
Celeste Documentation

Release 0.35.s.4

Kota Kasahara

July 16, 2015

CONTENTS

| | | |
|----------|-----------------------------------------|-----------|
| 1 | Introduction | 3 |
| 1.1 | About Celeste | 3 |
| 1.2 | Developers | 3 |
| 2 | Installation | 5 |
| 2.1 | Summary | 5 |
| 2.2 | Without neighbor search | 5 |
| 2.3 | Acceleration with GPU | 5 |
| 2.4 | Celeste Toolkit | 6 |
| 3 | System Preparation | 7 |
| 3.1 | Input files | 7 |
| 3.2 | Execute | 8 |
| 4 | In/Out Files | 9 |
| 4.1 | 入力ファイル | 9 |
| 4.2 | 出力ファイル | 12 |
| 5 | Analysis | 13 |
| 5.1 | 計算ログ | 13 |
| 5.2 | トラジェクトリの可視化 | 13 |
| 6 | Samples | 15 |
| 6.1 | サンプルファイル | 15 |
| 7 | 二体力計算ルーチンに関するメモ | 17 |
| 7.1 | 二体力計算に至るまでのながれ | 17 |
| 7.2 | Neighbor search なしでの計算ルーチンの概要 | 17 |
| 7.3 | 結合原子ペアの除外 | 18 |
| 7.4 | 二体力の計算 | 18 |
| 8 | Indices and tables | 21 |
| | Bibliography | 23 |

Contents:

INTRODUCTION

Author Kota Kasahara

1.1 About Celeste

“Celeste” is the code name of the molecular dynamics (MD) simulation software. For the Celeste has several unique features, such as, the zero-dipole summation method [\[Ref:ZD\]](#) and the virtual-system coupled multi-canonical MD method [Ref: TTP]_.

The current version of Celeste can perform,

1. MD simulations on NVE ensemble
2. MD simulations on NVT ensemble (The velocity rescaling, or Hoover-Evans thermostat)
3. MD simulations on Multi-canonical ensemble, with the Virtual-system coupled McMD method.
4. Applying constraints with SHAKE algorithm
5. Calculations of electrostatic potentials based on the zero-dipole summation method.
6. Calculations of pairwise potentials on GPGPU (powered by CUDA 6.0, Computer Capability 3.5 is required).

1.2 Developers

- KASAHARA, Kota, Osaka Univ., Japan
- MASHIMO, Tadaaki, AIST, Japan
- Gert-jan BEKKER, Osaka Univ., Japan
- Ma BENSON, UC Berkeley, US
- HARIYAMA, Masanori, Tohoku Univ., Japan
- Hasitha M. WAIDYASOORIYA, Tohoku Univ., Japan

INSTALLATION

Author Kota Kasahara

2.1 Summary

Celeste is written in C++ language. Build the binary by using *make* command.

There are three compile options:

1. For CPU
2. For CPU, without the neighbor search algorithm (not efficient)
3. For CPU with GPU (Cuda)

They can be switched by changing the EXECUTABLE variable in Makefile:

```
EXECUTABLE = celeste
#EXECUTABLE = celeste_wons    #without neighbor search
#EXECUTABLE = celeste_gpu     #CUDA
```

2.2 Without neighbor search

The neighbor search is effective for fast calculation, but the implementation is too complicated. For debugging or tests, the computation without the neighbor search is available.

```
EXECUTABLE = celeste_wons
```

2.3 Acceleration with GPU

For GPU mode, NVIDIA GPU Compute capability 3.5 or later is required.

