

2020年11月30日(月) オンライン講習会



第4回LAMMPS入門講習会

高度情報科学技術研究機構(RIST)

利用支援部 吉澤香奈子

Outline

- MateriApps LIVE!の設定
- LAMMPSについて
 - 分子動力学計算について
 - 力場(ポテンシャル)について
 - 実行時エラー
- LAMMPSのexamplesの実行
 - melt を実行する
 - ✓ 計算結果を可視化する
 - ✓ エネルギーをプロットする
 - micelle を実行する
 - ✓ 計算結果を可視化する
 - rerun を実行する
 - ✓ 動径分布関数(RDF)をプロットする
- 並列実行
- サイズを大きくした計算
- restartを行う
- HPCIにプリインストールされたアプリ情報

MateriApps LIVE! とは？

- MateriApps LIVE! ウェブサイト
<https://ma.issp.u-tokyo.ac.jp/materiapps-live>
- 仮想マシン(VirtualBox)上で直接ブートできるDebian Linux環境
 - Windows、Macなどで利用可
 - インストール作業なしで物質科学アプリを実行できる
- 最新版: バージョン3.2 (2020年10月26日公開)
- MateriAppsで紹介している公開アプリ・ツールを収録
 - abinit, AkaiKKR, ALPS, CONQUEST, CP2K, Feram ,ERmod, DCore, DSQSS, HΦ, LAMMPS, mVMC, OpenMX, Quantum ESPRESSO, SMASH, xTAPP 等
 - ParaView, Tapioca, OVITO, VESTA, VMD, XCrysDen, …
 - CASINO, GAMESS, VMDには自動インストーラーを準備
- MateriApps LIVE!サイトからダウンロード可能
 - 2013年7月以来、約6600コピーを配布

セットアップに必要なファイル

- VirtualBox インストーラ: VirtualBox-*-OSX.dmg, VirtualBox-*-Win.exe
(<https://www.virtualbox.org/wiki/Downloads> からダウンロード)
- MateriApps LIVE! VitualBox 仮想ディスクイメージ(OVAファイル): MateriAppsLive-*.ova
(<http://sourceforge.net/projects/materiappslive/files/> からダウンロード)
- ドキュメント:
README.html, README-en.html
[https://github.com/cm*si*/MateriAppsLive/wiki/MateriAppsLive-ova](https://github.com/cm<i>si</i>/MateriAppsLive/wiki/MateriAppsLive-ova)

MateriApps LIVE! のセットアップ

1. VirtualBox のダウンロードサイト

<https://www.virtualbox.org/wiki/Downloads>

からインストーラをダウンロードする。

- Windows版インストーラ: **VirtualBox-6.1.16-140961-Win.exe**
- Mac版インストーラ: **VirtualBox-6.1.16-140961-OSX.dmg**

