddft.org/programs/octopus/>

## 10.11 プログラムの理解

TAPP-3.0 系列とは元々同じものであったため変数名、サブルーチン名は共通点が多い。そのため TAPP-3.0 系列の文書にある情報は参考になる。