

MODYLAS Reference Manual

Version 1.0.2

4 Nov. 2014

目次

- 1 はじめに
- 2 ライセンス
- 3 動作環境
 - 3.1 マシン環境
 - 3.2 コンパイル方法
- 4 入出力ファイル
 - 4.1 入出力ファイル一覧
- 5 入力ファイルの作成
 - 5.1 入力ファイルのフォーマット概要
 - 5.2 入力ファイルの作成方法
 - 5.3 入力ファイルのバイナリー化について
- 6 実行方法
 - 6.1 実行方法
 - 6.2 再スタートの方法
- 7 計算結果のモニター
 - 7.1 物理量各種の出力結果
 - 7.2 軌跡データの座標ファイルへの変換
- 8 MODYLAS の並列化
 - 8.1 並列化実装
 - 8.2 空間の領域分割
- 9 [付録]入力キーワード・オプション一覧
 - 9.1 計算キーワード(.mddef)
 - 9.2 力場(.mdff)
 - 9.3 初期座標・速度(.mdxyz)
 - 9.4 原子の位置拘束(.posiconst)
- 10 参考文献
- 11 開発者

(Reference Manual 第3版 2014年11月4日 改定)

1. はじめに

MODYLAS (MOlecular DYnamics software for LArge System)は、物理、化学、生体の分野における巨大系に適した汎用分子動力学シミュレーションソフトウェアである。MODYLAS はほとんどの標準的な分子動力学計算の手法を備えている。特に、長距離静電相互作用計算については高速多重極子展開法(FMM)と周期境界条件下での多重極子についての Ewald 法により厳密に計算することができる。温度制御には能勢-Hoover-チェーン法、圧力制御には Andersen 法を用いることができ、*NVT*, *NPT* アンサンブルを生成できる。運動方程式の数値積分にはマルチタイムステップに対応した rRESPA 法が採用されている。原子間の距離拘束には SHAKE/RATTLE/ROLL 法により扱われる。力場として全原子力場である CHARMM22 with CMAP, CHARMM36 with CMAP が利用できる。近い将来, AMBER および OPLSAA にも対応する。さらに将来のアップデートにおいて熱力学積分法による自由エネルギー計算も利用可能となる。

また、MODYLAS には分子動力学計算の高並列化のために開発された新しい手法がいくつか採用されている。通信と演算操作に対して実質的にデータコピーが発生しないアルゴリズム、通信レイテンシー最小のアルゴリズム、そして FMM 上層部の多極子モーメントに関する並列バケツリレー通信アルゴリズムにより優れたスケーラビリティを実現している。さらに、ブロック化された演算操作によりメモリからキャッシュへの再読込を避けることができ、非常に小さいキャッシュミス率を達成している。京コンピュータ 65536 ノードを用いた MODYLAS のベンチマークテストでは、1000 万原子系に対する 1 ステップの通信を含めた全体の計算時間は 5 ms を記録した。これは 1000 万原子系の計算が 1 日で 35 ns 進むことに相当する。本プログラムを用いることにより、ウイルス、リポソーム、タンパク質凝集体、ミセル、高分子などの大規模系の研究が可能となる。

2. ライセンス

MODYLAS ソフトウェアプログラム使用許諾契約書

MODYLAS 著作権管理者（以下「ライセンサー」という）は、MODYLAS（以下「本プログラム」という）の著作権者として別紙1の著作権者一覧に記載された全ての者（以下「著作権者ら」という。）を代表して、申込みのページで確認できる人（以下「ライセンシー」という）に対して、本契約に定める条件に従って本プログラムを使用する権利を無償で許諾する。

第1条（定義）

1. 商業目的の使用とは、MODYLAS を有償で販売すること、MODYLAS を利用して有償で受託計算やコンサルティングを行うこと等、これらの例に限らず、MODYLAS を直接ビジネスに使うことを意味する。非商用利用は、それ以外の利用で、アカデミックな利用だけでなく、民間企業における研究開発および試用も含まれる。
2. 軍事利用とは、MODYLAS を使ったの兵器開発やその他の軍事に直接利用することを意味する。非軍事利用はそれ以外の利用を意味する。

第2条（本プログラムの提供）

1. 本プログラムの提供は、非軍事利用、非商用利用の研究目的（以下「本件使用目的」という）のために行われるものとする。MODYLAS の商用利用を希望される方は、末尾の連絡先までご連絡をお願いします。
2. ライセンサーは、ライセンシーに対し、本プログラムをソースコード形式により提供する。
3. ライセンサーは、ライセンシーに対し、本プログラムに関するリファレンスマニュアル、チュートリアルテキスト、サンプル入力と出力（以下、「マニュアル等」という）を提供する。

第3条（許諾された権利）

1. ライセンサーはライセンシーに対して、本契約の条件に従い、本プログラムを本件使用目的に使用する非独占で移転不可の権利を許諾する。ライセンシーは、いかなる場合であっても、本件使用目的以外の目的に本プログラムを使用することができない。
2. ライセンシーが営利を目的とする組織の構成員又は従業員である場合には、ライセンシーは、本件使用目的の範囲内に限り、本契約の各条項に従って本プログラム及び派生プログラム（本条3項で規定）を当該組織のための研究開発および試用に用いることができる。
3. ライセンシーは、本件使用目的のために必要な範囲で本プログラムを改変することができる。ライセンシーによる本プログラムの改変の結果、二次的著作物であるプログラム（以下「派生プログラム」という）が作成された場合、ライセンシーはライセンサーに対して直ちにかかる事実を通知しなければなら

らない。ライセンサーが要請した場合、ライセンシーはライセンサーに対して派生プログラムをソースコード形式で提供し、ライセンサーに対して、派生プログラムを使用及び改変し、かつ研究目的のために無償で（配布のための実費の補償を除く）第三者に配布及び使用許諾する非独占的な権利を許諾しなければならない。

4. ライセンシーは、ライセンサーの事前の書面による承諾なく、(i) 第三者に対して本プログラム又は派生プログラムを提供し、又は使用を認めてはならず、(ii) 本プログラム又は派生プログラムを公開してはならず、(iii) 本プログラム又は派生プログラムを本件使用目的以外に使用してはならない。

5. 前項に関わらず、同じ組織の構成員又は従業員である他のライセンシーとの間で、派生プログラムも含め情報交換、技術交流を行うことができる。

6. ライセンシーは、ライセンサーに対して、ライセンシーの氏名、所属する組織の名称、所属する組織における肩書、電子メールアドレスを申込みページより通知するものとし、本契約の期間中にこれらの事項につき変更があった場合には遅滞なくライセンサーに通知するものとする。

7. ライセンシーは、本プログラム及び派生プログラムを細心の注意を払って秘密として管理し、ライセンシーの故意又は過失による本プログラム又は派生プログラムの情報漏洩に関してライセンサーに対して責任を負う。

8. ライセンシーは、以下に設置される利用環境において本プログラムを試用または利用することができる。

- ・ 大学共同利用機関法人 自然科学研究機構 計算科学研究センター
- ・ 大学共同利用機関法人自然科学研究機構 分子科学研究所 計算分子科学研究拠点 (TCCI)
- ・ 公益財団法人 計算科学振興財団 (FOCUS)
- ・ 独立行政法人 理化学研究所 計算科学研究機構 (AICS)

但し、実際のマシン利用にあたっては、各々の環境の利用規定などに従うものとする。

第4条（商業目的の使用の禁止）

1. ライセンシーは、ライセンサーの事前の書面による明示的な承諾なく、本プログラム又は派生プログラムを研究開発および試用以外の商業目的に使用してはならない。

2. ライセンシーが営利を目的とする組織の構成員又は従業員である場合、ライセンシーが本プログラム又は派生プログラムを研究開発および試用以外の当該営利組織の業務目的に使用することは、前項で禁止される商業目的の使用に該当するものとする。

第5条（所有権及び権利の保持）

本プログラムの著作権者らは、本プログラムに対する所有権を含む全ての権利を保持する。本プログラムは、著作権及び他の知的財産権に関する法律及び条約で保護されている。

第6条（情報提供）

ライセンサーが要請した場合、ライセンシーは、ライセンサーに対して、本プログラム又は派生プログラムの使用に関連する情報（フィードバック、コメント、バグ情報等）を提供するものとする。

