FDPSユーザーチュートリアル

FDPS開発者

目 次

1 TODO

2　変更記録

- 2015/01/25 作成

3 概要

本節では、Framework for Developing Particle Simulator (FDPS) の概要について述べる。FDPS は粒子シミュレーションのコード開発を支援するフレームワークである。FDPS が行うのは、計算コストの最も大きな粒子間相互作用の計算と、粒子間相互作用の計算のコストを負荷分散するための処理である。これらはマルチプロセス、マルチスレッドで並列に処理することができる。比較的計算コストが小さく、並列処理を必要としない処理 (粒子の軌道計算など) はユーザーが行う。

FDPS が対応している座標系は、２次元直交座標系と３次元直交座標系である。また、境界条件としては、開放境界条件と周期境界条件に対応している。周期境界条件の場合、x、y、z 軸方向の任意の組み合わせの周期境界条件を課すことができる。

ユーザーは粒子間相互作用の形を定義する必要がある。定義できる粒子間相互作用の形には様々なものがある。粒子間相互作用の形を大きく分けると２種類あり、１つは長距離力、もう１つは短距離力である。 この２つの力は、 遠くの複数の粒子からの作用を１つの超粒子からの作用にまとめるか (長距離力)、 まとめないか (短距離力) という基準でもって分類される。

長距離力には、 小分類があり、 無限遠に存在する粒子からの力も計算するカットオフなし長距離力と、 ある距離以上離れた粒子からの力は計算しないカットオフあり長距離力がある。 前者は開境界条件下における重力やクーロン力に対して、 後者は周期境界条件下の重力やクーロン力に使うことができる。 後者のためには Particle Mesh 法などが必要となるが、これは FDPS の拡張機能として用意されている。

短距離力には、小分類が 4 つ存在する。短距離力の場合、粒子はある距離より離れた粒子からの作用は受けない。すなわち必ずカットオフが存在する。 このカットオフ長の決め方によって、 小分類がなされる。 すなわち、 全粒子のカットオフ長が等しいコンスタントカーネル、 カットオフ長が作用を受ける粒子固有の性質で決まるギャザーカーネル、 カットオフ長が作用を与える粒子固有の性質で決まるスキャッタカーネル、 カットオフ長が作用を受ける粒子と作用を与える粒子の両方の性質で決まるシンメトリックカーネルである。 コンスタントカーネルは分子動力学における LJ 力に適用でき、 その他のカーネルは SPH などに適用できる。

ユーザーは、粒子間相互作用や粒子の軌道積分などを、C++言語を用いて記述する。将来的には Fortran 言語でも記述できるように検討する。

4 入門

本節では、FDPS の入門について記述する。FDPS を使用する環境、FDPS に必要なソフトウェア、FDPS のインストール方法、サンプルコードの使用方法、の順で記述する。

4.1 動作環境

FDPS が動作する環境は以下の通りである。

- Linux のターミナル
- Mac OS X のターミナル
- Windows の Cygwin 上のターミナル

4.2 必要なソフトウェア

本節では、FDPS を使用する際に必要となるソフトウェアを記述する。まず標準機能を用いるのに必要なソフトウェア、次に拡張機能を用いるのに必要なソフトウェアを記述する。

4.2.1 標準機能

本節では、FDPS の標準機能のみを使用する際に必要なソフトウェアを記述する。最初に逐次処理機能のみを用いる場合（並列処理機能を用いない場合）に必要なソフトウェアを記述する。次に並列処理機能を用いる場合に必要なソフトウェアを記述する。

4.2.1.1 逐次処理

逐次処理の場合に必要なソフトウェアは以下の通りである。

- make
- C++03 以降のコンパイラ

4.2.1.2 並列処理

本節では、FDPS の並列処理機能を用いる際に必要なソフトウェアを記述する。まず、OpenMP を使用する際に必要なソフトウェア、次に MPI を使用する際に必要なソフトウェア、最後に OpenMP と MPI を同時に使用する際に必要なソフトウェアを記述する。

