

# GPLUM ユーザーズガイド

石城陽太

2019 年 4 月 22 日

GPLUM は P<sup>3</sup>T 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-1.1 です。GPLUM-1.1 は FDPS-5.0d での動作確認をしています。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。

## 1 ライセンス

GPLUM を使用した結果について発表する場合は、以下の論文を引用してください\*<sup>1</sup>。

- 1) Ishigaki, Y., 2019. Development of N-body simulation code for planetary system formation with particle-particle particle-tree scheme. M. thesis, University of Tokyo.
- 2) Iwasawa, M., Tanikawa, A., Hosono, N., Nitadori, K., Muranushi, T., Makino, J., 2016. Implementation and performance of FDPS: A framework for developing parallel particle simulation codes. Publications of the Astronomical Society of Japan, 68 (4), 54 – 75.

## 2 コンパイルと実行

### 2.1 コンパイル

GitHub から ディレクトリ GPLUM をダウンロードしてください。コンパイルするには、ディレクトリ GPLUM/src に移動し、

```
$ make
```

を実行してください。ディレクトリ GPLUM/src に実行ファイル `gplum.out` が作られます。実行ファイルを作り直す場合は、

```
$ make clean
```

を実行してから、実行ファイルを作ってください。

以下で、コンパイル時に Makefile で定義するマクロや指定するオプションを説明します。

**OpenMP** OpenMP はデフォルトでは不使用です。OpenMP を使用するには、マクロ `PARTICLE_SIMULATOR_THREAD_PARALLEL` を定義し、オプション `-fopenmp` を指定してください。

**MPI** MPI はデフォルトでは不使用です。MPI を使用するには、マクロ `PARTICLE_SIMULATOR_MPI_PARALLEL` を定義してください。

**SIMD 命令** AVX2 または AVX512 のいずれかに対応するオプションを指定してください。

---

\*<sup>1</sup> 発表する直前に最新の情報を確認してください

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、マクロ `USE_QUAD` を定義してください。

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、マクロ `CALC_EP_64bit` を定義してください。Essential-Particle と Super-Particle の重力相互作用計算を倍精度で計算する関数は未実装です。

カットオフ半径 カットオフ半径は、デフォルトでは共有カットオフ半径が使用されます。独立カットオフ半径を使用する場合は、マクロ `USE_INDIVIDUAL_RADII` を定義してください。

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、マクロ `COLLISION` を定義してください。

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。Kominami et al. (2019, in prep.) の衝突破壊モデルを使用する場合は、マクロ `KOMINAMI` を定義してください。Chambers (2013) の衝突破壊モデルを用いる場合は、マクロ `CHAMBERS` を定義してください。`KOMINAMI` と `CHAMBERS` が両方定義された場合は、Kominami et al. (2019, in prep.) のモデルが使用されます。

出力 マクロ `OUTPUT_DETAIL` を定義すると、出力ファイルに出力される内容が増えます。詳細は §4.2 を参照ください。

ランダム速度 粒子のランダム速度  $v_{\text{ran}}$  は、デフォルトでは

$$v_{\text{ran}} = \sqrt{\langle e^2 \rangle + \langle i^2 \rangle} v_{\text{Kep}} \quad (1)$$

と設定されます。 $e, i$  は粒子の軌道離心率、傾斜角、 $v_{\text{Kep}}$  は Kepler 速度です。粒子の軌道が円軌道に近くない場合、マクロ `ISOTROPIC` を定義すると、ランダム速度が、

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

と設定されます。 $\boldsymbol{v}$  は粒子の速度です。

## 2.2 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、

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

を実行してください。MPI 並列化を行う場合は、

```
$ mpirun -n (プロセス数) main_p3t.out
```

のように実行してください。

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。パラメータファイルで設定している初期条件は無視されます。
- 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  $\Delta t_{\text{tree}}$  を指定します。このオプションで  $n$  を設定した場合、 $\Delta t_{\text{tree}} = 2^{-n}$  と設定されます。
- R: 粒子のカットオフ半径を決めるパラメータ  $\tilde{R}_{\text{cut}}$  を設定します。

### 3 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,  $M_{\odot}$ ,  $\text{year}/2\pi$  で記述されることを想定しています。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、 $2.0 \text{ g/cm}^3$  が  $M_{\odot}/\text{AU}^3$  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。

#### 3.1 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。

#### 3.2 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、§4 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、§4 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。

#### 3.3 微惑星円盤の初期条件

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  $\Sigma_d$  が、

$$\Sigma_d = \begin{cases} 10f_d \left(\frac{r}{1 \text{ AU}}\right)^{-p} \text{ g/cm}^{-2} & (r \leq a_{\text{ice}}) \\ 10f_d\eta_{\text{ice}} \left(\frac{r}{1 \text{ AU}}\right)^{-p} \text{ g/cm}^{-2} & (r > a_{\text{ice}}) \end{cases} \quad (3)$$

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、初期条件ファイルの有無に関わらず、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  $n_{\text{init}}$  (0)。

**m\_init:** 初期粒子質量  $m_{\text{init}}$  (0.0)。 $n_{\text{init}}$ ,  $m_{\text{init}}$  のどちらか片方が指定された場合、(3) に従うようにもう一方の値が設定され、(3) 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, (3) 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:**  $\Sigma_d$  の  $r$  についての冪  $p$  (1.5).

