

自由エネルギー計算解析ツールモジュール仕様書

2018 年 2 月 5 日

Copyright (C) 2006-2018 Next Generation Natural Product Chemistry (N²PC)

— 目次 —

1. Umbrella_Method	4
1.1. Calc_UmbrellaEnergyH2.....	4
1.2. Calc_UmbrellaEnergyL1.....	4
1.3. Calc_UmbrellaEnergyL2.....	5
1.4. Calc_FilledPotential.....	5
1.5. Calc_UmbrellaPotential.....	6
1.6. Calc_PotentialEnergy.....	6
2. Utility	7
2.1. Input_UmbrellaFile.....	7
2.2. Output_UmbrellaFile.....	8
2.3. Output_UmbrellaAtomLine.....	9
2.4. Is_CommentLine.....	10
2.5. Read_CordTrajectory.....	10
2.6. Read_Trajectory.....	11
2.7. Write_Trajectory.....	12
2.8. Read_PDBFile.....	12
2.9. Get_AtomMass.....	13
2.10. Read_SelectTrajectory.....	14
2.11. Calc_FreeEnergy.....	15
2.12. Calc_Probability.....	16
3. Generate_NextFP	17
3.1. Generate_NextFP.....	17
4. Extract_Atom	18
4.1. Extract_Atom.....	18
5. Wham_Analysis	19
5.1. Wham_Analysis.....	19

各モジュールの仕様は以下の項目で構成する。なお、型の記述は Fortran90 の文法に従う。

項番	項目	概要
#1	名称	当該メソッドの名称を規定する。メソッドは手続きで表現する。
#2	呼び出し形式	メソッドの種別(手続き種別)と引数を規定する。
#3	引数	手続きの仮引数を規定する。
#4	戻り値	当該メソッドが関数である場合の戻り値を規定する。
#5	機能	当該メソッドの機能概要を規定する。

1. Umbrella_Method

ファイル名 : Umbrella_Method.f90

1.1. Calc_UmbrellaEnergyH2

名称 : Calc_UmbrellaEnergyH2

呼び出し形式 :

```
function Calc_UmbrellaEnergyH2(cord, totalNum, fillNum,           &
                               atomID, weight, ellipCoeff,       &
                               firstCord, secondCord) result (energy)
```

引数 :

```
real*8, intent(in), dimension(3, MAX_ATOM)::cord           ! 原子座標
integer, intent(in)::totalNum                               ! 原子数
integer, intent(in)::fillNum                               ! 関数の次数
real*8, intent(in), dimension(MAX_FILL)::weight           ! 関数の重み
integer, intent(in), dimension(MAX_ATOM)::atomID          ! 対象原子 ID
real*8, intent(in), dimension(MAX_ATOM, MAX_FILL)::ellipCoeff ! 関数の幅(楕円)
real*8, intent(in), dimension(3, MAX_ATOM, MAX_FILL)::firstCord ! 焦点 1
real*8, intent(in), dimension(3, MAX_ATOM, MAX_FILL)::secondCord ! 焦点 2
```

戻り値 :

```
real*8::energy      ! 総エネルギー
```

機能 :

調和型楕円体求心ポテンシャルを計算する。

1.2. Calc_UmbrellaEnergyL1

名称 : Calc_UmbrellaEnergyL1

呼び出し形式 :

```
function Calc_UmbrellaEnergyL1(cord, totalNum, fillNum,           &
                               atomID, weight, ellipCoeff,       &
                               firstCord, secondCord) result (energy)
```

引数 :

```
real*8, intent(in), dimension(3, MAX_ATOM)::cord           ! 原子座標
integer, intent(in)::totalNum                               ! 原子数
integer, intent(in)::fillNum                               ! 関数の次数
real*8, intent(in), dimension(MAX_FILL)::weight           ! 関数の重み
integer, intent(in), dimension(MAX_ATOM)::atomID          ! 対象原子 ID
real*8, intent(in), dimension(MAX_ATOM, MAX_FILL)::ellipCoeff ! 関数の幅(球)
real*8, intent(in), dimension(3, MAX_ATOM, MAX_FILL)::firstCord ! 球の中心座標
real*8, intent(in), dimension(3, MAX_ATOM, MAX_FILL)::secondCord ! 未使用
```