第7条（構成部分の使用の禁止）

本契約による許諾は、本プログラムを一つの分離不可能なプログラムとして使用することの許諾である。ライセンサーが明示的に書面で許可しない限り、ライセンシーは、本プログラム（派生プログラム中の本プログラムに由来する部分を含む。以下同様）のいかなる構成部分も本プログラムから分離してはならず、本プログラムのいかなる構成部分も別のプログラムに組み込んではならない。

第8条（契約期間及び解除）

1. 本契約は、MODYLAS プログラムファイルの一部のダウンロード時点または第3条8項における試用または使用開始時点（いずれか早い時点）に発効し、本条の規定により解除されない限り存続するものとする。
2. ライセンサーは、ライセンシーが本契約に違反した場合、即時に本契約を解除することができる。
3. いずれの当事者からも、14日前までに相手方に通知することにより、いつでも本契約を解除することができるものとする。ライセンシーへのメールによる通知がエラーとなる場合は、ライセンサーは通知に代えて、その内容をホームページ上に掲示することができる。
4. 本契約が解除された場合、ライセンシーは、本プログラム及び派生プログラムの使用を全て中止し、本プログラム及び派生プログラム、それらの構成部分並びにそれらの全ての複製物を破棄しなければならない。
5. 本契約第3条3項、第5条、第6条、第8条4項、第9条、第11条、第12条、第13条、第14条の規定は本契約が解除された場合でも有効に存続するものとする。

第9条（成果の公表）

ライセンシーが本プログラム又は派生プログラムを使用した研究成果を論文などの方法で公表する場合は、以下の本プログラムに関連する文献を引用するものとする。

Andoh, Y. et al., "MODYLAS: A Highly Parallelized General-Purpose Molecular Dynamics Simulation Program for Large-Scale Systems with Long-Range Forces Calculated by Fast Multipole Method (FMM) and Highly Scalable Fine-Grained New Parallel Processing Algorithms,"

J. Chem. Theory Comput., 2013, 9 (7), pp 3201–3209.

DOI: 10.1021/ct400203a

前述に拘わらず、ライセンシーが他のプログラムと計算性能を比較することを主目的としてその成果を論文、学会発表などの方法で公表する場合は、測定開始の1ヶ月以上前にライセンサーに相談するものとし、学会発表も含めて投稿前にライセンサーの許可を得るものとする。

第10条（ライセンサーの権限）

ライセンサーは、著作権者らから本契約に基づく使用許諾をする権限を付与されていることを、ここに表明及び保証する。

第11条（完全合意）

本契約は、その主題に関する当事者間の完全な合意を構成するものであり、その主題に関する全ての事前の書面又は口頭による合意、理解、提案及び表明に取って代わるものである。

第12条（免責）

ライセンサーは、本プログラムの欠陥、本プログラムの使用又は本プログラムの使用不能を含む（但しこれらに限られない）本プログラム又は本ライセンスに関連して生じ得るいかなる損害についても、ライセンシーに対して責任を負わない。ライセンサーが、かかる損害の可能性について事前に知らされていた場合であっても、同様である。

第13条（準拠法）

本契約は、日本法に準拠し、日本法に従って解釈される。

第14条（管轄）

本契約から発生する又は本契約に関連する全ての紛争は、日本国の名古屋地方裁判所の専属管轄に服する。

【連絡先】 MODYLAS グループ

〒444-8585 愛知県岡崎市明大寺町字西郷中 38 番地

自然科学研究機構 分子科学研究所

計算分子科学研究拠点(TCCI)事務局 気付

電子メール：MODYLAS[at]yfep2.ims.ac.jp

FAX: 0564-54-2254

電話：0564-55-7074

別紙1 著作権者一覧

著作者名 所属機関・職名／肩書き、または著作者住所（機関に所属しない場合）

*岡崎 進 国立大学法人 名古屋大学大学院工学研究科 教授
吉井 範行 国立大学法人 名古屋大学大学院工学研究科 特任准教授
水谷 文保 大学共同利用機関法人 自然科学研究機構 分子科学研究所 技術職員
岩橋 建輔 大学共同利用機関法人 自然科学研究機構 分子科学研究所 技術職員
山田 篤志 国立大学法人 名古屋大学大学院工学研究科 助教
安藤 嘉倫 国立大学法人 名古屋大学大学院工学研究科 研究員
藤本 和士 学校法人立命館 立命館大学薬学部 助教
小嶋 秀和 国立大学法人 名古屋大学大学院工学研究科 院生

*:MODYLAS 著作権管理者

3. 動作環境

3.1 マシン環境

- 下記の環境で動作確認を行っています。
京コンピューター, FX10, PC クラスタ
- コンパイラ
MPI でコンパイルするためのコマンドはシステムごとに異なりますので、システムで用意されているコマンドを確認してください。
 - [富士通] frtpr (mpifrtpr / mpifrt 等)
 - [インテル] ifort (mpif90/mpiifort 等)
 - [PGI] pgf90 (mpif90 等)
- MPI ライブラリ
通常は、MPI 用のコンパイルコマンドでリンクすれば自動で最適な MPI ライブラリをリンクします。

3.2 コンパイル方法

(A) modylas、modylas-text2bin、modylas-mdtrj2xyz を一括コンパイルする場合

```
tar xvzf modylas-1.0.0.tar.gz
cd modylas-1.0.0/source/
setenv FC MPIF90CMD
```

以下の `configure` コマンドで設定された `FC` の値が正しくない場合のみ行います。

MPIF90CMD には MPI 用 Fortran90 コンパイラを指定します。

```
setenv FCFLAGS FLAGS
```

以下の `configure` コマンドで設定された `FCFLAGS` の値が正しくない場合のみ行います。

FLAGS には最適化のレベルを指定します。

```
./configure --with-kind-fortran-compiler=XXX
```

XXX にはあらかじめ用意されたシステムのもの指定します。一致するものがない場合は最も近そうなシステムを指定します。あらかじめ用意されているシステムは、

K	京コンピューター用
FX10	富士通 FX10 用
INTEL	インテルコンパイラを使用するシステム用
PGI	PGI コンパイラを使用するシステム用

です。その後、必要に応じて上記の環境変数を設定して再度 `configure` を行うか `src/Makefile` を直接編集します。

```
cd src
```

```
make -j N
```

N は同時にコンパイルする数です。この `-j` オプションを使わない場合は一つずつコンパイルされます。

(B) modylas-text2bin、modylas-mdtrj2xyz のみを MPI を使わずにコンパイルする場合


```
tar xvzf modylas-1.0.0.tar.gz
cd modylas-1.0.0/source/
setenv FC F90CMD
```

以下の `configure` コマンドで設定された `FC` の値が正しくない場合のみ行います。

F90CMD には Fortran90 コンパイラーを指定します。

```
setenv FCFLAGS FLAGS
```

以下の `configure` コマンドで設定された `FCFLAGS` の値が正しくない場合のみ行います。
最適化のレベルなどの設定を行います。

```
./configure --with-kind-fortran-compiler=XXX --disable-mpi
```

XXX にはあらかじめ用意されたシステムのを指定します。一致するものがない場合は最も近そうなシステムを指定します。あらかじめ用意されているシステムは、

K	京コンピューター用
FX10	富士通 FX10 用
INTEL	インテルコンパイラーを使用するシステム用
PGI	PGI コンパイラーを使用するシステム用

です。その後、必要に応じて上記の環境変数を設定して再度 `configure` を行うか `src/Makefile` を直接編集します。

```
cd src
make clean
make -j N
```

N は同時にコンパイルする数です。この `-j` オプションを使わない場合は一つずつコンパイルされます。

4. 入出力ファイル

MODYLAS の入出力ファイル名は、拡張子以外の部分はすべて共通でなければならない。以下この部分を”sessionname”と表記する。sessionname は任意の文字列でかまわないが、一文字目には数字を使用することはできない。各入出力ファイルは拡張子により区別し、その一覧を次の節で示す。