The path to the CUDA library should be specified as CUDALOC variable in Makefile.

```
EXECUTABLE = celeste_gpu
CUDALOC = /opt/cuda/5.5/lib64
```

2.4 Celeste Toolkit

CelesteToolkit is a series of scripts for pre- and post-processing for MD simulations with Celeste. It requires *python2.7.x* and *numpy* library. In this manual, it is supposed that all files of *CelesteToolkit* are contained in the directory specified in the environmental variable `{CELESTETK}`. This path should be added in `{PYTHONPATH}`.

bash:

```
export CELESTETK="${HOME}/celeste/toolkit"
export PYTHONPATH=${CELESTETK}:${PYTHONPATH}
```

csh:

```
setenv CELESTETK "${HOME}/celeste/toolkit"
setenv PYTHONPATH ${CELESTETK}:${PYTHONPATH}
```

SYSTEM PREPARATION

Author Kota Kasahara

3.1 Input files

The input files required for MD simulations by Celeste are compatible with myPresto/Psygene. Celeste requires

1. Restart file (including the initial coordinates and velocities)
2. Topology file (generated by Tplgene program)
3. Shake file (generated by SHAKEinp program)

In addition, two configuration files are required.

4. System configuration file (system.cfg)
5. Simulation configuration file (md.cfg)

CelesteToolkit generates a binary file by combining some of the input files.

```
python2.7 ${CELESTETK}/mdinput_generator.py -i system.cfg -o system.cls
```

Example of system.cfg file is shown below:

system.cfg:

```
--fn-i-tpl          etl.tpl      ; Topology file
--fn-i-initial-pdb  etl.pdb      ; Structure file in .pdb format
--fn-i-restart      etl.restart  ; Restart file
--cell-x            61.2425     ; Specifying the each length of cell-axes
--cell-y            61.2425
--cell-z            61.2425
--fn-i-shake        system.shk   ; SHAKE file
;--fn-i-ttp-v-mcmd-inp      ttp_v_mcmd.inp ; For V-McMD
;--fn-i-ttp-v-mcmd-initial  start.vert    ; For V-McMD
```

md_input.cfg:

```
--mode              md          ; Only the keyword "md" is accepted
--integrator         leapfrog-presto ; "leapfrog-presto" or "zhang"
--thermostat         scaling      ; Thermostat algorithm
                        ; "none", "scaling", "hoover-evans"
--cutoff             12.0         ; Cutoff distance in angstrom
--n-steps            10           ; The number of steps to be calculated
--time-step          2.0         ; Integration time step in fs
--electrostatic       zero-dipole ; Only "zero-dipole" is accepted
```

```
--ele-alpha          0          ; Dumping factor for ZD
                        ;   only 0 is OK for GPU version
--temperature        300        ; The target temperature
--print-interval-log  1          ; Interval steps for printing logs
--print-interval-coord 1          ; Interval steps for output the trajectory
--fn-o-coord          trpc.trr    ; File name for the output trajectory
--format-o-coord       gromacs    ; File format for the output trajectory
                        ; "presto" or "gromacs"
;--fn-o-log           et1.log     ; Not used in the current version
;--fn-o-energy         et1.ene     ; Not used in the current version
--nsgrid-cutoff       13.0       ; Neighbor search cutoff radius (angstrom)
--nsgrid-update-intvl 50         ; Interval time steps for update the neighbor search grid

; Expanded ensemble
;--expanded-ensemble  v-mcmd      ; "none", or "v-mcmd"
;--fn-o-vmcmd-log      ttp_v_mcmd.out ; Output filename
;--fn-o-expand-lambda  mule.ene    ; Output filename
;--print-interval-expand-lambda 1    ; Interval steps for the output
;--format-o-expand-lambda  ascii    ; "ascii", or "binary"
```

3.2 Execute

Celeste executes a MD simulation with two configuration files provided as command line arguments. The current version of Celeste output log messages to the standard output. Redirecting to a log file is recommended.

```
celeste --inp et1.cls --cfg md_input.cfg > log.txt
```

IN/OUT FILES

Author Kota Kasahara

4.1 入力ファイル

1. システム設定 .cfg
2. シミュレーション設定 .cfg
3. 構造情報 .pdb
4. 初期構造/速度設定 .restart
5. トポロジー設定 .tpl
6. SHAKE 設定 .shk
7. V-McMD 設定 (2 種類) .inp, .vert
8. システム情報 .cls

4.1.1 システム定義ファイル

mdinput_generator.py 用の入力ファイルです。これらの情報を元にシステム情報ファイル .cls を生成します。1 行につき 1 エントリーで、各エントリーではキーワードと値のセットで設定を入力します。