戻り値 :

```
real*8::energy      ! 総エネルギー
```

機能 :

線型球体求心ポテンシャルを計算する。

1.3. Calc_UmbrellaEnergyL2

名称 : Calc_UmbrellaEnergyL2

呼び出し形式 :

```
function Calc_UmbrellaEnergyL2(cord, totalNum, fillNum,      &
                               atomID, weight, ellipCoeff,   &
                               firstCord, secondCord) result (energy)
```

引数 :

```
real*8, intent(in), dimension(3, MAX_ATOM)::cord           ! 原子座標
integer, intent(in)::totalNum                               ! 原子数
integer, intent(in)::fillNum                               ! 関数の次数
real*8, intent(in), dimension(MAX_FILL)::weight            ! 関数の重み
integer, intent(in), dimension(MAX_ATOM)::atomID           ! 対象原子 ID
real*8, intent(in), dimension(MAX_ATOM, MAX_FILL)::ellipCoeff ! 関数の幅(楕円)
real*8, intent(in), dimension(3, MAX_ATOM, MAX_FILL)::firstCord ! 焦点 1
real*8, intent(in), dimension(3, MAX_ATOM, MAX_FILL)::secondCord ! 焦点 2
```

戻り値 :

```
real*8::energy      ! 総エネルギー
```

機能 :

線型楕円体求心ポテンシャルを計算する。

1.4. Calc_FilledPotential

名称 : Calc_FilledPotential

呼び出し形式 :

```
function Calc_FilledPotential(cord, totalNum, fillNum,      &
                               atomID, weight, ellipCoeff,   &
                               firstCord, secondCord) result (energy)
```

引数 :

```
real*8, intent(in), dimension(3, MAX_ATOM)::cord           ! 原子座標
integer, intent(in)::totalNum                               ! 原子数
integer, intent(in)::fillNum                               ! 関数の次数
real*8, intent(in), dimension(MAX_FILL)::weight            ! 関数の重み
integer, intent(in), dimension(MAX_ATOM)::atomID           ! 対象原子 ID
real*8, intent(in), dimension(MAX_ATOM, MAX_FILL)::ellipCoeff ! 関数の幅(球)
real*8, intent(in), dimension(3, MAX_ATOM, MAX_FILL)::firstCord ! 球の中心座標
real*8, intent(in), dimension(3, MAX_ATOM, MAX_FILL)::secondCord ! 未使用
```

戻り値 :

```
real*8::energy      ! 総エネルギー
```

機能 :

球体反発ポテンシャルを計算する。

1.5. Calc_UmbrellaPotential

名称 : Calc_UmbrellaPotential

呼び出し形式 :

```
function Calc_UmbrellaPotential(cord, totalNum, fillNum, &
                               atomID, weight, ellipCoeff, &
                               firstCord, secondCord) result (energy)
```

引数 :

```
real*8, intent(in), dimension(3, MAX_ATOM)::cord      ! 原子座標
integer, intent(in)::totalNum                        ! 原子数
integer, intent(in)::fillNum                         ! 関数の次数
real*8, intent(in), dimension(MAX_FILL)::weight      ! 関数の重み
integer, intent(in), dimension(MAX_ATOM)::atomID     ! 対象原子 ID
real*8, intent(in), dimension(MAX_ATOM, MAX_FILL)::ellipCoeff ! 関数の幅(球)
real*8, intent(in), dimension(3, MAX_ATOM, MAX_FILL)::firstCord ! 球の中心座標
real*8, intent(in), dimension(3, MAX_ATOM, MAX_FILL)::secondCord ! 未使用
```

戻り値 :

```
real*8::energy      ! 総エネルギー
```

機能 :