4.1 入出力ファイル一覧

【入力ファイル】

入力ファイルは、"sessionname.mddef" に計算条件を記述する。"sessionname.mdff"には力場情報を、"sessionname.mdxyz"には原子座標および速度の初期値を記述する。力場と初期座標・速度の入力ファイルには、アスキー形式とバイナリ形式のどちらでも使うことができる。通常はアスキー形式の入力ファイルを推奨するが、巨大系の計算には、入力ファイルの読み込み時間を短縮するためにバイナリ形式の入力ファイルを使用することを推奨する。アスキーとバイナリ形式の両方が存在する場合には、バイナリ形式が優先される。

- sessionname.mddef
計算条件 (アスキー形式)
- sessionname.mdff
力場情報 (アスキー形式)
- sessionname.mdff.bin
sessionname.mdff のバイナリ形式版 (バイナリ形式)
- sessionname.mdxyz
構成原子の座標と速度および周期セル情報 (アスキー形式)
- sessionname.mdxyz.bin
sessionname.mdxyz のバイナリ形式版 (バイナリ形式)
- sessionname.posiconst (オプション)
原子の拘束(position constrain)の外部入力ファイルによる指定 (アスキー形式)。原子拘束は力場入力ファイル(sessionname.mdff)で指定できるが、この sessionname.posiconst が存在するときはこちらを優先し、力場ファイル内の指定は無効になる。巨大系の場合には読み込み速度の都合上、こちらの外部入力ファイルを用いることを推奨する。

(※) バイナリ形式のファイルは全て同じ endian 形式とする。京コンピュータで little-endian 形式の入力ファイルを使って実行する際には環境変数 FORT90L='-Wl,-T'を設定する必要がある。

【出力ファイル】

- sessionname.mdrun
計算に要した経過時間情報 (アスキー形式)
- sessionname.mdmntr
時間ステップごとの物理量(ハミルトニアン,温度,圧力など) (アスキー形式)
- sessionname.mdtrj.bin
原子座標と,速度の軌跡 (バイナリ形式)

- `sessionname.restart.bin`
リスタート用の座標・速度情報（バイナリ形式）。`sessionname.mdxyz.bin` の形式と同じ
- `sessionname.restart.asc`
計算の最終構造の座標と速度が、リスタート用の入力ファイル `sessionname.mdxyz` の形式で出力されたもの（アスキー形式）。デフォルトで出力される設定だが、`sessionname.mddef`において `<output>` タグ内の変数 `ascii=no` と指定した場合には出力されない。

5. 入力ファイルの作成

5.1 入力ファイルのフォーマット概要

入力ファイル内でのキーワードやパラメータの設定はタグ形式で記述する。各タグは `<tag>` で開き `</tag>` で閉じる。`<…>` 内の文字列” `tag`” でタグ種の区別をする。ここで” `tag`” には格納されたデータの意味にそった名前が付けられている。タグは階層構造になっており、親タグ子タグの関係が決まっているため、この階層の順序は変えてはならない。タグの中では、値が 1 つのみの場合は「変数名=値」の形式で与え、座標などの複数の値データがある場合にはタグの中に値をスペースや改行により区切った形式で記述する。使用できる変数名はその親タグによって決まっている。値には整数型、実数型、文字型の区別がある。空白と改行は入っていてもよいが、タブは許容されない。各入力ファイルの簡単な例は、サンプル入力ファイルにある（Tutorial 参照）。入力キーワードとオプションの一覧は本 Manual の付録にすべて記載してある。

5.2 入力ファイルの作成方法

MODYLAS の入力ファイルの作成を支援するソフトウェアを以下に示す

- 入力支援ソフト Nano-Ignition(フリーソフトウェア、web site: <http://nano-ignition.ims.ac.jp/>)。このソフトウェアを用いることにより、PDB ファイル形式等の原子座標の読み込み後に溶媒の付加や力場の設定などを行い最終的に MODYLAS の入力ファイルを出力することができる。
- VMD ソフトウェア(web site: <http://www.ks.uiuc.edu/Research/vmd/>)に MODYLAS 用の機能拡張用プログラム `plugin`(近日配布予定)を適用したもの。これにより VMD に取り込んだあるいはモデリングした分子系の情報を MODYLAS の入力ファイル形式に出力できる。
- MODYLAS の計算条件入力ファイル(`sessionname.mddef`)を近日配布予定のソフトウェアを用いて作成できる。

5.3 入力ファイルのバイナリー化について

アスキー形式の入力ファイルの `sessionname.mdff` および `sessionname.mdxyz`, からバイナリ形式の入力ファイル `sessionname.mdff.bin` および `sessionname.mdxyz.bin` へは、`source/src/` 以下にある変換プログラム `modylas-text2bin` を用いる。

```
%> (パス)/modylas-text2bin sessionname
```

(注) バイナリ形式の入力ファイルでは、原子位置の拘束キーワードは無効である（力場入力ファイル

内の<position constrain>タグ内のキーワード)。バイナリ形式の入力ファイルを用いて位置拘束を行いたい場合は、外部入力ファイル sessionname.posiconst により指定する。

6. 実行方法

6.1 実行方法

MODYLAS の実行の手順を以下に示す：

- (1) 入力ファイルを用意する(前章を参照)
- (2) コンパイルにより生成された実行モジュール”modylas”を入力ファイル一式の存在するディレクトリにコピーあるいはリンクを張る。
- (3) マシン環境のシステムに応じたジョブ投入スクリプトを作成し、ジョブ投入コマンドによりジョブを投入する（使用するマシン環境のマニュアルを参照）。

例として京コンピュータの場合のジョブ投入方法を示す。ジョブ投入スクリプト（後述）を作成し、次のコマンドでジョブを投入する

```
%> pjsub ./run_K.sh
```

“pjsub”が京コンピュータにジョブを投入するコマンドであり、“run_K.sh”はユーザーが作成するスクリプトであり、ファイル名やノード数などの並列化環境を指定する。

ジョブの状態の確認は、

```
%> pjstat
```

ジョブの削除は

```
%> pjdel (job の ID 番号 : pjstat で確認)
```

で行う。

バッチジョブ形式でなく、直接コマンドラインから計算を実行する場合には

```
%> mpiexec -n 8 (パス)/modylas sessionname
```

により計算を実行する。

【ジョブ投入スクリプトの記入法(京コンピュータの場合)】

ジョブスクリプト(“run_K.sh”)を次ページに示す。このスクリプトはサンプルの入力ファイルの中にもあるので、それをコピーして修正して使うと良い。オプションの詳細は京コンピュータ配布のマニュアルを参考して頂くとして、ここではユーザーが普段修正する箇所と知っておくべき所(次ページの①～⑧)のみを簡単に解説する。

- ① 並列ノード数の数。現在のバージョンでは2のべき乗
- ② 計算時間の上限値
- ③ ノード数に応じたジョブのタイプ：“small”, “large”, “huge”, “huge2”など
- ④ stage-in。京コンピュータ本体に入力ファイルを upload するファイル名とパス

(入力ファイルがバイナリー形式の場合は".bin"をつけてファイル名を変更する)