2. インストーラをダブルクリックして VirtualBox をインストールする。

3. MateriApps LIVE! 3.2 のダウンロードサイト

<https://sourceforge.net/projects/materiappslive/files/Release-3.2>

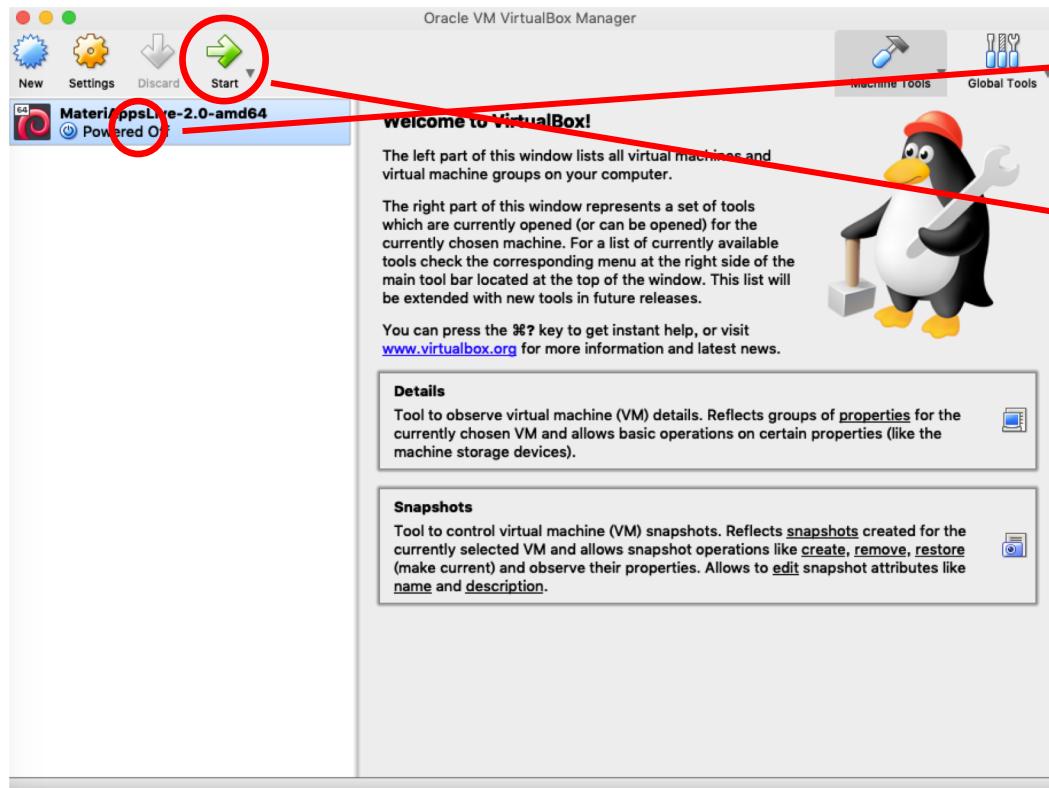
から MateriApps LIVE! 3.2 の OVAファイル(**MateriAppsLive-3.2-amd64.ova**)をダウンロードする。