調和型球体求心ポテンシャルを計算する。

1.6. Calc_PotentialEnergy

名称 : Calc_PotentialEnergy

呼び出し形式 :

```
function Calc_PotentialEnergy(cord, totalNum, fillNum, type, &
                              atomID, weight, ellipCoeff, &
                              firstCord, secondCord) result (energy)
```

引数 :

```
real*8, intent(in), dimension(3, MAX_ATOM)::cord      ! 原子座標
integer, intent(in)::totalNum                        ! 原子数
integer, intent(in)::fillNum                         ! 関数の次数
real*8, intent(in), dimension(MAX_FILL)::weight      ! 関数の重み
integer, intent(in), dimension(MAX_ATOM)::atomID     ! 対象原子 ID
integer, intent(in)::type                            ! 関数種別
real*8, intent(in), dimension(MAX_ATOM, MAX_FILL)::ellipCoeff ! 関数の幅
real*8, intent(in), dimension(3, MAX_ATOM, MAX_FILL)::firstCord ! 焦点 1
real*8, intent(in), dimension(3, MAX_ATOM, MAX_FILL)::secondCord ! 焦点 2
```

戻り値 :

```
real*8::energy      ! 総エネルギー
```

機能 :

入力した関数の種別に従い、各エネルギー計算処理を呼び出す。

2. Utility

ファイル名 : Utility.f90

2.1. Input_UmbrellaFile

名称 : Input_UmbrellaFile

呼び出し形式 :

```
subroutine Input_UmbrellaFile(umbFile,           &
                             firstCord, secondCord, ellipCoeff,   &
                             weight, atomID, fillNum, totalNum)
```

引数 :

```
character*80, intent(in)::umbFile      ! 入力ファイル名
real*8, intent(out), dimension(3, MAX_ATOM, MAX_FILL, MAX_TYPE)::firstCord
                                         ! 焦点 1 または球の中心座標
real*8, intent(out), dimension(3, MAX_ATOM, MAX_FILL, MAX_TYPE)::secondCord
                                         ! 焦点 2 の座標
real*8, intent(out), dimension(MAX_ATOM, MAX_FILL, MAX_TYPE)::ellipCoeff
                                         ! ポテンシャルの幅
real*8, intent(out), dimension(MAX_FILL, MAX_TYPE)::weight
                                         ! ポテンシャルの深さ
integer, intent(out), dimension(MAX_ATOM, MAX_TYPE)::atomID
                                         ! 対象原子 ID
integer, intent(out), dimension(MAX_TYPE)::fillNum
                                         ! 関数の次数
integer, intent(out), dimension(MAX_TYPE)::totalNum
                                         ! 対象原子数
```

戻り値 :

なし

機能 :

UmbrellaPotential 指定ファイルを読み込む、内部情報に変換する。

2.2. Output_UmbrellaFile

名称 : Output_UmbrellaFile

呼び出し形式 :

```
subroutine Output_UmbrellaFile(outFile,                                &
                               firstCord, secondCord, ellipCoeff,    &
                               weight, atomID, fillNum, totalNum)
```

引数 :

```
character*80, intent(in)::outFile          ! 出力ファイル名
real*8, intent(in), dimension(3, MAX_ATOM, MAX_FILL, MAX_TYPE)::firstCord
! 焦点 1 または球の中心座標
real*8, intent(in), dimension(3, MAX_ATOM, MAX_FILL, MAX_TYPE)::secondCord
! 焦点 2 の座標
real*8, intent(in), dimension(MAX_ATOM, MAX_FILL, MAX_TYPE)::ellipCoeff
! ポテンシャルの幅
real*8, intent(in), dimension(MAX_FILL, MAX_TYPE)::weight
! ポテンシャルの深さ
integer, intent(in), dimension(MAX_ATOM, MAX_TYPE)::atomID
! 対象原子 ID
integer, intent(in), dimension(MAX_TYPE)::fillNum
! 関数の次数
integer, intent(in), dimension(MAX_TYPE)::totalNum
! 対象原子数
```

戻り値 :

なし

機能 :