- ⑤stage-out。京コンピュータ本体から出力ファイルを download するファイル名とパス
- ⑥言語環境。現在の MODYLAS では言語環境の version は 15 に固定
- ⑦並列ノード数。
- ⑧MODYLAS の実行モジュールと入力ファイルの指定。実行モジュール modylas の後に sessionname を指定する。

```
#!/bin/sh
#PJM --rsc-list "node=16"           ←①
#PJM --rsc-list "elapse=0:10:00"   ←②
#PJM --rsc-list="rscgrp=small"     ←③
#PJM --stg-transfiles all
#PJM --mpi "use-rankdir"
#PJM --stgin "rank=* ./modylas      %r:/" ←④
#PJM --stgin "rank=* ./sessionname.mddef %r:/"
#PJM --stgin "rank=* ./sessionname.mdff %r:/"
#PJM --stgin "rank=* ./sessionname.mdxyz %r:/"
##
#PJM --stgout "rank=0 %r:./sessionname.mdmntr ./ " ←⑤
#PJM --stgout "rank=0 %r:./sessionname.mdrun ./ "
#PJM --stgout "rank=0 %r:./sessionname.restart.bin ./ "
#PJM --stgout "rank=0 %r:./sessionname.restart.asc ./ "
#PJM --stgout "rank=0 %r:./sessionname.mdtrj.bin ./ "
#PJM -s
#-----
# path
./work/system/Env_base_1.2.0-15 ←⑥

# Parameters
NPROCS=16 ←⑦
NTHREADS=8
export PARALLEL=${NTHREADS}
export OMP_NUM_THREADS=${NTHREADS}
LPG="/opt/FJSVxosmmm/sbin/lpgparm -t 4MB -s 4MB -h 4MB -d 4MB -p 4MB"

# Endian
export FORT90L='Wl,-T'

# MCA Parameters
MCA_PARAM="--mca common_tofu_fastmode_threshold 0"
MCA_PARAM="${MCA_PARAM} --mca common_tofu_max_fastmode_procs 40 --mca
common_tofu_large_recv_buf_size 2097152"
#-----
pwd

date
LD="./modylas ./sessionname " ←⑧
mpiexec -n ${NPROCS} ${MCA_PARAM} ${LPG} ${LD}
date
```

6.2 再スタートの方法

modylas の計算実行の終了後に、その続きの計算を行いたい場合、4.1 節で述べた入力ファイルを一式準備する。例えば、"sessionname1" というセッション名の計算における入出力ファイルから "sessionname2" というセッション名の入力ファイルを生成して続きを計算する時、sessionname2 の入力ファイル一式は次のように準備する：

- 計算条件ファイル：

前の計算から入力ファイルをコピーする

```
%> cp sessionname1.mddef sessionname2.mddef
```

必要あれば再スタート用ファイルの計算条件を修正する。

- 力場ファイル：

[アスキー形式] %> cp sessionname1.mdff sessionname2.mdff

[バイナリ形式] %> cp sessionname1.mdff.bin sessionname2.mdff.bin

- 座標・速度ファイル：

[アスキー形式] %> cp sessionname1.restart.asc sessionname2.mdxyz

[バイナリ形式] %> cp sessionname1.restart.bin sessionname2.mdxyz.bin

あるいはコピーではなくリンクでもよい(cp コマンドを”ln -s “で置き換える)

- オプションファイル

```
%> cp sessionname1.posiconst sessionname2.posiconst
```

再スタートの実行においては、前節と同様、入力ファイル名を sessionname2 に書きかえた上で計算ジョブの投稿あるいは計算実行コマンドの実行を行う。

7. 計算結果のモニター

7.1 物理量各種の出力結果

ハミルトニアン、温度、圧力などの物理量は、sessionname.mdmntr に sessionname.mddef 内の <monitor> タグ内の変数 interval で指定されたステップ間隔で逐次書き出される。

【sessionname.mdmntr の出力例】

```
## n0101.mdmntr -- monitor variables output from MD calculation by modylas
#
# datas below are formatted as:
# step      time          Hamiltonian      potential-E      kinetic-E      total energy
temperature  volume          pressure        box-length(x)   box-length(y)
box-length(z)
#
#           [sec]          [J/cell]        [J/cell]        [J/cell]        [J/cell]
[K]           [m3]          [Pa]           [m]             [m]
[m]
#
      1  2.000000000000E-15 -1.080596504280E-16 -2.464759127634E-16  1.305735314800E-16
-1.159023812835E-16  3.260776624164E+02  2.491621263963E-25  6.596619488522E+06
5.759307554640E-09  5.490470367063E-09  8.586602898096E-09
      2  4.000000000000E-15 -1.080970356668E-16 -2.469468756070E-16  1.310090310261E-16
-1.159378445810E-16  3.271652233668E+02  2.491678749128E-25  5.577704315386E+06
5.759411245174E-09  5.490486488292E-09  8.586695250716E-09
```

sessionname.mdmntr の中身のヘッダー部分 (#付) には出力される物理量名(4 行目)および各物理量の単位(5 行目)が記述されている。以下その説明である：

1. step	ステップ数。 Multi-Time-Step 法の場合は分子間長距離相互作用におけるステップ数	
2. time	経過時間 (=dt*step)	[sec]
3. Hamiltonian	系のハミルトニアン (粒子系+熱浴+圧力浴) (熱浴 : NVT アンサンブルの場合、圧力浴 : NPT アンサンブルの場合)	[J/cell]
4. potential-E	粒子系のポテンシャルエネルギー (分子間+分子内)	[J/cell]
5. kinetic-E	粒子系の運動エネルギー	[J/cell]
6. total energy	粒子系の総エネルギー : potential-E + kinetic-E	[J/cell]
7. temperature	粒子系の温度	[K]
8. volume	基本セル体積	[m ³]
9. pressure	系のバルク圧力 : (Pxx+Pyy+Pzz)/3	[Pa]
10. box-length(x)	基本セルベクトル a の絶対値	[m]
11. box-length(y)	基本セルベクトル b の絶対値	[m]
12. box-length(z)	基本セルベクトル c の絶対値	[m]

6 行目以降に各ステップでの物理量の値が出力されている。これはグラフ表示ソフトを用いることによりグラフ化することができる。

例えば gnuplot により簡単に表示可能である。gnuplot がインストールされている環境で

```
%> gnuplot
```

と入力し起動する。「gnuplot> 」につづく空白に決まった文字列を打つことでグラフが画面に表示される。たとえばハミルトニアンの時間依存性を見たいときには、

```
gnuplot> plot 'sessionname.mdmntr' using 2:3 with lines
```

と打つ。ここで '...' 内はファイル名を、using に続く X:Y は X 列に対して Y 列を表示することを、with lines はデータ間を線でむすぶことを意味する。

【sessionname.mdrun の出力例】

```
## n0101.mdrun -- run time information of MD calculation by modylas
#
step:          12500
CPU time:      6086.753741 [sec]
for MD:        6084.206923 [sec]
time/step:     0.486737 [sec/step]
```

sessionname.mdrun の中身は次のように構成される。

- 1-2 行目 : sessionname などが記されたヘッダー (#付)
- 3 行目以降 :
 - ・経過ステップ数
 - ・MODYLAS の計算全体の経過時間
 - ・うち I/O を除いた MD 部分のみの経過時間

- ・ 1 step あたりの MD 経過時間

sessionname.mdrun ファイルは毎ステップ出力されるため、ジョブを開始してからその時に完了している計算の時間ステップ数、エラー終了した場合の計算が止まった時間ステップ数、等を把握するのに役立つ。また 1 日の秒数 (86,400 sec) を 1 step あたりの経過時間で割ると、1 日あたりに進むステップ数が概算できる。

7.2 軌跡データの座標ファイルへの変換

sessionname.mdtrj.bin ファイルは研究を進める上で最も重要な出力データである。MD 計算の結果全ての原子についてその軌跡 (trajectory = 座標および速度) が時間の関数として求まる。sessionname.mdtrj.bin はバイナリ形式のファイルのためテキストエディタなどでそのまま読めない。MODYLAS とは別に sessionname.mdtrj.bin を xyz 形式のファイルに変換する補助プログラム modylas-mdtrj2xyz を用意している。使い方は、まずインプット一式および書き出された.mdtrj.bin ファイルがあるフォルダに移動し、次のように打つ。

```
%> ./modylas-mdtrj2xyz sessionname
How many atoms to display?
(if negative value is given, the all atoms are used)
26135                (書き出し対象の原子数)
How many interval steps to display?
1                    (書き出す間隔 (sessionname.mddef での interval とは別) )
```

正常に実行されると現在のフォルダに次のような書式の sessionname.xyz ファイルが生成される：

	26135			
#	12500			
N	0.1687303373E+02	0.2334932507E+01	0.1559506824E+02	←書き出し原子数
C	0.1762886632E+02	0.3602581521E+01	0.1560530104E+02	←ステップ数
...				←原子名、x, y, z
O	0.1328233085E+02	-0.1553688180E+02	-0.1786405457E+02	
H	0.1416506470E+02	-0.1588953921E+02	-0.1775160214E+02	
H	0.1339235805E+02	-0.1480280702E+02	-0.1846842207E+02	

sessionname.mddef 内で指定した分だけ系の座標が時系列に沿ってつながって出力されている。sessionname.xyz ファイルは解析対象とする他、VMD や rasmol などの分子の可視化ソフトウェアを用いて 3 次元画像として見ることもできる。

