

GPLUM ユーザーズガイド

石城陽太，小南淳子，牧野淳一郎，藤本正樹，岩澤全規

2020 年 1 月 29 日

目次

1	GPLUM の概要	3
2	ダウンロード	3
3	ファイル構成	3
3.1	ドキュメント	3
3.2	ソースファイル	3
3.3	サンプルファイル	3
3.4	FDPS	3
4	コンパイルと実行	3
4.1	Makefile の設定	3
4.2	コンパイル	5
4.3	実行	5
5	入力パラメータ	6
5.1	ランダムシード	6
5.2	入力・出力	6
5.3	微惑星円盤の初期条件	6
5.4	ガス円盤	7
5.5	領域分割	7
5.6	ツリー	7
5.7	タイムステップ	7
5.8	重力相互作用	8
5.9	衝突・破壊	9
6	出力ファイル	9
6.1	snapN.dat	9
6.2	collisionN.dat	10
6.3	removeN.dat	10
6.4	energy.dat	10
6.5	param.dat	11
7	衝突合体/破壊モデルの変更	11
7.1	Collision::collisionOutcome	11

7.2	Collision::readParameter	12
7.3	Collision::showParameter	12
8	ライセンス	12
9	現在分かっている不具合	13

1 GPLUM の概要

GPLUM は P^3T 法 (Oshino et al. 2011) を用いた惑星系形成 N 体計算用コードです。GPLUM は FDPS (Iwasawa et al. 2016) を使用しています。現在の最新バージョンは GPLUM-2.0 です。GPLUM-2.0 は FDPS-5.0d での動作確認をしています。

2 ダウンロード

GPLUM のソースコードは <https://github.com/YotaIshigaki/GPLUM> からダウンロードできます。

以下のコマンドを実行すれば、ディレクトリ GPLUM がダウンロードできます。

```
$ git clone https://github.com/YotaIshigaki/GPLUM
```

3 ファイル構成

3.1 ドキュメント

ドキュメント関係のファイルはディレクトリ `doc` にあります。UsersGuide_japanese.pdf が日本語のユーザーズガイドです。英語のユーザーズガイドは準備中です。

3.2 ソースファイル

ソースファイルは、ディレクトリ `src` にあります。

3.3 サンプルファイル

サンプルファイルは、ディレクトリ `sample` にあります。以下で、サンプルファイルの説明をします。

3.3.1 初期条件ファイル

INIT3000.dat は、初期条件ファイルのサンプルファイルです。1 行目にヘッダー、2 行目以降に粒子の情報が記述されています。初期条件ファイルの形式については、Section 6.4 を参照してください。

3.3.2 パラメータファイル

parameter.dat は、パラメータファイルのサンプルファイルです。設定できるパラメータについては、Section 5 を参照してください。

3.4 FDPS

FDPS に関するファイルは、ディレクトリ `FDPS-5.0d`, `FDPS-5.0d2` にあります。ディレクトリ内のファイル構成は、FDPS の仕様書を参照してください。FDPS-5.0d は FDPS-5.0d のファイル、FDPS-5.0d2 は FDPS-5.0d のいくつかの処理を更新したファイルです。

4 コンパイルと実行

4.1 Makefile の設定

src/Makefile で、コンパイラやオプションの設定ができます。

4.1.1 コンパイラ

src/Makefile の CC でコンパイラを指定します。GPLUM-2.0 は、GNU version 7.3.0 での動作確認をしています。

4.1.2 オプション

src/Makefile の CFLAGS でオプションを指定します。main_p3t.cpp のコンパイルにのみ使用するオプションは CFLAGS_M で、gravity_kernel.cpp のコンパイルにのみ使用するオプションは CFLAGS_G で指定します。以下で、指定できるオプションを説明します。

MPI MPI はデフォルトでは不使用です。MPI を使用するには、-DPARTICLE_SIMULATOR_MPI_PARALLEL を有効にしてください。

OpenMP OpenMP はデフォルトでは不使用です。OpenMP を使用するには、-DPARTICLE_SIMULATOR_THREAD_PARALLEL と -fopenmp を有効にしてください。

SIMD 命令 SIMD 命令はデフォルトでは不使用です。SIMD 命令は、AVX2 または AVX512DQ が使用可能です。いずれかに対応するオプションを有効にしてください。詳しくは、使用するコンパイラのマニュアルを参考にして下さい。