UmbrellaPotential の内部情報をファイルに出力する。

2.3. Output_UmbrellaAtomLine

名称 : Output_UmbrellaAtomLine

呼び出し形式 :

```
subroutine Output_UmbrellaAtomLine(firstCord, secondCord, num, fillDim, type, &
                                   atomID, unit)
```

引数 :

```
real*8, intent(in), dimension(3, MAX_ATOM, MAX_FILL, MAX_TYPE)::firstCode
    ! 焦点 1 または球の中心座標
real*8, intent(in), dimension(3, MAX_ATOM, MAX_FILL, MAX_TYPE)::secondCord
    ! 焦点 2 の座標
integer, intent(in), dimension(MAX_ATOM, MAX_TYPE)::atomID
    ! 対象原子 ID 配列
integer, intent(in)::num
    ! 対象原子 ID
integer, intent(in)::fillDim
    ! 関数の対象次元
integer, intent(in)::type
    ! 関数の型
integer, intent(in)::unit
    ! 論理出力番号
```

戻り値 :

なし

機能 :

当該原子の中心、あるいは焦点座標を出力する。

2.4. Is_CommentLine

名称 : Is_CommentLine

呼び出し形式 :

```
function Is_CommentLine(line) result (flag)
```

引数 :

```
character(*), intent(in)::line      ! 入力文字列
```

戻り値 :

```
logical::flag                       ! コメントフラグ
```

機能 :

当該行がコメント行(あるいは空行)かどうかを判定する。

2.5. Read_CordTrajectory

名称 : Read_CordTrajectory

呼び出し形式 :

```
subroutine Read_CordTrajectory(cord, atomNum, format, unit)
```

引数 :

```
integer, intent(in)::atomNum        ! 原子数
```

```
character(*), intent(in)::format    ! トラジェクトリファイルのフォーマット (s | d)
```

```
integer, intent(in)::unit           ! 論理装置番号
```

```
real*8, dimension(3, MAX_ATOM)::cord ! 原子座標 (A)
```

戻り値 :

なし

機能 :

指定された装置番号から、座標データのみを取り出す。

2.6. Read_Trajectory

名称 : Read_CordTrajectory

呼び出し形式 :

```
subroutine Read_Trajectory(loopNum, simulateTime, cpuTime,          &
                          totalEnergy, kineticEnergy, temp,      &
                          potentialEnergy, rmsf, num15vdw, num15hyd, &
                          rmsd, cord, atomNum, format, unit)
```

引数 :

```
integer, intent(in)::atomNum          ! 原子数
character(*), intent(in)::format      ! トラジェクトリファイルフォーマット (s | d)
integer, intent(in)::unit             ! 論理装置番号
integer, intent(out)::loopNum         ! MD ループ数
integer, intent(out)::num15vdw        ! van der Waals 相互作用数
integer, intent(out)::num15hyd        ! hydrogen-bond i 相互作用数
real*8, intent(out)::simulateTime     ! シミュレート時間(fs)
real*8, intent(out)::cpuTime          ! cpu 時間(s)
real*8, intent(out)::totalEnergy       ! 総エネルギー (KCAL/MOL)
real*8, intent(out)::kineticEnergy     ! 運動エネルギー (KCAL/MOL)
real*8, intent(out)::temp             ! 温度 (K)
real*8, intent(out)::potentialEnergy   ! ポテンシャルエネルギー (KCAL/MOL)
real*8, intent(out)::rmsf             ! RMSF (KCAL/MOL/A)
real*8, intent(out)::rmsd             ! RMSD (A)
real*8, dimension(3, MAX_ATOM)::cord  ! 原子座標 (A)
```

戻り値 :

なし

機能 :

指定された装置番号のファイルから、トラジェクトリデータを取り出す。

2.7. Write_Trajectory

名称 : Write_Trajectory

呼び出し形式 :

```
subroutine Write_Trajectory(loopNum, simulateTime, cpuTime,          &
                           totalEnergy, kineticEnergy, temp,       &
                           potentialEnergy, rmsf, num15vdw, num15hyd, &
                           rmsd, cord, atomNum, format, unit)
```