8. MODYLAS の並列化

8.1 並列化実装

MODYLAS はつぎの 3 階層の並列化を行なっている。

- ・ MPI (ただし本バージョン 1.0.0 ではプロセス数 NPROCS は 2^m に限る)
- ・ OpenMP
- ・ SIMD (富士通コンパイラに最適化)

基本的に、MPI はノード間、OpenMP はノード内コア間、SIMD はコア内ベクトル演算器間、での並

列を前提としている。MPI プロセスをノード内に 2 つ立て、スレッド並列数を 1/2 とすることも可能である。

MODYLAS の MPI 並列は、3 次元のトーラスネットワークにおいて通信時間が最短となるように最適化されている。3 次元トーラスネットワークとは、各ノードが x, y, z 方向に直方体状に配置され、かつノード間通信が x,y,z 方向に直接隣接するノード間でのみ可能な通信ネットワークの形態である (図 8-1)。したがって MODYLAS の MPI 並列性能を最大限に発揮するためには、ユーザーが指定したノードの 3 次元形状と、領域分割されたセルブロック (8.2 節) が割り当てられた各プロセスの 3 次元形状とを一致させる必要がある。たとえばノードを 16x8x8 という 3 次元形状に選択した場合、プロセス形状も `ndivx=16, ndivy=8, ndivz=8` (計 1024 プロセス) と指定すべきである。このようなノード形状指定、プロセス形状指定についての例を付録 B にのせた。

ノード形状とプロセス形状を完全に一致させなくても計算は実行可能であるが、通信時間についての最適化が失われる。ノード形状を指定することに伴う待ち時間増加と計算速度の向上の兼ね合いを考慮したうえで、形状指定するかどうかをユーザーがその時々判断しなければならない。

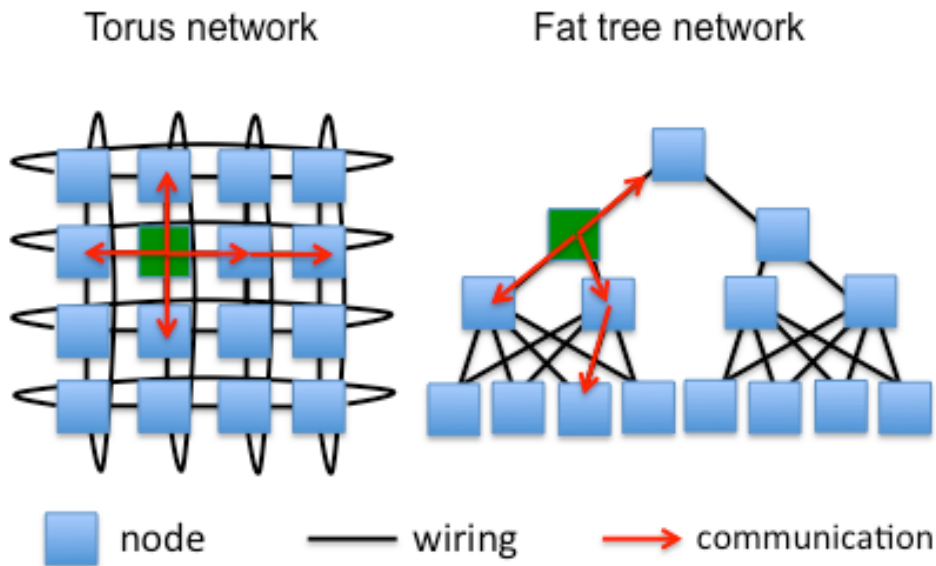


図 8-1 トーラスネットワークおよびファットツリーネットワーク

OpenMP を用いたスレッド並列に関しては、プログラム上の計算量の多い do ループ部分 (ホットスポット) に明示的なディレクティブが挿入されている。それ以外の do ループについては、コンパイラの自動並列化機能によるスレッド並列化がなされるように Makefile においてコンパイラオプションが指定されている。ノード内コア数に応じてスレッド数は 1, 2, 4, 8 および 16 に設定可能である (16 より多い場合については動作未チェック)。ディレクティブ挿入による並列部分については環境変数 `OMP_NUM_THREADS` によって、自動並列化部分については環境変数 `PARALLEL` によって指定する。これら 2 つの環境変数値は特別な理由が無い限り必ず一致させておく。

SIMD については、ホットスポットにおいてコンパイラによる自動 SIMD 並列化で高い性能が発揮できるよう、if 文の除去、ループ長の伸張といったコーディングがなされている。性能は富士通コンパイラ

frtpx に最適化されている。

8.2 領域分割

MPI 並列における各プロセスは、基本セルをグリッド状に領域分割することによって割り当てられた領域内の粒子についての演算を分担する。

各プロセスへの領域の割り当ては次の手順により行なう (図 8-2)。まず立方体の基本セルの各辺を入力パラメータ $n_{cell} (= 2^n)$ の数だけ等分割する。これら等分割された小領域を「サブセル」と呼ぶ。サブセル総数は $2^{3 \cdot n}$ である。サブセルの一辺長さは Lennard-Jones 相互作用のカットオフ距離の半分よりも大きく取らなければならない。次に入力パラメータ n_{divx} , n_{divy} , n_{divz} に応じて x, y, z 各辺にプロセスを割り当てる。その結果プロセスあたり $2^{3 \cdot n - m}$ 個のサブセルが等分配される (ここで $n_{divx} \cdot n_{divy} \cdot n_{divz} = 2^m$)。これら分配されたサブセル群を「セルブロック」と呼ぶ。

使用するノードの 3 次元形状とプロセスの 3 次元形状が一致している場合、MPI 並列化効率向上のために次の点を考慮するとよい。通信負荷均等化の観点からはセルブロックは立方体状であることが望ましい。セルブロックが直方体である場合には、 x 方向を薄く、 z 方向を厚くすると並列性能の向上が見込める。これはプログラム内において $z \rightarrow y \rightarrow x$ の順にデータ累積型通信を行なうためである。

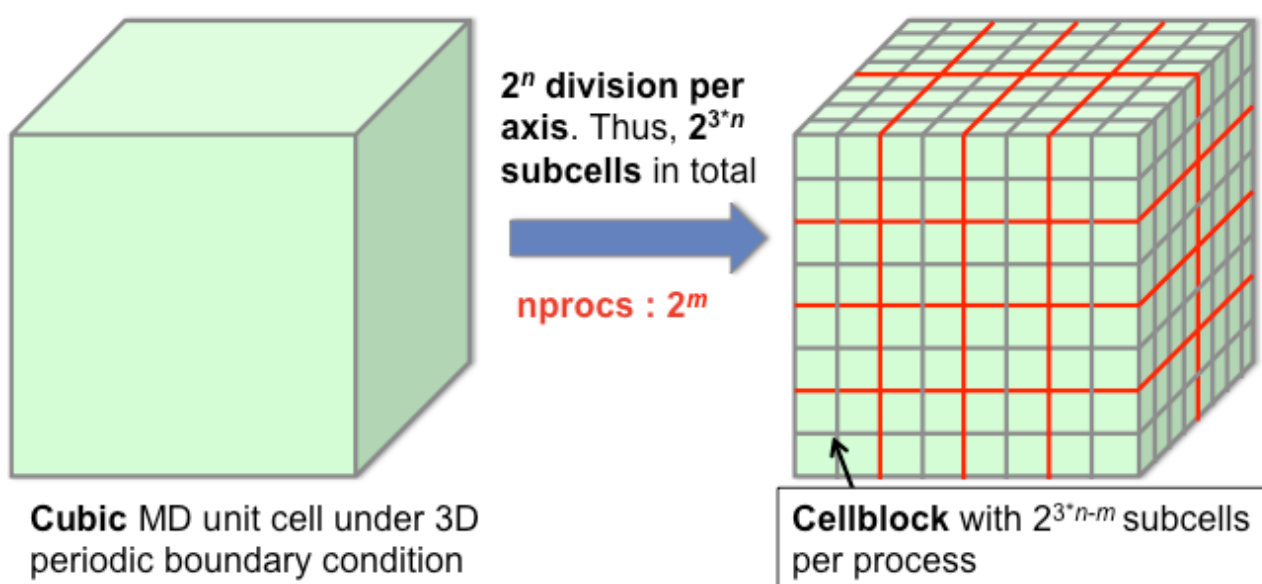


図 8-2 立方体基本セルのサブセルへの分割, および各プロセスへの割り当て。

9. [付録] 入力キーワード・オプション一覧

入力ファイルにおいて、入力値の記述方法は以下に示すタグ形式に従う。1つのパラメータに対して入力値が1つのみの場合は、"key=value"という形式でもよい。複数の入力値がある場合、それらの間に空白や改行が入っていてもよいが、タブの挿入は許されない。原子や分子などの通し番号は 0 からカウントすることに注意。