- **--fn-i-tpl** et1.tpl
 - トポロジー設定ファイル名を相対または絶対パスで指定します。
 - ファイルの詳細については後述します。
- **--fn-i-initial-pdb** et1.pdb
 - 構造情報を含む.pdb ファイルへのパスを指定します。
 - ファイルの詳細については後述します。
- **--fn-i-restart** et1.restart
 - 初期座標および速度を含む restart ファイルへのパスを指定します。
 - ファイルの詳細については後述します。
- **--cell-x** 61.2425
- **--cell-y** 61.2425
- **--cell-z** 61.2425

- 周期境界セルの X,Y,Z 軸のセル長を angstrome 単位で指定します。
- **--fn-i-shake** system.s
 - SHAKE 設定ファイルへのパスを指定します。
- **--fn-i-ttp-v-mcmd-inp** ttp_v_mcmd.inp
- **--fn-i-ttp-v-mcmd-initial** start.vert
 - V-McMD ファイルへのパスを指定します。

4.1.2 シミュレーション設定ファイル

Celeste 用入力ファイルです。このファイルと、システム情報ファイル .cls をセットで Celeste に入力します。

md_input.cfg:

| | |
|----------|----|
| * --mode | md |
|----------|----|

- 現状 “md” のみを許可
- **--integrator** leapfrog-presto
 - 現状下記 2 つのみ許可* leapfrog-presto
 - * Psygene と同様の積分アルゴリズム
 - * 熱浴なし、”scaling” 熱浴、SHAKE に対応
 - * zhang * Zhang [Zhang1997] の積分アルゴリズム* “hoover-evans” 熱浴のみ可。SHAKE 不可。
- **--thermostat** scaling
 - 現状下記 3 つのみ許可* none
 - * 熱浴無し。マイクロカノニカルアンサンブル
 - * scaling * 温度スケーリング
 - * hoover-evans * Hoover-Evans 熱浴。--integrator は zhang のみ対応。
- **--cutoff** 12.0
 - 非結合性ペアポテンシャルのカットオフ距離
- **--n-steps** 10
 - 計算ステップ数
- **--time-step** 2.0
 - 積分時間
- **--electrostatic** zero-dipole
 - 現状 “zero-dipole” のみ可
- **--ele-alpha** 0.0
 - Zero-multipole summation 法のダンピング係数
 - 現状、GPU 版は 0.0 のみ対応
- **--temperature** 300
 - 熱浴使用時の温度
- **--print-interval-log** 1

- ログの出力間隔
- **--print-interval-coord** 1
 - 座標情報の出力間隔
- **--fn-o-coord** et1.trr
 - 座標情報の出力先ファイル名
- **--format-o-coord** gromacs
 - 座標情報の出力形式
 - gromacs
 - presto
- **--fn-o-log** et1.log
 - 未実装
- **--fn-o-energy** et1.ene
 - 未実装
- **--nsgrid-cutoff** 13.0
 - 近傍探索用カットオフ
- **--nsgrid-update-intvl** 50
 - 近傍探索の間隔
- **--expanded-ensemble** v-mcmt
 - none * 拡張アンサンブルを使用しない
 - v-mcmt * TTP-V-McMD 法 [Higo2013]_
- **--fn-o-vmcmt-log** ttp_v_mcmc.out
 - V-McMD の仮想状態トラジェクトリの出力先ファイル名
- **--fn-o-expand-lambda** mule.ene
 - 拡張アンサンブルの反応座標トラジェクトリの出力先ファイル名
- **--print-interval-expand-lambda** 1
 - 反応座標トラジェクトリの出力間隔
- **--format-o-expand-lambda** ascii
 - 反応座標トラジェクトリの出力形式
 - ascii
 - binary

4.1.3 初期構造/速度設定ファイル

myPresto/Psygene の Restart file に準拠。附属の presto_generate_velocities.py スクリプトでも生成出来る。