引数 :

```
integer, intent(in)::atomNum          ! 原子数
character(*), intent(in)::format      ! トラジェクトリファイルフォーマット (s | d)
integer, intent(in)::unit             ! 論理装置番号
integer, intent(in)::loopNum          ! MD ループ数
integer, intent(in)::num15vdw        ! van der Waals 相互作用数
integer, intent(in)::num15hyd        ! hydrogen-bond i 相互作用数
real*8, intent(in)::simulateTime     ! シミュレート時間(fs)
real*8, intent(in)::cpuTime          ! cpu 時間(s)
real*8, intent(in)::totalEnergy      ! 総エネルギー (KCAL/MOL)
real*8, intent(in)::kineticEnergy    ! 運動エネルギー (KCAL/MOL)
real*8, intent(in)::temp             ! 温度 (K)
real*8, intent(in)::potentialEnergy  ! ポテンシャルエネルギー (KCAL/MOL)
real*8, intent(in)::rmsf             ! RMSF (KCAL/MOL/A)
real*8, intent(in)::rmsd             ! RMSD (A)
real*8, dimension(3, MAX_ATOM)::cord ! 原子座標 (A)
```

戻り値 :

なし

機能 :

指定された装置番号のファイルに、トラジェクトリデータを書き込む。

2.8. Read_PDBFile

名称 : Read_PDBFile

呼び出し形式 :

```
subroutine Read_PDBFile(unit, atomNum, mass, cord, display)
```

引数 :

```
integer, intent(in)::unit             ! 論理装置番号
integer, intent(out)::atomNum         ! 原子数
real*8, intent(out), dimension(MAX_ATOM)::mass ! 質量
real*8, intent(out), dimension(3, MAX_ATOM)::cord ! 原子座標
logical, intent(in)::display         ! 入力行表示フラグ
```

戻り値 :

なし

機能 :

指定された装置番号のファイルから PDB ファイルの座標を読み込み、各原子の質量を割り当てる。

2.9. Get_AtomMass

名称 : Get_AtomMass

呼び出し形式 :

```
function Get_AtomMass(name) result (mass)
```

引数 :

```
character*2, intent(in)::name      ! 原子名
```

戻り値 :

```
real*8::mass                       ! 質量
```

機能 :

原子の質量を求める。

2.10. Read_SelectTrajectory

名称 : Read_SelectTrajectory

呼び出し形式 :

```
subroutine Read_SelectTrajectory(cord, atomNum, equivNum, sampleNum,      &
                                trjNum, inTrjName, inTrjUnit, format)
```

引数 :

```
real*8, intent(out), dimension(:, :, :, :)::cord    ! 原子座標
integer, intent(in)::atomNum                        ! 原子数
integer, intent(in)::equivNum                      ! スキップするループ回数
integer, intent(in)::sampleNum                    ! サンプリングループ回数
integer, intent(in)::trjNum                       ! トラジェクトリファイル数
integer, intent(in)::inTrjUnit                    ! 論理装置番号
character*80, intent(in), dimension(:)::inTrjName ! トラジェクトリファイル名(複数)
character*1, intent(in)::format                   ! トラジェクトリフォーマット(s/d)
```

戻り値 :

なし

機能 :

トラジェクトリファイルから指定された範囲の MD 回数のトラジェクトリの座標を読み込む。

2.11. Calc_FreeEnergy

名称 : Calc_FreeEnergy

呼び出し形式 :

```
subroutine Calc_FreeEnergy (freeEnergy, rmsd, trjNum, sampleNum, equivNum, &
                           cord, totalNum, fillNum, atomID, weight,      &
                           ellipCoeff, firstCord, secondCord, type,      &
                           converge, inTrjUnit, inTrjName, useMemory, format)
```