(例)

```
<atom>
  natom = 3
  <positions>
    0.0  0.0  0.0
    1.0  0.0  0.0
    0.0  1.0  0.0
  </positions>
</atom>
```

上記の例では、<atom>というタグの中で `natom` と `positions` というパラメータの値を設定している。`natom=3` 個あり、それらの `positions` (位置座標) がそれぞれ (0.0 0.0 0.0)、(1.0 0.0 0.0)、(0.0 1.0 0.0) に置かれているということを意味している。

なお、この例で示された `natom` や `positions` といったパラメータが<atom>タグの中にあるということ

を以降では

```
'/atom/natom'
```

```
'/atom/positions'
```

のように表す表記法を用いて説明する。

以下の3節 (9.1, 9.2 および 9.3) では各キーワードを共通の書式で説明する。すなわち、丸括弧内の先頭には変数の型を示す語 "文字列", "整数"ないし"実数"が記され, "×" 記号ののちに, そのパラメータ値の数が記されている。たとえば, (文字列×1)。丸括弧の後ろの文章がキーワードの説明文である。

9.1 計算キーワード(.mddef)

```
/input/version
```

(文字列×1) インプットの書式のバージョンを指定。default 値は "1.0.0"。旧書式のインプットは "0.9.0b" を指定することで利用できる。

```
/output/ascii
```

(文字列×1) リスタート用のファイルは binary 形式、ascii 形式いずれでも出力可能である。指定なしなら ascii 形式が出力され、ascii 形式で書き出さない場合は"no"とする。

```
/output/trajectory/start
```

(整数×1) トラジェクトリーが出力される最初のステップの数。このステップまで出力がスキップされる。

/output/trajectory/interval

(整数×1) トラジェクトリーを書き出す間隔のステップ数。

/output/restart/start

(整数×1) リスタートファイルが最初に出力されるステップ数。通常は 0 を設定する。

/output/restart/interval

(整数×1) リスタートファイルを書き出す間隔 (ステップ数) を指定。このステップごとにリスタートファイルが出力され、上書きされる。

/output/monitor/start

(整数×1) 熱力学量のモニター出力を開始するステップ数。通常は 0 を設定する。

/output/monitor/interval

(整数×1) 熱力学量のモニター出力を書き出す間隔 (ステップ数) を指定。

/integrator/dt

(実数×1) MD 計算の時間刻みの 1 ステップの時間。

<単位> sec

/integrator/steps

(整数×1) MD 計算のステップ数。

/integrator/multiple time step/nstep_skip_middle

(整数×1) マルチタイムステップ法において、中距離相互作用(LJ 相互作用および FMM における直接相互作用)を 1 回計算する間に行う近距離相互作用(分子内の相互作用)の計算回数。default 値は 1。

/integrator/multiple time step/nstep_skip_long

(整数×1) マルチタイムステップ法において、長距離相互作用 (FMM 法における多重極子計算部分) を 1 回計算する間に行う中距離相互作用の計算回数。default 値は 1。

/integrator/optimize/step_length

(実数×1) 構造最適化のステップ幅。default 値は 0.2。

<単位> Å

/integrator/optimize/convergence_condition (本 version では無効)

(文字列×1) 構造最適化(opt キーワード)の収束を判定するための物理量。

“max_force” … 全原子で最も大きな力の大きさ(default)

“mean_force” … 全原子の中で平均の力の大きさ

“d_p_energy” … 1 step あたりのポテンシャルエネルギーの変化

/integrator/optimize/convergence (本 version では無効)

(実数×1) 構造最適化の収束条件。”convergence_condition”のキーワードで指定した物理量の収束閾値。

<単位> “max_force”および”mean_force”では kg m/s²、”d_p_energy”では J

/integrator/optimize/up_rate (本 version では無効)

(実数×1) 構造最適化のステップ幅を増やす割合。default 値は 1.0。
 /integrator/optimize/down_rate (本 version では無効)

(実数×1) 構造最適化のステップ幅を減らす割合。default 値は 1.0。
 /integrator/shake/maxiteration

(整数×1) P-SHAKE の繰り返し計算の最大回数。default 値は 1000
 /integrator/shake/shake_tolerance

(実数×1) P-SHAKE の収束閾値。P-RATTLE の収束値としても使用される。default 値は 10^{-10}
 <単位> 無次元

/ensemble/ensemble

(文字列×1) アンサンブルの種類：
 "opt"(構造最適化：steepest descent 法)
 "nve"(等体積等エネルギー)
 "nvt"(等体積等温)
 "npt_a"(非斜方セルの等圧等温)
 "npt_pr"(Parrinello-Rahman 法による斜方セルの等圧等温) (本 version では無効)
 のいずれかを設定する

/ensemble/temperature

(実数×1) 等温アンサンブル (nvt, npt_a) や速度スケーリングにおける設定温度。
 <単位> K

/ensemble/maxwell_velocities

(文字列×1) 初期速度を maxwell boltzmann 分布で与える場合、"yes"を設定する。

/ensemble /velocity_scaling

(文字列×1) 速度スケーリングを行う場合は、"yes"を設定する。

/ensemble/thermostat/tau_Q

(実数×1) Nose-Hoover チェイン法における温度制御のパラメータ τ_Q の値。
 アンサンブルが "nvt", "npt_a", "npt_pr (現時点で利用不可)" のときに必要。
 <単位> sec

/ensemble/thermostat/initialize

(文字列×1) 計算開始時に熱浴の位置および運動量の変数を 0 にリセットする場合、"yes"とする。

/ensemble/pressure

(実数×1) 等圧アンサンブル(npt_a)の設定圧力。
 <単位> Pa

/ensemble/Pref_inner

(実数×1) 近距離相互作用(分子内の相互作用)由来の部分圧力の設定値。
 <単位> Pa

/ensemble/Pref_outer

(実数×1) 中距離相互作用(LJ 相互作用および FMM 法における直接相互作用)由来の部分圧力の設定値。

<単位> Pa

/ensemble/Pref_outermost

(実数×1) 長距離相互作用 (FMM 法における多重極子計算部分) 由来の部分圧力の設定値。

<単位> Pa

/ensemble/barostat/tau_Q

(実数×1) Andersen 法における圧力制御のパラメーター τ_Q の値。

アンサンブルが"npt_a", "npt_pr (本 version で利用不可) "のときに必要。

<単位> sec

/ensemble/barostat/tau_W

(実数×1) 圧力制御のパラメーター τ_W の値。

アンサンブルが"npt_a", "npt_pr (本 version で利用不可) "のときに必要。

<単位> sec

/ensemble/barostat/initialize

(文字列×1) 計算開始時に熱浴の位置および運動量の変数を0にリセットする場合、"yes"とする。

/intermolecular interaction/twobody/cutoff

(実数×1) Lennard-Jones 相互作用のカットオフ距離。

<単位> Å

/intermolecular interaction/twobody/LJcorrection

(文字列×1) LJ 補正項を含める("yes")か、含めないか("no") のフラグ。

/intermolecular interaction/type

(文字列×1) 周期境界条件の種類。"fmm" (FMM 法)を指定する。

/intermolecular interaction/ncell

(整数×1) FMM における計算系の xyz それぞれの方向の領域の分割数。default 値は 32

/intermolecular interaction/fmm/ULswitch

(整数×1) FMM において多極子情報の通信方法の切り替えスイッチ。

最適値は ncell=8, 16, 32, 64 に対して, ULswitch=0, 1, 2, 3.