4. MateriApps LIVE! のOVAファイルをダブルクリックする。

5. VirtualBox が(自動的に)起動してインポート画面が開くので、「インポート」ボタンを押す。 2~3分かかるが完了するとマネージャーが起動する。

MateriApps LIVE! の起動

VirtualBoXを起動する。

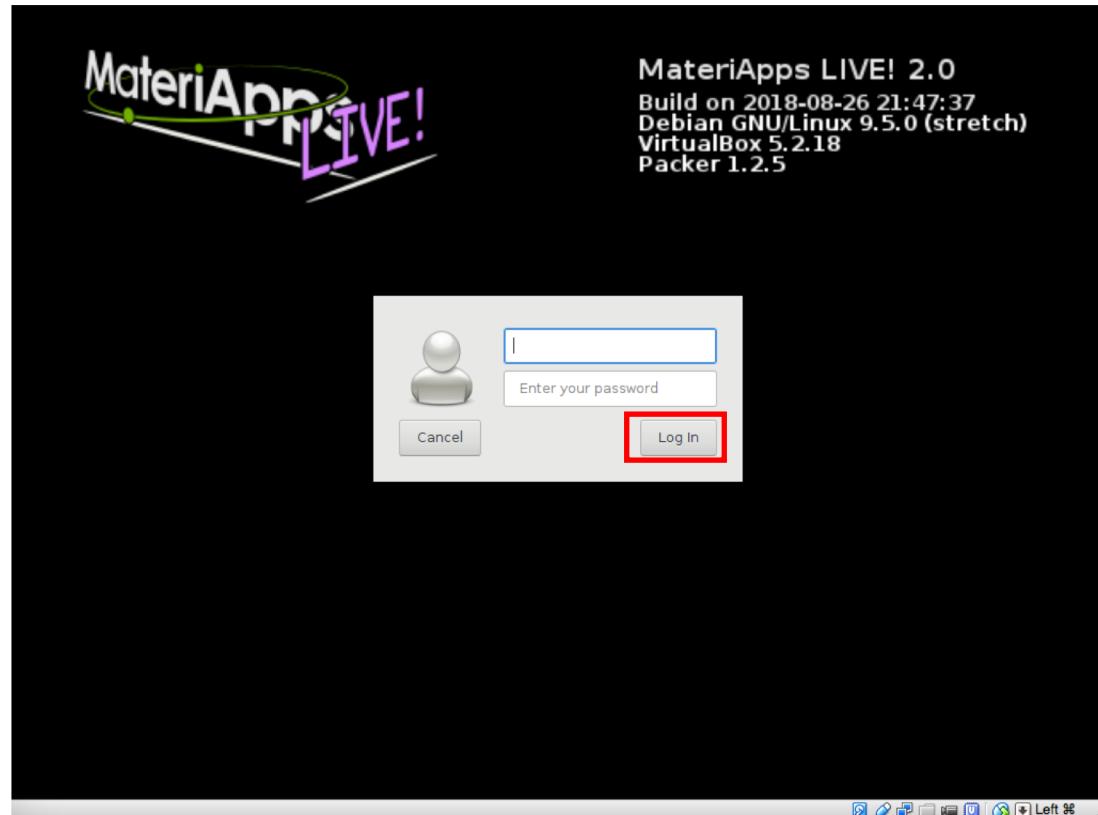


- ① 「MateriAppsLive…」を選択する。
- ② 起動ボタンを押す。
- ③ ログイン画面ができるまでそのまま待つ。

MateriApps LIVE!へのログイン

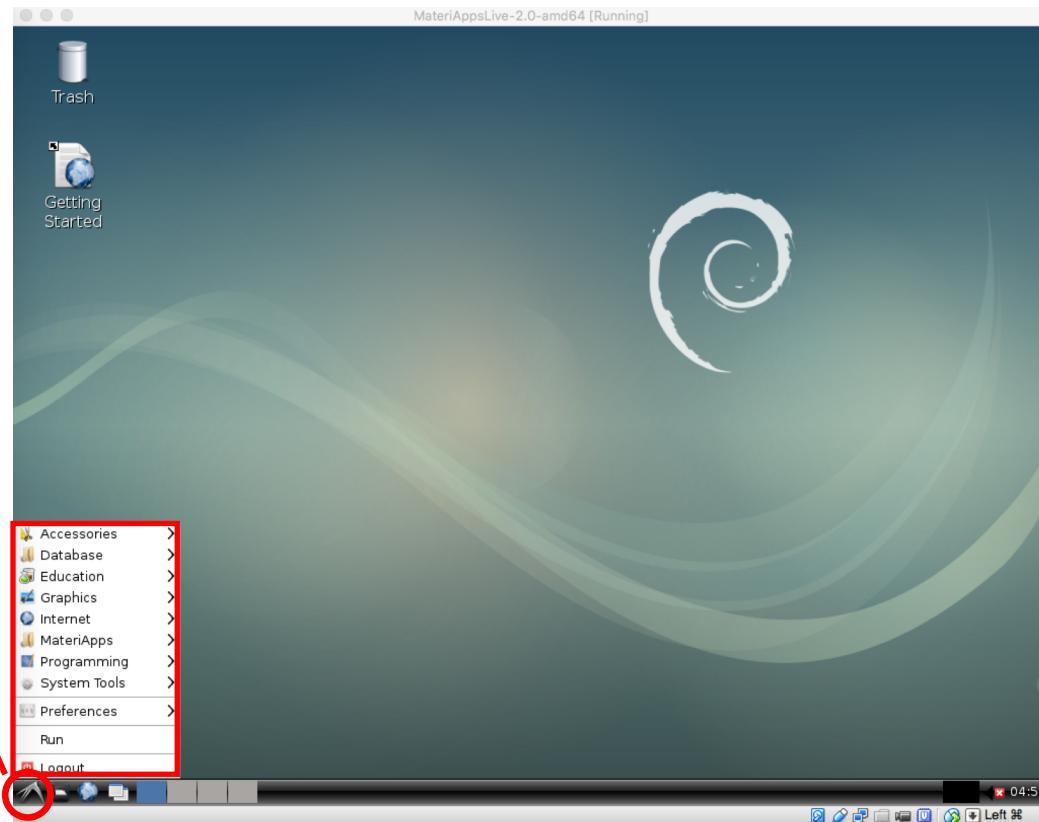
1. MateriApps LIVE!を起動すると、ログイン画面が表示される
2. 下記の情報を使ってログイン