カットオフ半径 カットオフは、デフォルトでは共有カットオフが使用されます。独立カットオフを使用する場合は、-DUSE_INDIVIDUAL_CUTOFF を有効にしてください。

ツリーの多重極展開 ツリーの多重極展開は、デフォルトでは単極子までです。四重極子まで展開する場合は、-DUSE_QUAD を有効にしてください。

ツリーの計算精度 ツリー法での重力相互作用計算は、デフォルトでは単精度です。essential-particle 同士の重力相互作用計算を倍精度で計算する場合は、-DCALC_EPEP_64bit を有効にしてください。essential-particle と super-particle の重力相互作用計算を倍精度で計算する場合は、-DCALC_EPSP_64bit を有効にしてください。

LET の通信 local essential tree (LET) の MPI 通信は、デフォルトでは MPI_Alltoall を使用します。MPI_Allgather を使用する場合は、-DUSE_ALLGATHER_EXLET を有効にしてください。

ネイバー粒子探索 ネイバー粒子探索は、デフォルトでは 1 段階です。Ishigaki et al. (2020) で説明するように 2 段階にする場合は、-DUSE_RE_SEARCH_NEIGHBOR を有効にしてください。

ランダム速度 粒子のランダム速度は、デフォルトでは全ての粒子が準 Kepler 運動をしているものとして計算されます。粒子が等方的に運動している系を扱う場合は、-DISOTROPIC を有効にしてください。ランダム速度の計算方法については、Section 5.8 を参照してください。

ガス抵抗 デフォルトではガス抵抗はありません。ガス抵抗を考慮する場合は、-DGAS_DRAG を有効にしてください。ガス抵抗は、Adachi et al. (1976) のモデルを使用します。

indirect 項 デフォルトでは indirect 項は計算されません。indirect 項の効果を反映する場合は、-DINDIRECT_TERM を有効にしてください。

衝突判定 粒子同士の衝突判定は、デフォルトでは行われません。衝突判定を行う場合は、-DCOLLISION を有効にしてください。

衝突合体/破壊モデル 衝突判定を行う場合、デフォルトでは完全合体モデルが使用されます。-DKOMINAMI を有効にすると、Kominami et al. (in prep.) の衝突破壊モデルを使用します。-DCHAMBERS を有効にすると、Chambers (2013) の衝突破壊モデルを使用します。それ以外の衝突破壊モデルを使用する方法については、Section 7 を参照してください。

出力 -DOUTPUT_DETAIL を有効にすると、出力ファイルに出力される内容が増えます。詳細は Section 6.4 を参照してください。

テスト粒子 テスト粒子 (質量が 0 の粒子) が存在すると、FDPS が異常終了することがあります。

-DTEST_PTCL を有効にすると、テスト粒子の質量に `double` 型の数値的な最小値を設定することで異常終了を防ぎます。テスト粒子同士の相互作用は、`Hard` では計算されません。

4.2 コンパイル

コンパイルするには、ディレクトリ `GPLUM/src` に移動し、以下のコマンドを実行してください。

```
$ make
```

ディレクトリ `GPLUM/src` に実行ファイル `gplum.out` が作成されます。

4.3 実行

計算を実行するディレクトリに実行ファイル `gplum.out` とパラメータファイルを移動し、以下のように実行してください。

```
$ ./gplum.out
```

MPI 並列化を行う場合は、以下のように実行してください。

```
$ mpirun -n (プロセス数) gplum.out
```

カレントディレクトリに出力ディレクトリが作成され、出力ディレクトリに結果が出力されます。出力の形式については、Section 6 を参照してください。

以下で、実行時に指定できるオプションを説明します。同じパラメータを、実行時のオプションとパラメータファイルの両方で設定した場合、実行時のオプションの設定が優先されます。

- p: パラメータファイルを指定します。このオプションを指定しない場合は、パラメータファイルは `parameter.dat` となります。 `parameter.dat` というファイル名以外のパラメータファイルを用いる場合に指定してください。
- r: 計算を再スタートするときに指定してください。このオプションを指定すると、出力ディレクトリにあるスナップショットファイルの中で番号が最も大きいファイルから初期条件を読み込み、計算をスタートします。
- i: 初期条件ファイルを指定します。
- s: 初期条件の作成や、衝突破壊モデルで用いられる乱数のシードを指定します。
- e: 実計算時間の上限を指定します。計算時間がこの時間を超えた場合、パラメータファイルで設定している終了条件を無視して終了処理に移行します。単位は時間です。
- o: 出力ディレクトリ名を指定します。このオプションで設定した名前のディレクトリに計算結果が出力されます。その名前のディレクトリが実行ディレクトリにない場合は、その名前のディレクトリが作成されます。
- D: ツリーのタイムステップ Δt_{tree} を整数で指定します。このオプションで指定した整数を n とすると、ツリーのタイムステップは $\Delta t_{\text{tree}} = 2^{-n}$ となります。
- R: 粒子のカットオフ半径を決めるパラメータ \tilde{R}_{cut} を指定します。 \tilde{R}_{cut} について、詳しくは Section 5.8 を参照してください。