4.2.1.2.1　OpenMP

OpenMP を使用する際に必要なソフトウェアは以下の通り。

- make
- OpenMP 対応の C++03 以降のコンパイラ

4.2.1.2.2　MPI

MPI を使用する際に必要なソフトウェアは以下の通り。

- make
- MPI 対応の C++03 以降のコンパイラ

4.2.1.2.3　MPI+OpenMP

MPI と OpenMP を同時に使用する際に必要なソフトウェアは以下の通り。

- make
- MPI と OpenMP に対応の C++03 以降のコンパイラ

4.2.2　拡張機能

本節では、FDPS の拡張機能を使用する際に必要なソフトウェアについて述べる。FDPS の拡張機能には Particle Mesh がある。以下では Particle Mesh を使用する際に必要なソフトウェアを述べる。

4.2.2.1　Particle Mesh

Particle Mesh を使用する際に必要なソフトウェアは以下の通りである。

- make
- MPI と OpenMP 対応の C++03 以降のコンパイラ
- FFTW 3.3 以降

4.3　インストール

本節では、FDPS のインストールについて述べる。取得方法、ビルド方法、サンプルコードについて述べる。

4.3.1 取得方法

以下の方法のいずれかで FDPS の圧縮ファイル fdps-20xx-xx-xx.tar.gz を取得できる。

- ブラウザから
  - ウェブサイト https://github.com/FDPS/FDPS からダウンロード
  - FDPS を展開したいディレクトリに移動し、コマンドライン上で以下を実行すると、圧縮ファイルを展開できる。

```
$ tar zxvf fdps-20xx-xx-xx.tar.gz
```

- コマンドラインからの場合は以下のコマンドを実行するとディレクトリ trunk の下に FDPS ができる
  - $ svn co –depth empty https://github.com/FDPS/FDPS
  - $ cd FDPS
  - $ svn up trunk
- コマンドラインから git で

4.3.2 ビルド方法

ウェブサイトから取得した場合はディレクトリ FDPS へ移動、コマンドラインから取得した場合はディレクトリ trunk へ移動する。ここで configure をする必要はない。

4.3.3 サンプルコードの使用方法

本節ではサンプルコードの使用方法について記述する。サンプルコードには重力 N 体シミュレーションコードと、SPH シミュレーションコードがある。最初に重力 N 体シミュレーションコード、次に SPH シミュレーションコードの使用について記述する。

4.3.3.1 重力 N 体シミュレーションコード

以下の手順で本コードを使用できる。

- ディレクトリ fdps/sample/nbody に移動
- カレントディレクトリにある Makefile を編集 (後述)
- コマンドライン上で make を実行
- make された実行ファイルを実行 (後述)

Makefile の編集項目は以下の通りである。OpenMP と MPI を使用するかどうかで編集方法が変ることに注意。

- OpenMP も MPI も使用しない場合
  - マクロ CC に C++コンパイラを代入する
- OpenMP のみ使用の場合
  - マクロ CC に OpenMP 対応の C++コンパイラを代入する
  - ”CFLAGS += -DPARTICLE_SIMULATOR_THREAD_PARALLEL -fopenmp” の行のコメントアウトを外す (インテルコンパイラの場合は-fopenmp を外す)
- MPI のみ使用の場合
  - マクロ CC に MPIC++コンパイラを代入する
  - ”CFLAGS += -DPARTICLE_SIMULATOR_MPI_PARALLEL”の行のコメントアウトを外す
- OpenMP と MPI の同時使用の場合
  - マクロ CC に MPI 対応の C++コンパイラを代入する
  - ”CFLAGS += -DPARTICLE_SIMULATOR_MPI_PARALLEL”の行のコメントアウトを外す
  - ”CFLAGS += -DPARTICLE_SIMULATOR_THREAD_PARALLEL -fopenmp” の行のコメントアウトを外す (インテルコンパイラの場合は-fopenmp を外す)