- ・ユーザ名(login): *user*
- ・パスワード(password): *live*



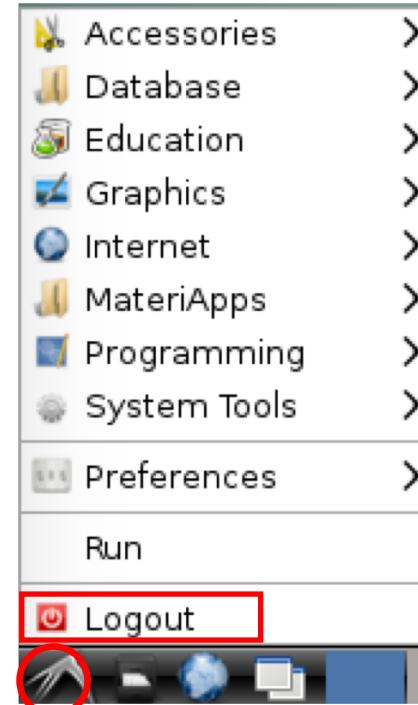
アプリケーションの起動

スタートメニュー



MateriApps LIVE! の終了

- ・「スタートメニュー」
⇒「Logout」
- ・下の画面の「Shutdown」選択



VirtualBox の設定

✓ 設定: 不要なポップアップメッセージを非表示にする

- Windows: ダウンロードしたVirtualBox 設定ファイル vbconfig.bat をダブルクリック

- Mac OS X: vbconfig.command をダブルクリック

あるいはターミナルで「*sh vbconfig.command*」を実行

✓ 設定: ホストOSのディスクに仮想マシンからアクセスできるように

- ① VirtualBox マネージャー画面で MateriAppsLive-* を選択し「設定」する。

- ② 「共有フォルダー」タブを開き、右側の「+」(新規共有フォルダーを追加します)をクリックする。

- ③ 「フォルダーのパス」の右側の「v」マークをクリックし、「その他」を選択。さきほどUSBメモリからコピーしたフォルダーを選択する。

- ④ 「自動マウント」をチェックし「OK」⇒ さらに「OK」

- ⑤ 仮想マシンを起動すると、3で選択したフォルダが、/media/sf_... の下に見える。

■ホスト (ホストOS) : もともと動いている OS (Windows、Mac OS X など)のこと

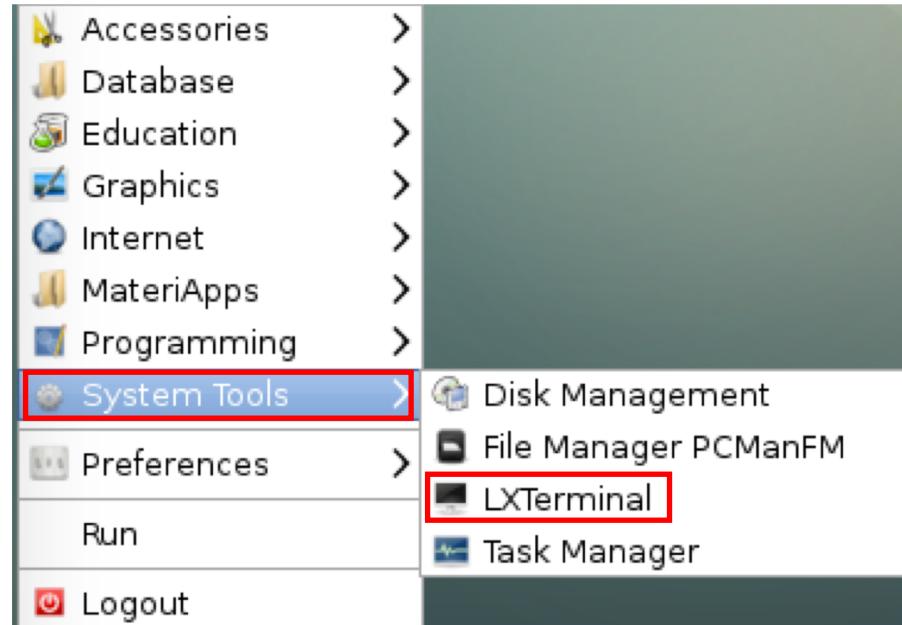
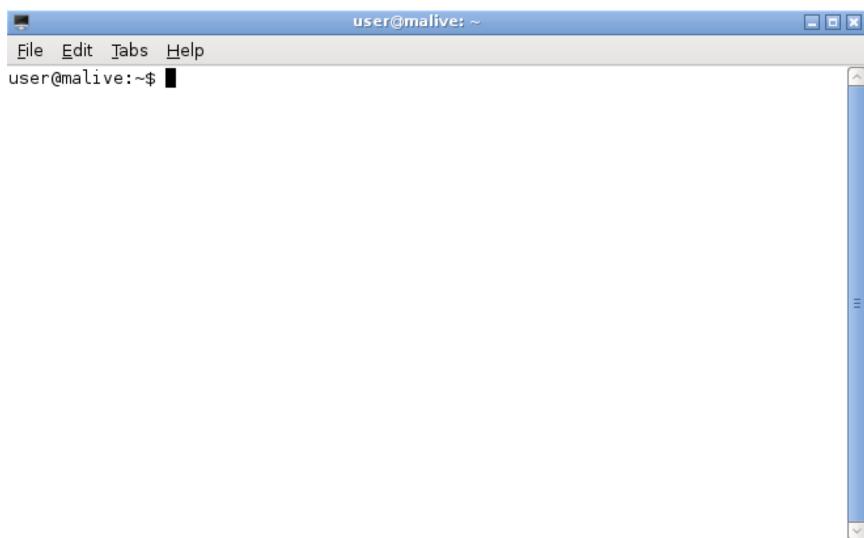
■仮想マシン (ゲストOS) : VirtualBox の中に動いている OS (= MateriApps LIVE!)