5 入力パラメータ

以下で、パラメータファイルで設定できるパラメータを説明します。距離、質量、時間の単位は、それぞれ AU, M_{\odot} , year/ 2π です。値の末尾に **MKS** や **CGS** とつけると、それぞれ **MKS** 単位系, **CGS** 単位系の値で記述することができます。例えば、密度の次元を持つパラメータの値として **2.0CGS** と記述すると、 2.0 g/cm^3 が M_{\odot}/AU^3 の単位に変換されて設定されます。以下の各項目の () の値は、パラメータが設定されなかった場合に設定されるデフォルト値です。

5.1 ランダムシード

seed: ランダムシード (1). 初期条件の作成や、衝突破壊モデルで用いられる乱数のシードです。

5.2 入力・出力

Init_file: 初期条件ファイル名 (INIT_3000.dat). 初期条件ファイルの形式は、Section 6.1 で説明します。

Header: 初期条件ファイルにヘッダがある場合に 1, ヘッダがない場合に 0 を設定します (0). ヘッダの形式は、Section 6.1 で説明します。ヘッダがない場合、時刻は 0 となり、粒子数やエネルギーは初期条件ファイルから計算されます。

output_dir: 出力ディレクトリ名 (OUTPUT). ここで設定した名前のディレクトリに計算結果が出力されます。その名前のディレクトリがカレントディレクトリにない場合は、その名前のディレクトリが作成されます。

Restart: 計算を再スタートさせる場合は 1, そうでない場合には 0 を設定します (0). この値に 1 を設定すると、出力ディレクトリにあるスナップショットファイルの中で番号が最も大きいファイルから初期条件を読み込み、計算をスタートします。

5.3 微惑星円盤の初期条件

初期条件ファイルから初期条件を読み込む場合には使われません。微惑星円盤の面密度 Σ_d が,

$$\Sigma_d = \begin{cases} 10f_d \left(\frac{r}{1 \text{ AU}}\right)^{-p} \text{ g/cm}^{-2} & (r \leq a_{\text{ice}}) \\ 10f_d\eta_{\text{ice}} \left(\frac{r}{1 \text{ AU}}\right)^{-p} \text{ g/cm}^{-2} & (r > a_{\text{ice}}) \end{cases} \quad (1)$$

と表されるモデルを使用します (Hayashi 1981). r は中心星からの距離です。

makeInit: 粒子の初期条件をプログラム内で作成する場合に 1, ファイルから読み込む場合に 0 を設定します (0). この値に 1 を設定すると、初期条件ファイルが設定されていても、プログラム内で粒子の初期条件を作成します。

n_init: 初期粒子数 n_{init} (0).

m_init: 初期粒子質量 m_{init} (0.0). n_{init} , m_{init} のどちらか片方が指定された場合, (1) に従うようにもう一方の値が設定され, (1) の分布に従って粒子が配置されます。どちらも指定された場合は、設定された f_d の値に関わらず, (1) に従うように f_d の値を調整して粒子が配置されます。どちらも指定されない場合は、エラーとなります。

p: Σ_d の r についての冪 p (1.5).

f_dust: Σ_d のスケーリングファクター f_d (0.71).

eta_ice: 雪線の外側領域における雪線境界ファクター η_{ice} (4.2).

a_in: 初期微惑星が配置される領域の内側境界 a_{in} (0.98).

a_out: 初期微惑星が配置される領域の外側境界 a_{out} (1.02).

a_ice: 雪線の半径 a_{ice} (2.0).