実行ファイルの実行方法は以下の通りである。

- MPI を使用しない場合、コマンドライン上で以下のコマンドを実行する

```
$ ./nbody
```

- MPI を使用する場合、コマンドライン上で以下のコマンドを実行する

```
$ MPIRUN -np NPROC ./nbody
```

  ここで、”MPIRUN”には mpirun や mpiexec などが、”NPROC”には使用する MPI プロセスの数が入る。

  以下の手順で xy 平面に射影した粒子分布の変化を見ることができる。

  - gnuplot を起動する
  - plot ”filename_time.dat” using 1:2

  出力ファイルフォーマットは 1 列目から順に粒子の位置の x, y, z 座標、粒子の x, y, z 軸方向の速度、粒子質量である。

4.3.3.2 SPH シミュレーションコード

以下の手順で本コードを使用できる。

- ディレクトリ fdps/sample/sph に移動
- カレントディレクトリにある Makefile を編集 (後述)
- コマンドライン上で make を実行
- make された実行ファイルを実行 (後述)

Makefile の編集項目は以下の通りである。OpenMP と MPI を使用するかどうかで編集方法が変ることに注意。

- OpenMP も MPI も使用しない場合
  - マクロ CC に C++コンパイラを代入する
- OpenMP のみ使用の場合
  - マクロ CC に OpenMP 対応の C++コンパイラを代入する
  - ”CFLAGS += -DPARTICLE_SIMULATOR_THREAD_PARALLEL -fopenmp” の行のコメントアウトを外す (インテルコンパイラの場合は-fopenmp を外す)
- MPI のみ使用の場合
  - マクロ CC に MPIC++コンパイラを代入する
  - ”CFLAGS += -DPARTICLE_SIMULATOR_MPI_PARALLEL”の行のコメントアウトを外す
- OpenMP と MPI の同時使用の場合
  - マクロ CC に MPI 対応の C++コンパイラを代入する
  - ”CFLAGS += -DPARTICLE_SIMULATOR_MPI_PARALLEL”の行のコメントアウトを外す
  - ”CFLAGS += -DPARTICLE_SIMULATOR_THREAD_PARALLEL -fopenmp” の行のコメントアウトを外す (インテルコンパイラの場合は-fopenmp を外す)

実行ファイルの実行方法は以下の通りである。

- MPI を使用しない場合、コマンドライン上で以下のコマンドを実行する

```
$ ./sph
```

- MPI を使用する場合、コマンドライン上で以下のコマンドを実行する

```
$ MPIRUN -np NPROC ./sph
```

  ここで、”MPIRUN”には mpirun や mpiexec などが、”NPROC”には使用する MPI プロセスの数が入る。

以下の手順で xy 平面に射影した粒子分布の変化を見ることができる。

- gnuplot を起動する
- plot ”filename_time.dat” using 1:2

出力ファイルフォーマットは 1 列目から順に粒子の位置の x, y, z 座標、粒子の x, y, z 軸方向の速度、粒子質量、密度、内部エネルギーである。

5　使ってみよう

5.1　サンプルコードのコンパイルと実行

5.2　前提知識

5.2.1　Vector 型

5.3　(穴埋め式で) 固定長 SPH シミュレーションコードを書く

5.3.1　作業ディレクトリ

5.3.2　インクルード

5.3.3　ユーザー定義クラス

5.3.3.1　概要

5.3.3.2　FullParticle 型

5.3.3.3　(EssentialParticleI 型)

5.3.3.4　(EssentialParticleJ 型)

5.3.3.5　(SuperParticleJ 型)

5.3.3.6　Force 型

5.3.3.7　calcForceEpEp 型

5.3.3.8　(calcForceEpSp 型)

5.3.4　プログラム本体

5.3.4.1　概要

5.3.4.2　全体の初期化

5.3.4.3　インスタンスの生成

5.3.4.4　(領域クラスの初期化)

5.3.4.5　(粒子群クラスの初期化)

5.3.4.6　粒子データの入力

5.3.4.7　(相互作用ツリークラスの初期化)

5.3.4.8　領域分割の実行

5.3.4.9　粒子交換の実行

5.3.4.10　相互作用計算の実行

5.3.4.11　時間積分

6　サンプルコード

6.1　N 体シミュレーション

6.2　SPH シミュレーション

6.3　N 体+SPH シミュレーション

7　よくあるエラーメッセージ

8　ユーザーサポート

9　ライセンス

MIT ライセンスに準ずる。Particle Mesh は GreeM の MIT ライセンスである。