引数 :

```
real*8, intent(inout), dimension(:)::freeEnergy    ! 自由エネルギーの exp 値
real*8, intent(inout), dimension(:)::rmsd          ! root mean square distance
integer, intent(in)::trjNum                        ! トラジェクトリファイル数
integer, intent(in)::sampleNum                    ! サンプリンググループ数
integer, intent(in)::equivNum                     ! スキップするループ回数
real*8, intent(in), dimension(:, :, :, :)::cord    ! 原子座標 (xyz, 原子, ループ回数, ファイル)
integer, intent(in), dimension(MAX_TYPE)::totalNum ! 原子数
integer, intent(in), dimension(MAX_TYPE)::fillNum  ! 関数の次数
integer, intent(in)::type                          ! 関数の型
integer, intent(in), dimension(MAX_ATOM, MAX_TYPE)::atomID ! 原子 ID
real*8, intent(in), dimension(MAX_FILL, MAX_TYPE)::weight ! ポテンシャルの重み
real*8, intent(in), dimension(MAX_ATOM, MAX_FILL, MAX_TYPE)::ellipCoeff
                                                    ! ポテンシャルの幅
real*8, intent(in), dimension(3, MAX_ATOM, MAX_FILL, MAX_TYPE)::firstCord
                                                    ! 焦点 1、あるいは中心
real*8, intent(in), dimension(3, MAX_ATOM, MAX_FILL, MAX_TYPE)::secondCord
                                                    ! 焦点 2
integer, intent(in)::converge                      ! 収束ループ回数
integer, intent(in)::inTrjUnit                     ! トラジェクトリファイルの論理装置番号
character*80, intent(in), dimension(:)::inTrjName ! トラジェクトリファイル名
character*1::format                                ! トラジェクトリのフォーマット
logical, intent(in)::useMemory                     ! メモリ使用フラグ
```

戻り値 :

なし

機能 :

各トラジェクトリに対し、Umbrella ポテンシャルの自由エネルギー計算を行う。

2.12. Calc_Probability

名称 : Calc_Probability

呼び出し形式 :

```
subroutine Calc_Probability(freeEnergy, rmsd, trjNum, sampleNum, equivNum, &
                           cord, totalNum, fillNum, atomID, weight,      &
                           ellipCoeff, firstCord, secondCord, type,      &
                           radius, inTrjUnit, inTrjName, useMemory, format)
```

引数 :

```
real*8, intent(inout), dimension(:)::freeEnergy    ! 自由エネルギーの exp 値
real*8, intent(inout), dimension(:)::rmsd          ! root mean square distance
integer, intent(in)::trjNum                        ! トラジェクトリファイル数
integer, intent(in)::sampleNum                    ! サンプルンググループ数
integer, intent(in)::equivNum                     ! スキップするループ回数
real*8, intent(in), dimension(:, :, :, :)::cord   ! 原子座標 (xyz, 原子, ループ回数, ファイル)

integer, intent(in), dimension(MAX_TYPE)::totalNum ! 原子数
integer, intent(in), dimension(MAX_TYPE)::fillNum  ! 関数の次数
integer, intent(in)::type                          ! 関数の型
integer, intent(in), dimension(MAX_ATOM, MAX_TYPE)::atomID ! 原子 ID
real*8, intent(in), dimension(MAX_FILL, MAX_TYPE)::weight ! ポテンシャルの重み
real*8, intent(in), dimension(MAX_ATOM, MAX_FILL, MAX_TYPE)::ellipCoeff
                                                    ! ポテンシャルの幅
real*8, intent(in), dimension(3, MAX_ATOM, MAX_FILL, MAX_TYPE)::firstCord
                                                    ! 焦点 1、あるいは中心
real*8, intent(in), dimension(3, MAX_ATOM, MAX_FILL, MAX_TYPE)::secondCord
                                                    ! 焦点 2

real*8, intent(in)::radius                        ! 平均座標の範囲判定値
integer, intent(in)::inTrjUnit                    ! トラジェクトリファイルの論理装置番号
character*80, intent(in), dimension(:)::inTrjName ! トラジェクトリファイル名
character*1::format                               ! トラジェクトリのフォーマット
logical, intent(in)::useMemory                    ! メモリ使用フラグ
```