5.4 ガス円盤

マクロ GAS_DRAG が定義されない場合には使用されません。ガス円盤の面密度 Σ_g は,

$$\Sigma_g = 2400 f_g \left(\frac{a}{1 \text{ AU}} \right)^{-p} \text{ g/cm}^{-2} \quad (2)$$

のように表され, ガス密度 ρ_g , 円盤温度 T が,

$$\rho_g = 1.4 \times 10^{-9} f_g \left(\frac{r}{1 \text{ AU}} \right)^{-\alpha} \text{ g/cm}^{-3} \quad (3)$$

$$T = 2.8 \times 10^2 \left(\frac{r}{1 \text{ AU}} \right)^{-\beta} \text{ K} \quad (4)$$

のように表されるモデルを使用します (Hayashi 1981). r は中心星からの距離です。ガス抵抗は, Adachi et al. (1976) のモデルを使用します。

alpha_gas: ρ_g の r についての冪 α (11.0/4.0).

beta_gas: T の r についての冪 β (0.5).

f_gas: Σ_g のスケーリングファクター f_g (0.71).

tau_gas: ガス円盤の散逸時定数 τ_g (0). この値を 0 以外に設定すると, ρ_g が $\exp(-t/\tau_{\text{gas}})$ に従って変化します。 t は系の時刻です。この値を 0 に設定すると, ρ_g は時間変化しません。

C_d: 無次元のガス抵抗係数 C_D (1.0).

mu: ガスの平均分子量 μ (2.34).

5.5 領域分割

coef_ema: 領域分割の指数移動平均の平滑化係数 (0.3).

nx,ny: x, y 方向のルートドメインの分割数 n_x, n_y (MPI プロセス数を n_{proc} として, $n_x = \sqrt{n_{\text{proc}}}$ に最も近い $n_{\text{proc}} = n_x n_y$ となる整数).

reset_step: 領域分割をやり直すステップの間隔 (1024). カットオフ半径の再設定もこの間隔で行われます。

5.6 ツリー

theta: ツリーの見込み角 θ (0.5).

n_leaf_limit: ツリーを切るのをやめる粒子数の上限 (8).

n_group_limit: 相互作用リストを共有する粒子数の上限 (256).

5.7 タイムステップ

t_end: プログラムを終了する系の時刻 t_{end} (1.0).

dt_tree: ツリーのタイムステップ Δt_{tree} (1/32). 2 の冪乗でなければなりません。

dt_snap: 出力の時間間隔 Δt_{snap} (1/32). Δt_{tree} の整数倍でなければなりません。

dt_min: Hermite 法におけるタイムステップの最小値 Δt_{min} (1/8192). 2 の冪乗でなければなりません。また, $\Delta t_{\text{min}} < 0.5 \Delta t_{\text{tree}}$ でなければなりません。

eta: Hermite 法におけるタイムステップを決めるパラメータ η (0.01).

eta_sun: Hermite 法におけるタイムステップを決めるパラメータ η_{sun} (0.01).

eta_0: Hermite 法における初期タイムステップを決めるパラメータ η_0 (0.001).

eta_sun0: Hermite 法における初期タイムステップを決めるパラメータ $\eta_{\text{sun},0}$ (0.001).

alpha: Hermite 法におけるタイムステップを決めるパラメータ α (1.0).

Hermite 法において、次ステップのタイムステップは、以下を基準に 2 の冪乗で設定されます.

$$\Delta t^* = \min \left(\eta \sqrt{\frac{\sqrt{|\mathbf{a}_{\text{int}}|^2 + \alpha^2 a_0^2} |\mathbf{s}_{\text{int}}| + |\mathbf{j}_{\text{int}}|^2}{|\mathbf{j}_{\text{int}}| |\mathbf{c}_{\text{int}}| + |\mathbf{s}_{\text{int}}|^2}}, \eta_{\text{sun}} \sqrt{\frac{|\mathbf{a}_{\text{sun}}| |\mathbf{s}_{\text{sun}}| + |\mathbf{j}_{\text{sun}}|^2}{|\mathbf{j}_{\text{sun}}| |\mathbf{c}_{\text{sun}}| + |\mathbf{s}_{\text{sun}}|^2}} \right). \quad (5)$$

\mathbf{a} , \mathbf{j} , \mathbf{s} , \mathbf{c} は粒子の Hard の加速度, 加速度の 1 階, 2 階, 3 階時間微分で, 添字 int は粒子間相互作用を, sun は中心星との相互作用を表します. a_0 は近接粒子との相互作用によるカットオフのない加速度の大ききで, 近接粒子が複数ある場合はその平均です. ただし, 初期タイムステップや衝突直後のタイムステップは, 以下を基準に 2 の冪乗で設定されます.