日本語キーボード、コピー&ペースト

- ・日本語キーボード(「@」が「P」の右にあるタイプ)では、記号が正しく入力できません。その場合、以下の設定を行ってください。
 - ① 「スタートメニュー」⇒「System Tools」⇒「LXTerminal」
 - ② ターミナル(端末)が立ち上がるるので「`setxkbmap -layout jp`」と入力し、リターン
 - ③ 「@」が正しく入力できることを確認
(英語配列に戻したいとき:「`setxkbmap -layout us`」)
- ・ホストOSでPDFファイルからコピーした文字列を、仮想マシンの端末でペーストする方法
 - ① 端末上で右クリック ⇒「Paste」
 - ② あるいは、「shift」と「control」を同時に押しながら「V」
 - ③ 文字列のコピーは、右クリック ⇒「Copy」あるいは「shift + control + C」

ターミナルの起動

- ・「スタートメニュー」
⇒「System Tools」
⇒「LXTerminal」



LAMMPSについて

- Large-scale **A**tomic/**M**olecular **M**assively **P**arallel **S**imulator
<https://lammps.sandia.gov/>
- 並列計算機のために設計された分子動力学シミュレータ
 - 物質系を構成する原子や分子の1つ1つに対する運動方程式を数値的に計算している
 - 高並列計算のため2007年にC++でオブジェクト指向で完全に、rewriteされている
オブジェクト指向 → コード拡張・公開 → 多分野・多スケールに渡る計算
 - 現在、粒子系統合シミュレータとして高い支持
- ソースコードダウンロード
<https://lammps.sandia.gov/download.html>
 - 最新のStable version (29 Oct 2020) : **2020年11月24日現在**
ソースファイル名 : **lammps-stable.tar.gz**
 - Stable以外の最新のコードもある
過去のメジャーなコード : <http://lammps.sandia.gov/tars/>
- ドキュメント
<https://lammps.sandia.gov/doc/Manual.html>

何か不明なことがあつたらこの
ページを見る

分子動力学計算について

- ニュートンの運動方程式

$$F=ma$$

- 複数の粒子の運動を記述する運動方程式を時間に関して積分して粒子の軌跡を求める。
- 力は経験的に決められたポテンシャル（力場）の空間座標に対する微分である。
- 例：モデリング（平衡化）の流れ
 - 平衡化計算
エネルギー極小化
 - 平衡化計算
温度一定MD
 - 平衡化計算
温度・圧力一定MD
 - 本計算
温度・圧力一定MD

力場(ポテンシャル)について(1)

- LAMMPS Force Fields
https://lammps.sandia.gov/doc/99/force_fields.html
- まずは、パッケージにある examples や potentials を参考にする。
 (例) LAMMPSのパッケージ内にある potentials の
`/usr/share/lammps/potentials`
 を参考にする。
- 目安としては、計算対象の物性が正しいものを選ぶ。
 (例) 常温常圧の鉄は、体心立方格子構造(bcc構造)
`/usr/share/lammps/potentials/Fe_mm.eam.fs` を用いる

```
# Interatomic potential – Embedded Atom Method
pair style eam/fs
# interaction pairs , filename , Element parameters
pair coeff * * Fe mm.eam. fs Fe
```