**f\_dust:**  $\Sigma_d$  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  $\eta_{ice}$  (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).

### 3.4 ガス円盤

ガス抵抗を考慮しない場合には使われません. ガス円盤の面密度  $\Sigma_g$  は,

$$\Sigma_g = 2400 f_g \left( \frac{a}{1 \text{ AU}} \right)^{-p} \text{ g/cm}^{-2} \quad (4)$$

のように表され, ガス密度  $\rho_g$ , 円盤温度  $T$  が,

$$\rho_g = 1.4 \times 10^{-9} f_g \left( \frac{r}{1 \text{ AU}} \right)^{-\alpha} \text{ g/cm}^{-3} \quad (5)$$

$$T = 2.8 \times 10^2 \left( \frac{r}{1 \text{ AU}} \right)^{-\beta} \text{ K} \quad (6)$$

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:**  $\rho_d$  の  $r$  についての冪  $\alpha$  (11.0/4.0).

**alpha\_gas:**  $T$  の  $r$  についての冪  $\beta$  (0.5).

**f\_gas:**  $\Sigma_g$  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  $\tau_g$  (0). この値を 0 以外に設定すると,  $\rho_g$  が  $\exp(-t/\tau_{gas})$  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,  $\rho_g$  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  $\mu$  (2.34).

### 3.5 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,  $n_x = \sqrt{n_{proc}}$  に最も近い  $n_{proc} = n_x n_y$  となる整数).

### 3.6 ツリー

**theta:** ツリーの見込み角  $\theta$  (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).

### 3.7 タイムステップ

**t\_end:** プログラムを終了する系の時刻  $t_{end}$  (1/4).

**dt\_tree:** ツリーのタイムステップ  $\Delta t_{tree}$  (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  $\Delta t_{\text{snap}}$  (1/32).  $\Delta t_{\text{tree}}$  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  $\Delta t_{\text{min}}$  (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  $\eta$  (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  $\eta_0$  (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  $\alpha$  (0.1).

### 3.8 重力相互作用

m\_sun: 中心星質量  $m_{\text{Sun}}$  (1.0).

dens: 粒子密度  $\rho_{\text{planet}}$  ( $5.049667 \times 10^6$ ).

eps: 重力のソフトニングパラメータ  $\epsilon$  (0.0).

rHill\_max, rHill\_min: カットオフ半径, 探索半径の設定に用いられる Hill 半径  $r_{\text{Hill}}$  の最大値  $r_{\text{Hill,max}}$ , 最小値  $r_{\text{Hill,min}}$  (0.0, 0.0). 粒子の質量, 軌道長半径から計算される Hill 半径は

$$r_{\text{Hill},i}^* = \begin{cases} \left(\frac{m_i}{3M_\odot}\right)^{1/3} a_i & (e_i < 1) \\ \left(\frac{m_i}{3M_\odot}\right)^{1/3} r_i & (e_i \geq 1) \end{cases} \quad (7)$$

と設定され,  $r_{\text{Hill},i}$  は,  $r_{\text{Hill,max}} > 0$  のとき,

$$r_{\text{Hill},i} = \begin{cases} r_{\text{Hill,max}} & (r_{\text{Hill,max}} \leq r_{\text{Hill},i}^*) \\ r_{\text{Hill},i}^* & (r_{\text{Hill,min}} \leq r_{\text{Hill},i}^* < r_{\text{Hill,max}}) \\ r_{\text{Hill,min}} & (r_{\text{Hill},i}^* < r_{\text{Hill,min}}) \end{cases} \quad (8)$$

と設定されます.  $m_i, a_i, e_i, r_i$  はそれぞれ粒子  $i$  の質量, 軌道長半径, 軌道離心率, 中心星からの距離です.  $r_{\text{Hill,max}} = 0$  のときは,

$$r_{\text{Hill},i} = \max(r_{\text{Hill},i}^*, r_{\text{Hill,min}}) \quad (9)$$

と設定されます.

R\_cut: 粒子のカットオフ半径を決めるパラメータ  $\tilde{R}_{\text{cut}}$  (1.0). (8) または (9) で設定される  $r_{\text{Hill},i}$  を用いて, 粒子  $i$  についての外側カットオフ半径  $r_{\text{out},i}$  が,

$$r_{\text{out},i} = \tilde{R}_{\text{cut}} r_{\text{Hill},i} \quad (10)$$

と設定されます. 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.