$$\Delta t^* = \min \left(\eta_0 \frac{\sqrt{|\mathbf{a}_{\text{int}}|^2 + \alpha^2 a_0^2}}{|\mathbf{j}_{\text{int}}|}, \eta_{\text{sun},0} \frac{|\mathbf{a}_{\text{sun}}|}{|\mathbf{j}_{\text{sun}}|} \right). \quad (6)$$

5.8 重力相互作用

m_sun: 中心星質量 m_{Sun} (1.0).

dens: 粒子密度 ρ_{planet} (5.049667×10^6).

eps: 重力のソフトニングパラメータ ϵ (0.0).

r_cut_max, r_cut_min: カットオフ半径の最大値 $r_{\text{cut,max}}$, 最小値 $r_{\text{cut,min}}$ (0.0, 0.0).

p_cut: カットオフ半径の軌道長半径についての冪 p_{cut} (0.0).

R_cut: 粒子のカットオフ半径を決めるパラメータ \tilde{R}_{cut} (1.0).

R_search0, R_search1: 粒子の探索半径を決めるパラメータ $\tilde{R}_{\text{search0}}, \tilde{R}_{\text{search1}}$ (1.0, 1.0). $\tilde{R}_{\text{search0}} \geq 1$ でなければなりません.

粒子 i の外側カットオフ半径 $r_{\text{out},i}$ は, $r_{\text{cut,max}} > 0$ のとき,

$$r_{\text{out},i} = \begin{cases} r_{\text{cut,max}} & (r_{\text{cut,max}} \leq r_{\text{out},i}^*) \\ r_{\text{out},i}^* & (r_{\text{cut,min}} \leq r_{\text{out},i}^* < r_{\text{cut,max}}) \\ r_{\text{cut,min}} & (r_{\text{out},i}^* < r_{\text{cut,min}}) \end{cases}, \quad (7)$$

$r_{\text{cut,max}} = 0$ のとき,

$$r_{\text{out},i} = \max(r_{\text{out},i}^*, r_{\text{cut,min}}) \quad (8)$$

と設定されます. ここで, $r_{\text{out},i}^*$ は,

$$r_{\text{out},i}^* = \tilde{R}_{\text{cut}} \left(\frac{a_i}{1 \text{ AU}} \right)^{-p_{\text{cut}}} r_{\text{Hill},i} \quad (9)$$

$$r_{\text{Hill},i} = \left(\frac{m_i}{3M_{\odot}} \right)^{1/3} a_i \quad (10)$$

となります. m_i , a_i はそれぞれ粒子 i の質量, 軌道長半径です. 粒子の軌道離心率が 0.5 以上の場合は, a_i の代わりに中心星からの距離が使用されます.

粒子 i, j の相互作用についての外側カットオフ半径 $r_{\text{out},ij}$ は, 共有カットオフ半径の場合,

$$r_{\text{out},ij} = \max_k (r_{\text{out},k}) \quad (11)$$

独立カットオフ半径の場合,

$$r_{\text{out},ij} = \max(r_{\text{out},i}, r_{\text{out},j}) \quad (12)$$

と設定されます.

粒子 i の探索半径 $r_{\text{search},i}$ は,

$$r_{\text{search},i} = \tilde{R}_{\text{search}0} r_{\text{out},i} + \tilde{R}_{\text{search}1} v_{\text{ran}} \Delta t_{\text{tree}} \quad (13)$$

と設定されます. v_{ran} は系の粒子のランダム速度です. 粒子のランダム速度 v_{ran} は, デフォルトでは,

$$v_{\text{ran}} = \sqrt{\langle e^2 \rangle + \langle i^2 \rangle} v_{\text{Kep}} \quad (14)$$

と設定されます. e, i は粒子の軌道離心率, 傾斜角, v_{Kep} は Kepler 速度です. マクロ ISOTROPIC が定義されている場合は, v_{ran} は,

$$v_{\text{ran}} = \sqrt{\langle |\mathbf{v}|^2 \rangle - |\langle \mathbf{v} \rangle|^2} \quad (15)$$

と設定されます. \mathbf{v} は粒子の速度です. 粒子 i, j の相互作用についての探索半径 $r_{\text{search},ij}$ は, $r_{\text{search},i}$ を用いて式 (11),(12) と同様に設定されます.

gamma: 粒子の外側カットオフ半径, 内側カットオフ半径の比 $\gamma = r_{\text{in}}/r_{\text{out}}$ (0.1).

r_max: リング領域の外側境界 r_{max} (40). 粒子と中心星の距離がこの距離より大きくなったらその粒子を削除します.

r_min: リング領域の外側境界 r_{min} (0.1). 粒子と中心星の距離がこの距離より小さくなったらその粒子を削除します.