入力ファイルの書式例

- 自分でカスタマイズできるが、初心者には難しい。
 (→ 専門家に相談)
- 力場(ポテンシャル)にこだわっても、結局は古典計算?
 経験的に決める
 → 過去の成功例、論文になっているポテンシャルを用いる。
 (非経験的に決めるなら第一原理計算)

物性が正しいように決める

力場(ポテンシャル)について(2)

- ポテンシャルのデータベース
Interatomic Potentials Repository Project
<https://www.ctcms.nist.gov/potentials/>
(例) Al-Ni : <https://www.ctcms.nist.gov/potentials/Al-Ni.html>
Mishin-Ni-Al-2009.eam.alloy を用いる
- 分子モデリングソフトウェアで設定する
下記のURLなどを参考にする
Winmostar: https://winmostar.com/jp/manual_jp.html
Molby: <https://molby.osdn.jp/doc/ja/index.html>

実行時エラー

- 実行時にエラーが出た場合

ホームページ

<https://lammps.sandia.gov/doc/Errors.html>

13.3. Error messages

1-3 bond count is inconsistent

An inconsistency was detected when computing the number of 1-3 neighbors for each atom. This likely means something is wrong with the bond topologies you have defined.

1-4 bond count is inconsistent

An inconsistency was detected when computing the number of 1-4 neighbors for each atom. This likely means something is wrong with the bond topologies you have defined.

Accelerator sharing is not currently supported on system

Multiple MPI processes cannot share the accelerator on your system. For NVIDIA GPUs, see the nvidia-smi command to change this setting.

All angle coeffs are not set

All angle coefficients must be set in the data file or by the angle_coeff command before running a simulation.

...

を参考にする。

LAMMPSのexamplesについて

- LAMMPS の examples

MateriApps LIVE!: `/usr/share/lammps/examples`

- examples の解説

<https://lammps.sandia.gov/doc/Examples.html>

- 2次元系

- colloid, crack, flow, friction,
micelle, nemd, obstacle, shear

- 3次元系

- **melt**, peptide

- 力場

- dreiding, kim, reax

- 高速化機能

- cuda, gpu, intel

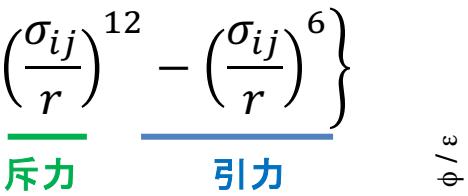
高分子物理向けのExamples

例	説明
melt	rapid melt of 3d LJ system
micelle	self-assembly of small lipid-like molecules into 2d bilayers
colloid	big colloid particles in a small particle solvent, 2d system
peptide	dynamics of a small solvated peptide chain (5-mer)

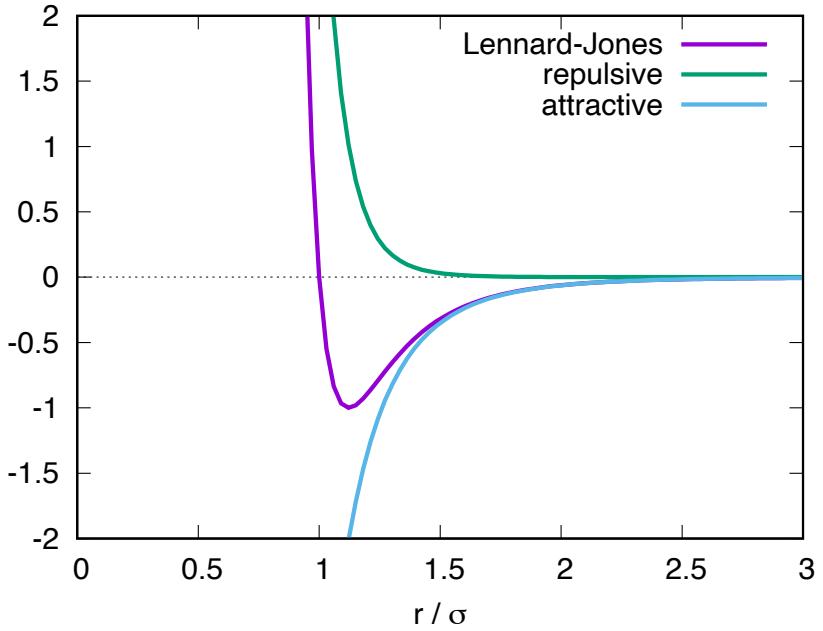
meltについて

- melt: FCC 格子状に並んだ Lennard-Jones (LJ) 粒子が溶けて拡散
- Lennard-Jones ポテンシャル

$$\phi_{ij}(r) = 4\epsilon_{ij} \left\{ \left(\frac{\sigma_{ij}}{r} \right)^{12} - \left(\frac{\sigma_{ij}}{r} \right)^6 \right\}$$