4.1.4 トポロジー設定ファイル

myPresto/Psygene に準拠。myPresto パッケージ附属の tplgene プログラムで生成出来る。

4.1.5 SHAKE 設定ファイル

myPresto/Psygene に準拠。myPresto パッケージ附属の SHAKEinp プログラムで生成出来る。

4.1.6 V-McMD 設定ファイル

myPresto/Psygene に準拠。 * -fn-i-ttp-v-mcmd-inp ttp_v_mcmd.inp * -fn-i-ttp-v-mcmd-initial start.vert

4.2 出力ファイル

1. 構造トラジェクトリファイル
2. 最終構造/速度ファイル
3. V-McMD 出力ファイル (2 種類)

ANALYSIS

Author Kota Kasahara

5.1 計算ログ

エネルギー値が標準出力に打ち出されます。Microcanonical ensemble では理論上 Total energy は一定となります。Total energy のドリフトが大きい場合は、

1. -nsgrid-cutoff を増やす
2. -nsgrid-update-intvl を減らす
3. -cutoff を増やす

などの対策が必要となります。

```
Step:      0      Time:      0.0000
Total:     -7.4102976562e+04
Potential: -8.7917343750e+04      Kinetic:  1.3814368164e+04
Bond:      2.2974748345e+02      Angle:    9.5691961023e+01
Torsion:   2.9869403994e+02      Improper: 2.9828235618e+00
14-VDW:    1.0501907707e+02      14-El:   1.8181411150e+03
VDW:       1.5527498751e+04      Ele:     -1.0599511706e+05

Step:  100000      Time: 50000.0000
Total:     -7.4026648438e+04
Potential: -8.7759882812e+04      Kinetic:  1.3733237305e+04
Bond:      9.4730405123e+03      Angle:    2.4954034353e+02
Torsion:   3.2906814547e+02      Improper: 1.2255384929e+01
14-VDW:    1.2528642296e+02      14-El:   1.8064233115e+03
VDW:       1.7495527442e+04      Ele:     -1.1725102074e+05
```

5.2 トrajекトリの可視化

Celeste が出力するトラジェクトリは Gromacs .trr 形式で書かれています。VMD¹ などの可視化ソフトウェアで読み込んでください。

¹ <http://www.ks.uiuc.edu/Research/vmd/>

SAMPLES

Author Kota Kasahara

6.1 サンプルファイル

サンプル計算ファイルとして Endothelin-1 dimer (PDB-ID: 1t7h) を付属しています。

```
python2.7 ${CELESTETK}/mdinput_generator.py -i gen_input.cfg -o et1.cls  
celeste_gpu --inp et1.cls --cfg md_input.cfg
```


二体力計算ルーチンに関するメモ

Author Kota Kasahara

7.1 二体力計算に至るまでのながれ

Celeste では原子の座標をまず MmSystem クラスの crd 配列に読み込みます。MmSystem.crd 配列は [原子数, 3] の二次元配列で、各原子の x,y,z 座標を記録します。:

```
crd[atomid][0] = x;  
crd[atomid][1] = y;  
crd[atomid][2] = z;
```

計算の前に、crd の内容は SubBox クラスの crd へと渡されます。これは将来的にはプロセス並列を実装するためのもので、分割した空間の一部のみが SubBox へ渡されることになります。現バージョンでは空間分割を実装していないので、MmSystem.crd と SubBox.crd は同じ内容になります。但し SubBox.crd は長さが (原子数*3) の一次元配列になっており、xyzxyzxyz... の順番で値が記録されます。また SubBox.atomids 配列には MmSystem 側との原子 ID の対応を記録します。:

```
SubBox.atomids[SubBox での原子 ID] = MmSystem での原子 ID  
MmSystem.crd[SubBox.atomids[i]][0] = crd[i*3+0]
```

7.2 Neighbor search なしでの計算ルーチンの概要

Neighbor search を行わずに二体力計算を行う場合には、crd 配列に対して二重ループを回すことで全対全の二体力を計算します。

また特定の原子ペアについては二体力計算をスキップしなくてはならないため、条件分岐を行います。(後述)

具体的な二体力の計算式は ForceField クラスの calc_pairwise 関数に座標とパラメータを渡すことで実行します。原子間距離がカットオフ以内だった場合は戻り値 0 が得られますので、そのときに限り計算結果を記録します。