5.9 衝突・破壊

f: 粒子の半径のふくらし係数 f (1.0).

その他, **collisionB.h** で定義される **Collision** クラスで設定されるパラメータもパラメータファイルから読み込むことができます. 詳細は, Section 7 を参照してください.

6 出力ファイル

出力ディレクトリには, **snapN.dat** と, **collisionN.dat**, **removeN.dat**, **energy.dat**, **param.dat** が出力されます. N には 6 桁の整数が 0 から入ります.

6.1 snapN.dat

スナップショットファイル **snapN.dat** の形式は, 1 行目がヘッダー, 2 行目から各粒子の情報です. ヘッダーには, 系の情報が以下の順に出力されます.

t n ID_{MAX} $E_{\text{tot,init}}$ $E_{\text{kin,init}}$ $E_{\text{sun,init}}$ $E_{\text{planet,init}}$ ΔE_{init} $E_{\text{tot,now}}$ $E_{\text{kin,now}}$ $E_{\text{sun,now}}$ $E_{\text{planet,now}}$ ΔE_{now}

t は系の時刻, n は粒子数, ID_{MAX} は粒子 ID の最大値, $E_{\text{tot}}, E_{\text{kin}}, E_{\text{sun}}, E_{\text{planet}}, \Delta E$ はそれぞれ全力学的エネルギー, 運動エネルギー, 中心星重力ポテンシャルエネルギー, 粒子間相互作用ポテンシャルエネルギー, 衝突やガス抵抗, 粒子の消去に伴う力学的エネルギー変化で, **init,now** はそれぞれ初期と時刻 t における値を表します. $E_{\text{tot}} = E_{\text{kin}} + E_{\text{sun}} + E_{\text{planet}}$ となり, 理論的には $E_{\text{tot}} - \Delta E$ が保存量になります.

2 行目からの各粒子の情報は以下の順に出力されます.

ID m x y z v_x v_y v_z n_{neighbor}

ID は粒子 ID, m は粒子の質量, $\mathbf{r}_i = (x, y, z)$ は粒子の位置, $\mathbf{v}_i = (v_x, v_y, v_z)$ は粒子の速度, n_{neighbor} は粒子の近傍粒子数です.

初期条件ファイルも、スナップショットファイルと同じ形式で記述してください。ただし、初期条件ファイルの n_{neighbor} は、計算には使用されません。初期条件で同じ粒子 ID を持つ粒子が複数存在する場合は、正しい動作が保証されません。

パラメータ **Header** が 1 の場合は、初期条件ファイルのヘッダーからエネルギーなどの情報を読み込み、それらを使用します。パラメータ **Header** が 0 の場合は、1 行目から粒子の情報を読み込みます。この場合は、 $t = 0$, $\Delta E_{\text{init}} = 0.0$ と設定し、ヘッダーの情報は初期条件から計算されます。初期条件ファイルのヘッダー情報を使用しない場合は、初期条件ファイルのヘッダーの行を削除し、**Header** を 0 として初期条件を読み込んでください。

計算途中で衝突破壊などで新しい粒子が生成される場合は、粒子 ID は $\text{ID}_{\text{MAX}} + 1$ から順に設定されます。

スナップショットファイルに出力する粒子情報、初期条件ファイルから読み込む粒子情報は、それぞれ `particle.h` の `FPGrav::writeAscii` 関数、`FPGrav::readAscii` 関数を書き換えることで変更できます。ただし、計算を再スタートするためには、`FPGrav::writeAscii` 関数で書き出す情報と `FPGrav::readAscii` 関数で読み込む情報の形式が一致している必要があります。