斥力 引力

- ポテンシャルの傾きが力として働く。
- 分子同士が十分離れているとき($r \rightarrow$ 大)、力は働くかない。
- 分子同士が近づくと、引力が働くようになり、 $r \sim 1.12$ 付近でエネルギーが最小となり、分子が近づきすぎると斥力(反発力)が働き、エネルギーは急激に増大する。



- ◆ 引力と斥力を持ち、希ガスの性質(ファンデルワールス力)を良く表す粒子モデルとして知られている。
- ◆ 古典MD計算のモデルポテンシャルとしてよく用いられる。

examples/melt(1)

入力ファイル: in.melt

```
# 3d Lennard-Jones melt
```

units lj
atom_style atomic

Lennard-Jones (LJ)
古典粒子

lattice fcc 0.8442
region box block 0 10 0 10 0 10
create_box 1 box
create_atoms 1 box
mass 1 1.0

格子定数: FCC 0.8442
シミュレーションボックスの指定
→ 今は、4,000原子の計算
create_atoms で、シミュレーション
ボックスの中に原子を配置する。

velocity all create 3.0 87287 速度の設定

pair_style lj/cut 2.5
pair_coeff 1 1 1.0 1.0 2.5

力場パラメータの設定 (ljポテンシャルを指定)

neighbor 0.3 bin
neigh_modify every 20 delay 0 check no

neighborでは、ポテンシャルcutoffよりどれだけ外側までを隣接原子として探す
かを指定(距離内にある粒子で neighbor list を構成する)。ここでは (cutoff +
0.3)Ang の範囲を隣接とし、bin で O(N) の探索手法を用いる

fix

1 all nve

fixで系の制御

NVEアンサンブル

examples/melt(2)

入力ファイル: in.melt (続き)

```
#dump id all atom 50 dump.melt

#dump 2 all image 25 image.*.jpg type type &      # は、コメントアウト
#       axes yes 0.8 0.02 view 60 -30           dump:出力形式の指定
#dump_modify 2 pad 3

#dump 3 all movie 25 movie.mpg type type &
#       axes yes 0.8 0.02 view 60 -30
#dump_modify 3 pad 3

thermo 50 何ステップ毎に温度情報を出力するかの指定
run    250 計算実行(step)数
```

詳しくは、

Commands

https://lammps.sandia.gov/doc/commands_list.html

を参照

実行準備

- ホームディレクトリに作業ディレクトリを作る。(例: \$HOME/lammps)

```
$ cd $HOME
```

```
$ mkdir lammps
```

- 作成した作業ディレクトリへ移動する。(\$HOME/lammps)

```
$ cd lammps
```

- examplesファイルのコピー (melt, micelle, rerun)

```
$ cp -r /usr/share/lammps/examples/melt ./
```

```
$ cp -r /usr/share/lammps/examples/micelle ./
```

```
$ cp -r /usr/share/lammps/examples/rerun ./
```

meltの実行

- melt へ移動

```
$ cd melt
```

- melt の中のファイルの確認

```
$ ls
```

```
in.melt  log.27Nov18.melt.g++.1  log.27Nov18.melt.g++.4
```

- in.melt の実行

```
$ lammmps < in.melt
```

```
LAMMPS (21 Jul 2020)
```

```
OMP_NUM_THREADS environment is not set. Defaulting to 1 thread.
```

```
(../comm.cpp:90)
```

```
using 1 OpenMP thread(s) per MPI task
```

```
...
```

```
Total # of neighbors = 151513
```

```
Ave neights/atom = 37.8783
```

```
Neighbor list builds = 12
```

```
Dangerous builds not checked
```

```
Total wall time: 0:00:00
```

エラーメッセージなしで
Total wall time が表示されれば
正常終了

- log.lammps というログファイルも出力される。

melt: 計算結果を可視化する(1)

- melt の入力ファイル(in.melt)を編集する。
- 可視化用の入力ファイル in.melt.viz を作成する。
 \$ cp in.melt in.melt.viz
- エディタ(emacs)を起動する。
 \$ emacs in.melt.viz &

melt: 計算結果を可視化する(2)