戻り値 :

なし

機能 :

各トラジェクトリに対し、関数の中心に付近に原子座標が存在する平均確率を計算し、標準出力に出力する。

3. Generate_NextFP

ファイル名 : Generate_NextFP.f90

3.1. Generate_NextFP

名称 : Generate_NextFP

呼び出し形式 :

```
program Generate_NextFP
```

入力 :

(1) 制御ファイル

(1-1) 前回の MD での Umbrella Potential 指定ファイル名

(1-2) 出力 Umbrella Potential 指定ファイル名

(1-3) 初期座標 PDB ファイル名

(1-4) 前回の MD での座標トラジェクトリファイル名

(1-5) 座標トラジェクトリの読み飛ばし回数

(1-6) 座標トラジェクトリ読み込み回数

(1-7) 座標トラジェクトリファイル形式 ("s"ingle | "d"ouble)

(1-8) PDB ファイルの画面表示オプション("y"es | "n"o)

(1-9) 求心関数の種別

(1-10) 温度

(1-11) ガウス型反発関数の高さ

(1-12) ガウス型ポテンシャル中心座標の更新間隔を制御する範囲

(1-13) ガウス型反発関数の幅

(1-14) 目標とする最終座標

(2) 初期座標 PDB ファイル

(3) 前回の MD での座標トラジェクトリファイル

(4) 前回の MD 入力の Umbrella Potential 指定ファイル

出力 :

(1) Umbrella Potential ファイル

機能 :

入力した UmbrellaPotential のガウス型反発関数の次数を追加し、また、追加した次数に対応する求心関数を付加する。

追加した反発関数の中心座標は初期座標から 0.2 Å 程度の幅で、最終座標に近づくように設定し、楕円体の場合は焦点の中心が最終座標に近づくように設定する。

4. Extract_Atom

ファイル名 : Extract_Atom.f90

4.1. Extract_Atom

名称 : Extract_Atom

呼び出し形式 :

program Extract_Atom

入力 :

(1) 制御ファイル

(1-1) Umbrella Potential 指定ファイル名

(1-2) 座標トラジェクトリファイルの原子数

(1-3) 座標トラジェクトリファイル数

(1-4) 入力、出力座標トラジェクトリファイル名

(1-5) 座標トラジェクトリの読み飛ばし回数

(1-6) 座標トラジェクトリ読み込み回数

(1-7) 座標トラジェクトリファイル形式 ("single" | "double")

(2) 座標トラジェクトリファイル群

(3) 前回の MD 入力の Umbrella Potential ファイル

出力 :

(1) Umbrella Potential 対象原子のトラジェクトリファイル

機能 :

トラジェクトリファイルと Umbrella Potential 指定ファイルを読み込み、指定された範囲の Umbrella Potential 対象原子のトラジェクトリファイルを生成する。

5. Wham_Analysis

ファイル名 : Wham_Analysis.f90

5.1. Wham_Analysis

名称 : Wham_Analysis

呼び出し形式 :

program Wham_Analysis

入力 :

(1) 制御ファイル

(1-1) Umbrella Potential 指定ファイル名

(1-2) 座標トラジェクトリファイル数

(1-3) 入力座標トラジェクトリファイル名

(1-4) 座標トラジェクトリの読み飛ばし回数

(1-5) 座標トラジェクトリ読み込み回数

(1-6) 座標トラジェクトリファイル形式 ("single" | "double")

(1-7) 自由エネルギー計算の平均値計算のサンプリング距離

(1-8) WHAM 解析の収束ループ数

(1-7) WHAM 解析時のメモリ指定 ("memory" | "speed")

(2) 座標トラジェクトリファイル群

(3) 前回の MD 入力の Umbrella Potential ファイル

出力 :

(1) 自由エネルギー計算結果

機能 :

入力したトラジェクトリファイルの座標に対し、Umbrella Potential 指定ファイルに記述しているポテンシャルを計算、各トラジェクトリに対する自由エネルギーと座標の確率を計算する。

—以上—