6.2 collisionN.dat

`collisionN.dat` には、`snap(N - 1).dat` の時刻から `snapN.dat` の時刻までの間の時間に起こった衝突について、情報が以下の順に出力されます。

t ID_{imp} ID_{tar} n_{frag} ID_{frag} m_{imp} m_{tar} m_{frag} r_{imp} v_{imp} θ_{imp} Hit&Run $|\Delta \mathbf{r}_g|/|\mathbf{r}_g|$ $|\Delta \mathbf{v}_g|/|\mathbf{v}_g|$

t は衝突が起こった時刻、ID, m はそれぞれ粒子の ID と質量で、imp, tar はそれぞれインパクター粒子、ターゲット粒子を表します。衝突する 2 粒子のうち、質量が小さい方をインパクター粒子、大きい方をターゲット粒子としています。 n_{frag} , ID_{frag}, m_{frag} はそれぞれ衝突で生じる破片粒子数、1 つ目の破片粒子の ID、破片粒子の総質量で、この衝突により生じる破片粒子の ID は ID_{frag} から ID_{frag} + n_{frag} - 1 までとなります。 r_{imp} , v_{imp} , θ_{imp} はターゲット粒子に対するインパクター粒子の相対距離、相対速度、衝突角です。Hit&Run は、衝突がヒットアンドランであれば 1, そうでなければ 0 となります。 \mathbf{r}_g , \mathbf{v}_g はそれぞれインパクター粒子、ターゲット粒子、破片粒子の系の重心位置、重心速度で、 $\Delta \mathbf{r}_g$, $\Delta \mathbf{v}_g$ はそれらの衝突前と衝突後の差です。

`collision.dat` に出力する情報は、`collisionA.h` の `Collision0::write2File` 関数を書き換えるか、`collisionB.h` の `Collision` クラスで `write2File` をオーバーライドして書き換えることで変更できます。衝突モデルの変更の方法については Section 7 を参照ください。

6.3 removeN.dat

`removeN.dat` には、`snap(N - 1).dat` の時刻から `snapN.dat` の時刻までの間の時間に削除された粒子の情報が以下の順に出力されます。

t ID m x y z v_x v_y v_z

t は粒子が削除された時刻、ID は粒子 ID, m は粒子の質量、 $\mathbf{r}_i = (x, y, z)$ は粒子の位置、 $\mathbf{v}_i = (v_x, v_y, v_z)$ は粒子の速度です。

6.4 energy.dat

`energy.dat` には、スナップショットファイルを出力するごとに系の情報が以下の順に出力されます。

t n $E_{\text{tot,now}}$ $(E_{\text{tot,now}} - E_{\text{tot,init}} - \Delta E_{\text{now}})/E_{\text{tot,init}}$ $n_{\text{largecluter}}$ n_{cluster} $n_{\text{isoparticle}}$

$n_{\text{largecluster}}$, n_{cluster} , $n_{\text{isoparticle}}$ は, 直前の 1 ステップの最大のクラスターの粒子数, クラスター数, 近接粒子を持たない粒子数です.

マクロ `OUTPUT_DETAIL` を定義している場合は, この後に, 最大質量, 平均質量, 平均近接粒子数も出力されます.

6.5 param.dat

`param.dat` には, 設定されたパラメータが出力されます. 計算を再スタートすると, その度にその計算で設定されたパラメータが書き加えられます.

7 衝突合体/破壊モデルの変更

衝突合体/破壊モデルは, `collisionB.h` の `Collision` クラスで設定できます. `Collision` クラスは, `collisionA.h` の `Collision0` クラスを継承しています. 現在は, サンプルとして `collisionB.h` には完全合体モデル, Kominami et al. (in prep.), Chambers (2013) の衝突破壊モデルに対応する `Collision` クラスが記述されています.

`Collision` クラスには以下のメンバ関数を設定してください.

7.1 Collision::collisionOutcome

```
template <class Tp>
PS::S32 Collision::collisionOutcome(std::vector<Tp> & pfrag)
```

引数 `pfrag`: 出力. `std::vector<Tp>` 型. 衝突により生じる破片粒子のリスト. `Tp` は `particle.h` の `FPHard` クラスを想定しています.

返値 `PS::S32` 型. 衝突により生じる破片粒子数.

機能 以下のメンバ変数 (入力) の値を用いて, メンバ変数 (出力) の値を設定します.