SubBox::calc_energy_pairwise_wo_neighborsearch:

```
int SubBox::calc_energy_pairwise_wo_neighborsearch(){  
    for(int atomid1 = 0, atomid1_3=0; atomid1 < n_atoms_exbox; atomid1++, atomid1_3+=3){  
        real crd1[3] = {crd[atomid1_3], crd[atomid1_3+1], crd[atomid1_3+2]};  
        for(int atomid2 = 0, atomid2_3=0; atomid2 < atomid1; atomid2++, atomid2_3+=3){  
            real crd2[3] = {crd[atomid2_3], crd[atomid2_3+1], crd[atomid2_3+2]};  
            bool flg=true;  
            for(int i = atomid1 * max_n_nb15off;  
                i < atomid1 * max_n_nb15off + max_n_nb15off;  
                i++){
```

```

        if(nb15off[i] == atomid2) flg = false;
    }
    if(!flg) continue;

    real_pw tmp_ene_vdw = 0.0;
    real_pw tmp_ene_ele = 0.0;
    real_fc tmp_work[3] = {0.0, 0.0, 0.0};
    real param_6term = lj_6term[atom_type[atomid1] * n_lj_types + atom_type[atomid2]];
    real param_12term = lj_12term[atom_type[atomid1] * n_lj_types + atom_type[atomid2]];

    for(int d=0; d < 3; d++){
        if(crd2[d]-crd1[d] >= pbc->L_half[d])
            crd2[d] -= pbc->L[d];
        else if(crd2[d]-crd1[d] <= -pbc->L_half[d])
            crd2[d] += pbc->L[d];
    }
    if(ff.calc_pairwise(tmp_ene_vdw, tmp_ene_ele, tmp_work,
                       crd1, crd2,
                       param_6term, param_12term,
                       charge[atomid1],
                       charge[atomid2])!=0){
        pote_vdw += tmp_ene_vdw;
        pote_ele += tmp_ene_ele;
        work[atomid1_3] += tmp_work[0];
        work[atomid1_3+1] += tmp_work[1];
        work[atomid1_3+2] += tmp_work[2];
        work[atomid2_3] -= tmp_work[0];
        work[atomid2_3+1] -= tmp_work[1];
        work[atomid2_3+2] -= tmp_work[2];
    }
}
}
return 0;
}

```

7.3 結合原子ペアの除外

結合性ポテンシャルが定義されている原子ペアについては、非結合性の二体ポテンシャルを計算しません。つまり直接共有結合している隣の原子や、共有結合を 2 つおよび 3 つ挟んだ近傍の原子ペアはスキップします。nb15off 配列にどの原子ペアでの計算をスキップするかを記録してあるので、これを随時参照して判定します。

nb15off は長さ (原子数 * max_n_nb15off) の int 型配列です。配列の $i \cdot \text{max_n_nb15off}$ 番目から max_n_nb15off 個の値は、 i 番目の原子との相互作用を計算しない原子 ID のリストになっています。 i - j ペアに対して判定を行う際は、この範囲に j が含まれていればスキップを行います。

7.4 二体力の計算

計算には原子座標の他にパラメータ charge、atom_type、lj_6term、lj_12term を利用します。charge は各原子の部分電荷を示す float 配列です。atom_type は各原子がどのタイプに属するかを指定します。原子は n_lj_types 種類 (40 程度) の原子タイプに分かれ、原子タイプのペアごとに van der Waals パラメータが決まります。lj_6term と lj_12term はペア毎のパラメータを示す配列で、それぞれ ($n_lj_types * n_lj_types$) の長さを持っています。

```
real param_6term = lj_6term[atom_type[atomid1] * n_lj_types + atom_type[atomid2]];
real param_12term = lj_12term[atom_type[atomid1] * n_lj_types + atom_type[atomid2]];
```

なお静電相互作用の計算ルーチンは ZeroMultipoleSum.cpp に実装されています。ZeroDipoleSum.cpp は現状なにもしていません。

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

BIBLIOGRAPHY

[Ref:ZD] Zero-dipole summation method

[Ref:TTP] Trivial Trajectory Parallelization, Virtual system coupled, Multicanonical MD