/intermolecular interaction/fmm/sterm

(文字列×1) Ewald surface term の寄与を含める("yes")か、含めないか("no")。default は"no"。

/intermolecular interaction/fmm/nmax

(整数×1) 多極子の展開次数。default 値は 4

/mpi/division

(文字列×1) xyz 方向それぞれに割り当てるプロセス数の設定を自動で行う場合は"auto"を設定し、これを手動で行う場合は "manual"を設定して次の"nxdiv"、"nydiv"、"nzdiv"の値を入力する。

/mpi/nxdiv

(整数×1) x方向に割り当てるプロセス数。2の乗数でなければならない。"nxdiv"×"nydiv"×"nzdiv"をプロセス数と等しくする。

/mpi/nydiv

(整数×1) y方向に割り当てるプロセス数。2の乗数でなければならない。"nxdiv"×"nydiv"×"nzdiv"を

プロセス数と等しくする。

`/mpi/nzdiv`

(整数×1) z 方向に割り当てるプロセス数。2 の乗数でなければならない。“nxdiv”×“nydiv”×“nzdiv”をプロセス数と等しくする。

9.2 力場(.mdff)

/forcefield/type

(文字列×1) 力場の種類。入力可能な文字列は, "charmm", "oplsaa", "amber" の 3 種類。

(※)ただし本 version(1.0.0)は charmm のみサポート。

/forcefield/cmap version

(文字列×1) 力場に CHARMM を選択した場合の CMAP の version の指定。”22”または”36”。

default 値は 36

/system

([整数 1, 整数 2] × [分子の種類の数])

「整数 1」は分子の種類を表す ID 番号。0 から割り振る。

「整数 2」はその種類の分子の数。

/position constrain/type

(文字列×1) 位置拘束の種類。入力可能な文字列は以下の 3 種類。

"harmonic"(原子位置を絶対座標に調和振動バネで拘束。

すべてのアンサンブルおよび構造最適化で使用可能)。

"huge_mass"(質量を巨大にする拘束。SHAKE との併用可能。

NVE および NVT アンサンブルで使用可能。

以下の'/position constrain/atom/molecule'で指定した原子と同じ

ID 番号を持つ原子はすべて拘束される)。

"fix"(構造最適化でのみ使用可能。原子の座標を入力値に固定)

/position constrain/force_constant

(実数×1) 位置拘束が harmonic のときのバネ定数。

<単位> kcal/(mol Å²)

/position constrain/atom/atom_type

(文字列×1) '/position constrain/atom/molecules'を指定したとき、

拘束条件をかける原子の種類を指定。

"all"(すべての種類の原子)か"heavy_atoms"(水素以外の重原子のみ)が設定可能。

/position constrain/atom/molecules

(文字列×1) 拘束対象の分子の指定で、"all"か"specified"を設定。

"specified"の場合は、下記の'/position constrain/atom/molecule'および

'/position constrain/atom/nmolecule'の指定が必要。

/position constrain/atom/nmolecule

(整数×1) 位置拘束する分子の数。

/position constrain/atom/molecule

([整数 1, 整数 2] × [位置拘束する分子の数])

「整数 1」には分子種の ID 番号。「整数 2」には分子の(分子種内での)通し番号。

/position constrain/position/type

(文字列×1) 拘束する座標を設定

"initial"(初期座標のまま)か"specify"(指定座標)を設定できる。

"specified"を設定した場合は、以下の'/position constrain/position/coordinate'
を設定しなければならない。

/position constrain/position/coordinate

([整数,実数 1,実数 2,実数 3]×[拘束する原子の数])

「整数」には指定座標を設定する原子の番号。

「実数 1」,「実数 2」,「実数 3」にはそれぞれ指定座標の x,y,z。

<単位> 実数 1 : Å、実数 2 : Å、実数 3 : Å

/position constrain/atom/natom_allow_unconstrain

(整数×1) 拘束される分子に属しているが位置拘束されない原子の数。

位置拘束の設定よりも優先される。

/position constrain/atom/allow atom unconstrain

(整数×拘束される分子に属しているが位置拘束されない原子の数)

拘束される分子に属しているが位置拘束されない原子の全系における通し番号。

/topology and parameters/nspecies

(整数×1) 系に含まれる分子の種類の数。分子の総数ではありません。

/topology and parameters/species

分子種ごとにその詳細を指定するタグ。このタグは分子の種類の数だけ並列に記述する。

/topology and parameters/species/id

(整数×1) 分子の種類 ID 番号。

/topology and parameters/species/natom

(整数×1) その分子を構成する原子の数。

/topology and parameters/species/segments/nsegment

(整数×1) 分子のセグメント数。

Charmm のトポロジーファイルでグループ分けされている塊を
modylas では便宜上セグメントと呼んでいる。

/topology and parameters/species/segments/segment

segment ごとにその詳細を記述するタグ。このタグはセグメントの数だけ記述する。

/topology and parameters/species/segments/segment/ID

(整数×1) セグメントの ID 番号。0 から割り振る。

/topology and parameters/species/segments/segment/natom

(整数×1) セグメントに含まれる原子の数。

/topology and parameters/species/segment/atom

(整数×セグメントに含まれる原子の数) セグメントに含まれる原子の分子内での番号。
分子内の番号は 0 から始める。

/topology and parameters/species/mass

(実数×その分子の原子数) 分子に所属する原子の原子量を、原子の並び順に列記する。

<単位> g/mol

/topology and parameters/species/charge
 (実数×その分子の原子数) 分子に所属する原子の電荷を、原子の並び順に列記する。
 <単位> クーロン

/topology and parameters/species/nvoidpair_coulomb
 (整数×1) 分子内の coulomb void 対の数。

/topology and parameters/species/coulomb void pair
 ([整数 1, 整数 2]×[分子内の coulomb void 対の数])
 「整数 1」, 「整数 2」は void 対を構成する原子の分子内の番号。

/topology and parameters/species/nspecialpair_coulomb
 (整数×1) 分子内の coulomb special 対の数。

/molecules/molecule/coulomb special pair (※charmm では使用しない)
 ([整数 1, 整数 2, 実数]×[分子内 coulomb special 対の数])
 「整数 1」, 「整数 2」は special 対を構成する原子の分子内の番号。
 「実数」は special 対用の電荷。
 <単位> 実数 : クーロン

/topology and parameters/species/epsilon
 ([実数]×[原子数]) 分子に所属する原子の LJ の ϵ 。
 <単位> kcal/mol

/topology and parameters/species/r
 ([実数]×[原子数]) 分子に所属する原子の LJ の R_0 。
 <単位> Å

/topology and parameters/species/
 (nvoidpair_lj = [整数]) 分子内の LJ void 対の数。

/topology and parameters/species/lj void pair
 ([整数 1, 整数 2]×[LJ void 分子対の数]) LJ void 対を構成する 1 組の原子の分子内番号, [整数 1, 整数 2]を LJ void 対の数だけ記述。

/topology and parameters/species /
 (nspecialpair_lj = [整数]) [整数]に 分子内の LJ special 対の数。

/topology and parameters/species/lj special pair
 ([整数 1, 整数 2, 実数 1, 実数 2]×[LJ special 分子対の数])
 整数 1、整数 2 は special 対を構成する原子の分子内の番号
 実数 1 は special 対用の ϵ_{14} の値
 実数 2 は special 対用の R_{14} の値。
 以上を LJ special 分子対の数だけ繰り返す。
 <単位> 整数 1 : なし、整数 2 : なし、実数 1 : kcal/mol、実数 2 : Å

/topology and parameters/species/shake pair
 ([整数 1, 整数 2, 実数 1]×[拘束数]) 整数 1 と整数 2 の値は、拘束に関与する原子の分子内