- メンバ変数 (入力)
 - `pos_imp`: `PS::F64vec` 型. 衝突直前のインパクター粒子の位置.
 - `pos_tar`: `PS::F64vec` 型. 衝突直前のターゲット粒子の位置.
 - `pos_g`: `PS::F64vec` 型. 衝突直前のインパクター粒子, ターゲット粒子の重心位置.
 - `vel_imp`: `PS::F64vec` 型. 衝突直前のインパクター粒子の速度.
 - `vel_tar`: `PS::F64vec` 型. 衝突直前のターゲット粒子の速度.
 - `vel_g`: `PS::F64vec` 型. 衝突直前のインパクター粒子, ターゲット粒子の重心速度.
 - `mass_imp`: `PS::F64` 型. 衝突直前のインパクター粒子の質量.
 - `mass_tar`: `PS::F64` 型. 衝突直前のターゲット粒子の質量.
 - `col_angle`: `PS::F64` 型. 衝突角.
- メンバ変数 (出力)
 - `pos_imp_new`: `PS::F64vec` 型. 衝突直後のインパクター粒子の位置.
 - `pos_tar_new`: `PS::F64vec` 型. 衝突直後のターゲット粒子の位置.
 - `vel_imp_new`: `PS::F64vec` 型. 衝突直後のインパクター粒子の速度.
 - `vel_tar_new`: `PS::F64vec` 型. 衝突直後のターゲット粒子の速度.
 - `mass_frag`: `PS::F64` 型. 破片粒子の総質量.

n_frag: PS::S32 型. 破片粒子数.

HitAndRun: bool 型. 衝突がヒットアンドランなら 1, そうでなければ 0.

また, 生じた破片粒子のリストを `std::vector<Tp>` 型で作り, 引数の **pfrag** に代入します. 破片粒子が生じない場合は, **mass_frag** と **n_frag** を 0 にし, **pfrag** のサイズを 0 にします.

ヒットアンドラン衝突の場合, 破片は全てインパクター粒子から生じたものとなります. ヒットアンドラン衝突でない場合, **pos_imp_new** と **pos_tar_new**, **vel_imp_new** と **vel_tar_new** は等しくなければなりません.

破片粒子数 **n_frag** を返します.

また, `Collision` クラスでパラメータの設定を行う必要がある場合, 必要に応じて, 以下のメンバ関数を設定してください.

7.2 Collision::readParameter

```
static void readParameter(std::string name, std::string value)
```

引数 **name**: 入力. `std::string` 型. パラメータ変数名.

value: 入力. `std::string` 型. 文字列として読み取ったパラメータ変数の値.

返値 なし.

機能 パラメータファイルから読み込んだパラメータ変数名とパラメータ変数名を受け取り, メンバ変数の値を設定します.

7.3 Collision::showParameter

```
static void readParameter()
```

引数 なし.

返値 なし.

機能 標準出力にパラメータの値を出力します.

```
static void readParameter(std::ofstream & fout)
```

引数 **fout**: 入力. `std::ofstream&` 型. 出力ファイル (`param.dat`).

返値 なし.

機能 `param.dat` にパラメータの値を出力します.

8 ライセンス

MIT ライセンスに準じます. GPLUM を使用した結果を発表する場合は, 以下の論文の引用をお願いします*1.

*1 発表する直前に最新の情報を確認してください

- 1) Iwasawa et al. (2016, PASJ, 68, 59)
- 2) Ishigaki (2019, M. Thesis, University of Tokyo) または Ishigaki et al. (in prep.)

Copyright 2019 Yota Ishigaki, Junko Kominami, Junichiro Makino, Masaki Fujimoto, Masaki Iwasawa

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

9 現在分かっている不具合

- プログラムが以下のようなエラーを出力して異常終了することがあります。

```
gplum.out: ../FDPS-5.0d2/src/memory_pool.hpp:185: static void ParticleSimulator::MemoryPool::alloc(size_t, int&, void*, void*&): Assertion 'N_SEGMENT_LIMIT > getInstance().n_segment_+1' failed.
```

```
=====
```

```
= BAD TERMINATION OF ONE OF YOUR APPLICATION PROCESSES
= PID 64 RUNNING AT LAPTOP-PMT9ABH2
= EXIT CODE: 134
= CLEANING UP REMAINING PROCESSES
= YOU CAN IGNORE THE BELOW CLEANUP MESSAGES
```

```
=====
```

```
YOUR APPLICATION TERMINATED WITH THE EXIT STRING: Aborted (signal 6)
```

```
This typically refers to a problem with your application.
```

```
Please see the FAQ page for debugging suggestions
```

Memory Pool 関連のエラーは FDPS-5.0d のバグのようです。FDPS-5.0c か、最新バージョンの FDPS を使ってください。

不明な点やバグ等あれば、石城 (y.ishigaki@stp.isas.jaxa.jp) までご連絡ください。