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).  $\tilde{R}_{\text{search0}} \geq \tilde{R}_{\text{cut}}$  でなければなりません. 粒子  $i$  の探索半径  $r_{\text{search},i}$  は,

$$r_{\text{search},i} = \tilde{R}_{\text{search0}} r_{\text{Hill},i} + \tilde{R}_{\text{search1}} v_{\text{ran}} \Delta t_{\text{tree}} \quad (13)$$

と設定されます.  $v_{\text{ran}}$  は系の粒子のランダム速度です (§2.1 のランダム速度を参照). 粒子  $i, j$  の相互作用についての探索半径は, 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  $r_{\text{max}}$  (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  $r_{\text{min}}$  (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.

### 3.9 衝突・破壊

**f:** 粒子の半径のふくらまし係数  $f$  (5.0).

その他, `collisionB.h` で定義される `Collision` クラスで設定させ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 (§5 参照).

## 4 出力ファイル

出力ディレクトリには, スナップショットファイル `snap?????.dat` と, `energy.dat`, `collision.dat`, `param.dat`, `remove.dat` が出力されます.

### 4.1 スナップショットファイル

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

$t \quad n \quad E_{\text{tot,init}} \quad E_{\text{kin,init}} \quad E_{\text{sun,init}} \quad E_{\text{planet,init}} \quad \Delta E_{\text{init}} \quad E_{\text{tot,now}} \quad E_{\text{kin,now}} \quad E_{\text{sun,now}} \quad E_{\text{planet,now}} \quad \Delta E_{\text{now}}$

$t$  は系の時刻,  $n$  は粒子数,  $E_{\text{tot}}, E_{\text{kin}}, E_{\text{sun}}, E_{\text{planet}}, \Delta E$  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.

$\text{ID} \quad m \quad x \quad y \quad z \quad v_x \quad v_y \quad v_z \quad n_{\text{neighbor}}$

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,  $n_{\text{neighbor}}$  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  $n_{\text{neighbor}}$  は, 計算には使用されません. 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  $t, E_{\text{tot,init}}, E_{\text{kin,init}}, E_{\text{sun,init}}, E_{\text{planet,init}}, \Delta E_{\text{init}}$  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, 初期条件ファイルにヘッダがなく, 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,  $E_{\text{tot,init}}, E_{\text{kin,init}}, E_{\text{sun,init}}, E_{\text{planet,init}}$  は初期条件ファイルから計算された値を設定します. 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.

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.

## 4.2 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。

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

$n_{\text{largecluster}}$ ,  $n_{\text{cluster}}$ ,  $n_{\text{isoparticle}}$  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。

## 4.3 collision.dat

collision.dat には、衝突が起こるごとに衝突の情報が以下の順に出力されます。

$$t \quad \text{ID}_{\text{imp}} \quad \text{ID}_{\text{tar}} \quad n_{\text{frag}} \quad \text{ID}_{\text{frag}} \quad m_{\text{imp}} \quad m_{\text{tar}} \quad m_{\text{frag}} \quad r_{\text{imp}} \quad v_{\text{imp}} \quad \theta_{\text{imp}} \quad \text{Hit\&Run} \quad \Delta r_g/r_g \quad \Delta v_g/v_g$$

$t$  は衝突が起こった時刻,  $\text{ID}$ ,  $m$  はそれぞれ粒子の ID と質量で,  $\text{imp}$ ,  $\text{tar}$  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。  $n_{\text{frag}}$ ,  $\text{ID}_{\text{frag}}$ ,  $m_{\text{frag}}$  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  $\text{ID}_{\text{frag}}$  から  $\text{ID}_{\text{frag}} + n_{\text{frag}} - 1$  までとなります。  $r_{\text{imp}}$ ,  $v_{\text{imp}}$ ,  $\theta_{\text{imp}}$  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。  $r_g$ ,  $v_g$  はインパクター粒子, ターゲット粒子, 破片粒子の系の重心位置, 重心速度で,  $\Delta r_g$ ,  $\Delta v_g$  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。衝突モデルの変更については §5 を参照ください。

## 4.4 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。

## 4.5 remove.dat

remove.dat には, 削除された粒子の情報が以下の順に出力されます。

$$t \quad \text{ID} \quad m \quad x \quad y \quad z \quad v_x \quad v_y \quad v_z$$

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。

粒子の削除は, 1 ステップに最大 1 粒子しか行われません。

## 5 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. (2019, 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。

### 5.0.1 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 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.

以上のメンバ変数を用いて, 以下のメンバ変数の値を設定します.

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.  
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.

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.

生じた破片粒子のリストを std::vector<Tp>型で作る, 引数の pfrag に代入します.

破片粒子数 n\_frag を返します.

また, 必要であれば, 以下のメンバ関数を設定してください.

### 5.0.2 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.



### 5.0.3 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 `fout` にパラメータの値を出力します.

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