の番号です。実数 1 の値は、拘束の距離。以上を拘束数だけ繰り返す。

<単位> 整数 1 : なし、整数 2 : なし、実数 1 : Å

/topology and parameters/species/

(nbond = [整数]) [整数]は分子内の(力場が設定される)bond の数。

/topology and parameters/species/bond

([整数 1, 整数 2, 実数 1, 実数 2] × [bond の数])

整数 1 と整数 2 の値は、bond を構成する原子の分子内の番号

実数 3 の値は、 k_b の値

実数 4 の値は、平衡長の距離(b_0)

以上を bond の数だけ繰り返す

<単位> 整数 1 : なし、整数 2 : なし、実数 1 : (kcal/mol)/(Å²)、実数 2 : Å

/topology and parameters/species/

(nangle = [整数]) [整数]は分子内の(力場が設定される)angle の数。

/topology and parameters/species/angle

([整数 1, 整数 2, 整数 3, 実数 1, 実数 2] × [angle の数]) 整数 1 目と整数 2 と整数 3 の値は、angle

を構成する原子の分子内の番号。1-2-3 のように結合してはいなければならない。実数 1 の値は、 K_θ の値。実数 2 の値は、 θ_0 の値。以上を angle の数だけ繰り返す。

<単位> 整数 1, 整数 2, 整数 3 : なし、実数 1 : (kcal/mol)/(radian²)、実数 2 : degree

/topology and parameters/species/

(nub=[整数]) [整数]は 分子内の ub の数。

/topology and parameters/species/ub

([整数 1, 整数 2, 整数 3, 実数 1, 実数 2] × [ub の数]) 整数 1 と整数 2 と整数 3 番の値は、ub

を構成する原子の分子内の番号。1-2-3 のように結合してはいなければならない。実数 1 の値は、 K_{ub} の値、実数 2 の値は、 s_0 の値。以上を ub の数だけ繰り返す。

<単位>整数 1 : なし、整数 2 : なし、整数 3 : なし、実数 1 : (kcal/mol)/(Å²)、実数 2 : Å

/topology and parameters/species/

(ndihedral=[整数]) [整数]は 分子内の dihedral の数。

/topology and parameters/species/dihedral

([整数 1, 整数 2, 整数 3, 整数 4, 実数 1, 整数 5, 実数 2] × [dihedral の数]) 整数 1 と整数 2 と整数

3 と整数 4 の値は、dihedral を構成する原子の分子内の番号。1-2-3-4 のように結合してはいなければならない。実数 1 の値は、 K_x の値。整数 5 値は、 n の値。実数 2 の値は、 δ の値。以上を dihedral の数だけ繰り返す。

<単位> 整数 1、整数 2、整数 3、整数 4 : なし、

実数 1 : kcal/mol、整数 5 : なし、実数 2 : degree

/topology and parameters/species/

(ncmap=[整数]) [整数]は分子内の cmap の数

/topology and parameters/species/cmap

([整数 1, 整数 2, 整数 3, 整数 4, 整数 5, 整数 6] × [cmap の数])

整数 1 の値は、C の原子の分子内の番号。
整数 2 の値は、N の原子の分子内の番号。
整数 3 の値は、CA の原子の分子内の番号。
整数 4 の値は、C の原子の分子内の番号。
整数 5 の値は、N の原子の分子内の番号。
整数 6 の値は、CMAP の種類の番号。

以上を cmap の数だけ繰り返す。

/topology and parameters/species/nitorsion

(nitorsion=[整数]) [整数]は improper torsion の数。

/topology and parameters/species/itorsion

([整数 1, 整数 2, 整数 3, 実数 1, 実数 2] × [improper torsion の数]) 整数 1 と整数 2 と整数 3 と整数 4 の値は、improper torsion を構成する原子の分子内の番号。1 が中心の原子で、1-2-3 の平面と 2-3-4 の平面の角度に対する力場。実数 1 の値は、 K_{ψ} の値。実数 2 は、 ψ_0 の値。以上を improper torsion の数だけ繰り返す。

<単位> 整数 1、整数 2、整数 3、整数 4 : なし、実数 1 : (kcal/mol)/(radian²)、実数 2 : degree

9.3 初期座標・速度(.mdxyz)

/atom/natom

([整数]×1) 系の全原子数。

/atom/positions

([実数]×[3×原子数]) 原子の座標を x,y,z の順に記述し、原子の数だけ繰り返す。

<単位> Å

/atom/velocities

([実数]×[3×原子数]) 原子の速度を x,y,z の順に記述し、原子の数だけ繰り返す。

<単位> Å/sec

/thermostat/nthermostat

([整数]×1) 等温アンサンブル (nvt, npt_a) における Nose-Hoover チェインのサーモスタットの段数。

/thermostat/positions

([実数]×[1×サーモスタットの数]) サーモスタットの一般化座標の位置座標。サーモスタットの数だけ繰り返す。一般的に初期値は 0.0 でよい。

/thermostat/velocities

([実数]×[1×サーモスタットの数]) サーモスタットの速度。サーモスタットの数だけ繰り返す。一般的に初期値は 0.0 でよい。

/barostat/nbarostat

([整数]×1) 等圧アンサンブル (npt_a) のときのバロスタットの温度制御のためのサーモスタットの段数。

/barostat/positions

([実数]×[1×バロスタットの数]) バロスタット用サーモスタットの位置。サーモスタットの段数だけ繰り返す。一般的に初期値は 0.0 でよい。

/barostat/velocities

([実数]×[1×バロスタットの数]) バロスタット用サーモスタットの速度。サーモスタットの段数だけ繰り返す。一般的に初期値は 0.0 でよい。

/periodic cell/length

([実数]×3) 周期境界条件のときのセルの x, y, z 方向の長さ。(本 version では x=y=z の時のみサポート(立方体セルのみ))

<単位> Å

/periodic cell/angle/

([実数]×3) 斜方セルの α , β , γ の角度。直方セルの場合は、90.0。

<単位> degree

/periodic cell/vboxg

([実数]×[3×3]) 斜方セルの自由度の速度。3x3 の行列なので、9 つの実数を並べる。初期値は全て 0.0 とすればよい。

9.4 原子の位置拘束(.posiconst)

この入力ファイルにはタグ形式は使わず以下の形式を用いる：

1 行目： キーワード：

“no-position-constrain” … 位置拘束なし

“fix”, “huge_mass”, “harmonic” … mdff ファイルの/position constrain/type を参照

ただし、“harmonic”キーワードを用いる場合には、同じ 1 行目に力定数[kcal/mol/Å²]の値も書く必要がある。

(例) harmonic 1.0

2 行目以降(全原子数の行数)： 各行に“0” または“1” を書く。

“0” … 拘束する原子

“1” … 拘束しない原子

ただし、“harmonic”キーワードの場合にのみ、0/1 の後の原子拘束の位置 xyz 座標を書く必要がある。

(例)

0 1.00 -12.123 2.234

0 0.10 2.245 7.999

1 0.30 1.345 9.999

0 0.20 2.345 2.999

1 23.00 4.01 3.000

… (全原子数分の行)

10. 参考文献

[1] "MODYLAS: A Highly Parallelized General-Purpose Molecular Dynamics Simulation Program for Large-Scale Systems with Long-Range Forces Calculated by Fast Multipole Method (FMM) and Highly Scalable Fine-Grained New Parallel Processing Algorithms", *J. Chem. Theo. Comp.*, **9**, 3201-3209 (2013), Yoshimichi Andoh, Noriyuki Yoshii, Kazushi Fujimoto, Keisuke Mizutani, Hidekazu Kojima, Atsushi Yamada, Susumu Okazaki, Kazutomo Kawaguchi, Hidemi Nagao, Kensuke Iwahashi, Fumiyasu Mizutani, Kazuo Minami, Shin-ichi Ichikawa, Hidemi Komatsu, Shigeru Ishizuki, Yasuhiro Takeda, and Masao Fukushima

11. 開発者

- 安藤 嘉倫 (名古屋大学)
- 吉井 範行 (名古屋大学)
- 藤本 和士 (立命館大学)
- 小嶋 秀和 (名古屋大学)
- 山田 篤志 (名古屋大学)
- 岡崎 進 (名古屋大学)
- 岩橋 建輔 (分子科学研究所)
- 水谷 文保 (分子科学研究所)