FDPS講習会の手引

谷川衝、岩澤全規、細野七月、似鳥啓吾、村主崇行、牧野淳一郎

平成27年7月8日

目 次

1 プログラム

- 13:00 – 14:00 FDPS の講義
  - 13:00 – 13:05 イントロダクション (牧野淳一郎)
  - 13:05 – 13:20 概要説明 (谷川衝)
  - 13:20 – 13:35 FDPS 詳細 1 – API と内部構造 (岩澤全規)
  - 13:35 – 13:50 FDPS 詳細 2 – サンプルコード解説 (細野七月)
  - 13:50 – 14:00 Q&A
- 14:00 – 15:30 FDPS の実習
  - FDPS のインストール
  - サンプルコードの使用 1 (重力 N 体シミュレーションコード)
  - サンプルコードの使用 2 (SPH シミュレーションコード)
- 15:30 – 17:00 FDPS 使用に関する相談

2 FDPSの実習

2.1 環境の設定

FOCUSの計算機にログインしたら、以下のコマンドを実行する。

```
$ module load gnu/openmpi165
```

2.2 FDPSの入手

- FOCUSの計算機を使用する場合:

  以下のコマンドを実行し、FDPSを自分のカレントディレクトリにコピーする(「$」はコマンドプロンプトであるので、「$」を打ち込む必要はない)。

  ```
  $ cp -r ~uiud0014/FDPS-master .
  ```

  以上により、ディレクトリFDPS-masterがコピーされる。

- FOCUSの計算機以外を使用する場合:

  https://github.com/FDPS/FDPS からFDPSの最新版をダウンロードし、好きなディレクトリ下で解凍する。これによってディレクトリFDPS-masterが出来る。

以下ではディレクトリFDPS-masterがあるディレクトリの名前をfdpsとする。

2.3 サンプルコードの使用

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。ここでの使用方法の説明はFOCUSの計算機を使用する場合に限る。

2.3.1 重力N体シミュレーションコード

2.3.1.1 概要

以下の手順で本コードを使用できる。

- ディレクトリfdps/FDPS-master/sample/nbodyに移動
- makeを実行
- ジョブの投入

- 結果の解析
- OpenMP/MPI の利用 (オプション)

2.3.1.2 ディレクトリ移動

ディレクトリ fdps/FDPS-master/sample/nbody に移動する。

2.3.1.3 make の実行

make コマンドを実行する。

2.3.1.4 ジョブの投入

以下のコマンドを実行し、ジョブの投入を行う。

```
$ sbatch nbody1.sh
```

正しくジョブが投入されている場合、squeue コマンドを実行すると、例えば以下のようにログが出力される。

```
 JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
189090     c006m    nbody uiud0014  R       0:00      1 c017
```

ジョブのキャンセルは以下のコマンドで実行できる。

```
$ scancel JOBID
```

JOBID は上の squeue コマンドで表示された JOBIJ の番号である (上で言えば、189090)。

正しくジョブが終了すると、ファイル stdout.log の最後には以下のようなログが出力されている。energy error は絶対値で 1×10^{-3} のオーダーに収まっていればよい。

```
time:  9.5000000 energy error: -3.804653e-03
time:  9.6250000 energy error: -3.971175e-03
time:  9.7500000 energy error: -3.822343e-03
time:  9.8750000 energy error: -3.884310e-03
******** FDPS has successfully finished. ********
```

2.3.1.5 結果の解析

ディレクトリ result に粒子分布を出力したファイル”000x.dat”ができている。x は 0 から 9 の値で、時刻を表す。出力ファイルフォーマットは 1 列目から順に粒子の ID, 粒子の質量、位置の x, y, z 座標、粒子の x, y, z 軸方向の速度である。

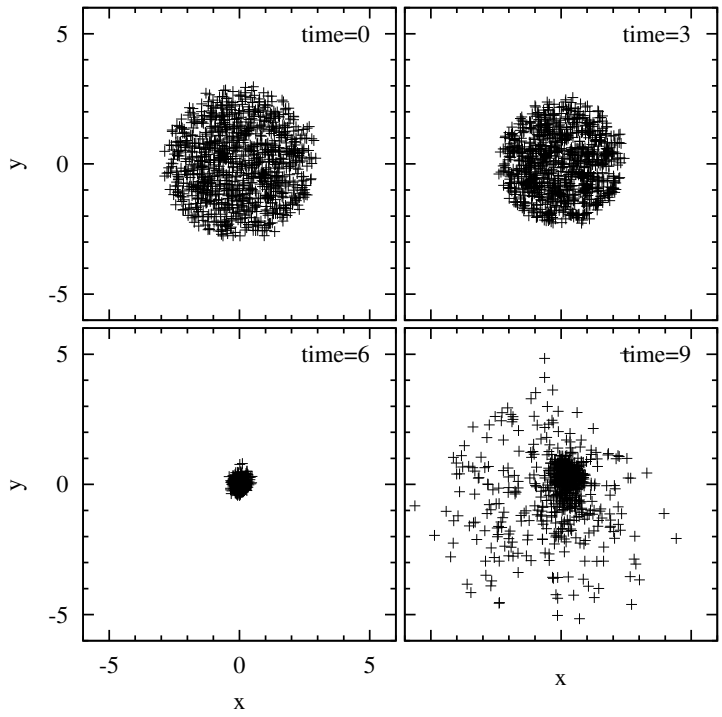

図 1:

ここで実行したのは、粒子数 1024 個からなる一様球 (半径 3) のコールドコラプスである。コマンドライン上で以下のコマンドを実行すれば、時刻 9 における xy 平面に射影した粒子分布を見ることができる。

```
$ gnuplot
$ plot "result/0009.dat" using 3:4
```

他の時刻の粒子分布をプロットすると、一様球が次第に収縮し、その後もう一度膨張する様子を見ることができる (図 1 参照)。

2.3.1.6 OpenMP/MPI の利用

OpenMP や MPI を利用する場合について以下に記述する。

- OpenMP のみ使用の場合
  - Makefile の編集
    - 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
  - make コマンドを実行する。

- 「sbatch nbody2.sh」というコマンドを実行し、ジョブを投入する。
- ファイル stdout.log 内でスレッド数が 4 であることを確認できる。

- MPI のみ使用の場合
  - Makefile の編集
    - 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
  - make コマンドを実行する。
  - 「sbatch nbody3.sh」というコマンドを実行し、ジョブを投入する。
  - ファイル stdout.log 内でプロセス数が 2 であることを確認できる。

- OpenMP と MPI の同時使用の場合
  - Makefile の編集
    - マクロ CC に MPI 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)
    - ”CFLAGS += -DPARTICLE_SIMULATOR_MPI_PARALLEL”の行のコメントアウトを外す
  - make コマンドを実行する。
  - 「sbatch nbody4.sh」というコマンドを実行し、ジョブを投入する。
  - ファイル stdout.log 内でプロセス数が 2、スレッド数が 4 であることを確認できる。

2.3.2 SPH シミュレーションコード

2.3.2.1 概要

以下の手順で本コードを使用できる。

- ディレクトリ fdps/FDPS-master/sample/sph に移動
- make を実行
- ジョブの投入
- 結果の解析
- OpenMP/MPI の利用 (オプション)

2.3.2.2 ディレクトリ移動

ディレクトリ fdps/FDPS-master/sample/sph に移動に移動する。

2.3.2.3 make の実行

make コマンドを実行する。

2.3.2.4 ジョブの投入

以下のコマンドを実行し、ジョブの投入を行う。

```
$ sbatch sph1.sh
```

正しくジョブが投入されている場合、squeue コマンドを実行すると、例えば以下のようにログが出力される。

```
 JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
189090     c006m      sph uiud0014  R       0:00      1 c017
```

ジョブのキャンセルは以下のコマンドで実行できる。

```
$ scancel JOBID
```

JOBID は上の squeue コマンドで表示された JOBID 番号である (上で言えば、189090)。

正しくジョブが終了すると、ファイル stdout.log の最後には以下のようなログが出力されている。

```
******** FDPS has successfully finished. ********
```

2.3.2.5 結果の解析

ディレクトリ result にファイルが出力されている。ファイル名は”00xx.dat”(x には数字が入る) となっている。ファイル名は時刻を表す。出力ファイルフォーマットは 1 列目から順に粒子の ID、粒子の質量、位置の x, y, z 座標、粒子の x, y, z 軸方向の速度、密度、内部エネルギー、圧力である。

これは 3 次元の衝撃波管問題である。以下のコマンドを実行すれば、横軸に粒子の x 座標、縦軸に粒子の密度をプロットできる (時刻は 40)。

```
$ gnuplot
$ plot "result/0040.dat" using 3:9
```

正しい答が得られれば、図 2 のような図を描ける。

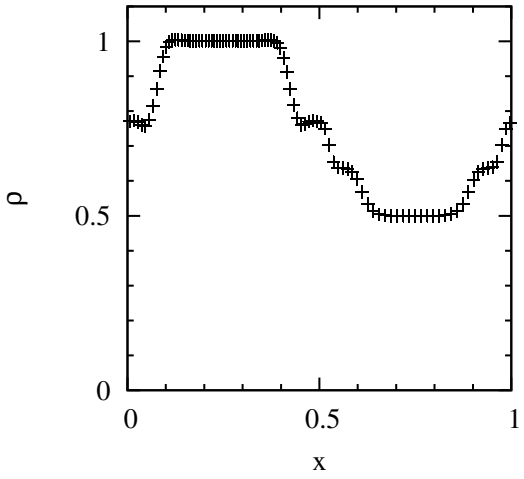

図 2:

2.3.2.6 OpenMP/MPI の利用

OpenMP や MPI を利用する場合を以下に示す。

- OpenMP のみ使用の場合
  - Makefile の編集
    - 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
  - make コマンドを実行する。
  - 「sbatch sph2.sh」というコマンドを実行し、ジョブを投入する。
  - ファイル stdout.log 内でスレッド数が 4 であることを確認できる。
- MPI のみ使用の場合
  - Makefile の編集
    - 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
  - make コマンドを実行する。
  - 「sbatch sph3.sh」というコマンドを実行し、ジョブを投入する。
  - ファイル stdout.log 内でプロセス数が 2 であることを確認できる。

- OpenMP と MPI の同時使用の場合
  - Makefile の編集
    - マクロ CC に MPI 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)
    - ”CFLAGS += -DPARTICLE_SIMULATOR_MPI_PARALLEL”の行のコメントアウトを外す
  - make コマンドを実行する。
  - 「sbatch sph4.sh」というコマンドを実行し、ジョブを投入する。
  - ファイル stdout.log 内でプロセス数が 2、スレッド数が 4 であることを確認できる。