- emacs の設定

emacs@malive.local

File Edit Options Buffers Tools Help

3d Lennard-Jones melt

```

units      lj
atom_style atomic

lattice    fcc 0.8442
region     box block 0 10 0 10 0 10
create_box 1 box
create_atoms 1 box
mass       1 1.0

velocity   all create 3.0 87287

pair_style lj/cut 2.5
pair_coeff 1 1 1.0 1.0 2.5

neighbor   0.3 bin

```

in.melt Top L1 (Fundamental)-----

Welcome to GNU Emacs, one component of the GNU/Linux operating system.

[Emacs Tutorial](#) Learn basic keystroke commands
[Emacs Guided Tour](#) Overview of Emacs features at gnu.org
[View Emacs Manual](#) View the Emacs manual using Info
[Absence of Warranty](#) GNU Emacs comes with ABSOLUTELY NO WARRANTY
[Copying Conditions](#) Conditions for redistributing and changing Emacs
[Ordering Manuals](#) Purchasing printed copies of manuals
To quit a partially entered command, type Control-g.

This is GNU Emacs 23.4.1 (i486-pc-linux-gnu, GTK+ Version 2.24.10)
of 2012-09-10 on murphy, modified by Debian
Copyright (C) 2012 Free Software Foundation, Inc.

[Dismiss this startup screen](#) Never show it again.

-U:%%- *GNU Emacs* All L3 (Fundamental)-----

For information about GNU Emacs and the GNU system, type C-h C-a.

①

Welcome to GNU Emacs, one component of the GNU/Linux operating system.

[Emacs Tutorial](#) Learn basic keystroke commands
[Emacs Guided Tour](#) Overview of Emacs features at gnu.org
[View Emacs Manual](#) View the Emacs manual using Info
[Absence of Warranty](#) GNU Emacs comes with ABSOLUTELY NO WARRANTY
[Copying Conditions](#) Conditions for redistributing and changing Emacs
[Ordering Manuals](#) Purchasing printed copies of manuals
To quit a partially entered command, type Control-g.

This is GNU Emacs 23.4.1 (i486-pc-linux-gnu, GTK+ Version 2.24.10)
of 2012-09-10 on murphy, modified by Debian
Copyright (C) 2012 Free Software Foundation, Inc.

[Dismiss this startup screen](#) Never show it again.

② ここ(青字)をクリック

① チェックを入れる

②

emacs@malive.local

File Edit Options Buffers Tools Help

3d Lennard-Jones melt

```

units      lj
atom_style atomic

lattice    fcc 0.8442
region     box block 0 10 0 10 0 10
create_box 1 box
create_atoms 1 box
mass       1 1.0

velocity   all create 3.0 87287

pair_style lj/cut 2.5
pair_coeff 1 1 1.0 1.0 2.5

neighbor   0.3 bin
neigh_modify every 20 delay 0 check no

fix        1 all nve

#dump      id all atom 50 dump.melt

#dump      2 all image 25 image.*.jpg type type &
#           axes yes 0.8 0.02 view 60 -30
#           pad 3

#dump      3 all movie 25 movie.mpg type type &
#           axes yes 0.8 0.02 view 60 -30
#           pad 3

thermo    50
run       250

```

in.melt All L1 (Fundamental)-----

Wrote /home/user/.emacs

③

melt: 計算結果を可視化する(3)

- melt の入力ファイル(in.melt.viz)の編集

```
# 3d Lennard-Jones melt

units          lj
atom_style     atomic

lattice        fcc 0.8442
region         box block 0 10 0 10 0 10
create_box     1 box
create_atoms   1 box
mass           1 1.0

velocity       all create 3.0 87287

pair_style     lj/cut 2.5
pair_coeff    1 1 1.0 1.0 2.5

neighbor       0.3 bin
neigh_modify   every 5 delay 0 check no
fix            1 all nve

dump          id all atom 5 dump.melt
...
```

20 から 5 に変更する

#を取り、四角で囲まれた内容
に修正する

melt: 計算結果を可視化する(4)

- 編集した in.melt.viz の実行

```
$ lammmps < in.melt.viz
```

エラーメッセージなしで
Total wall time が表示されれば
正常終了

- dump.melt というファイルができているか確認する。

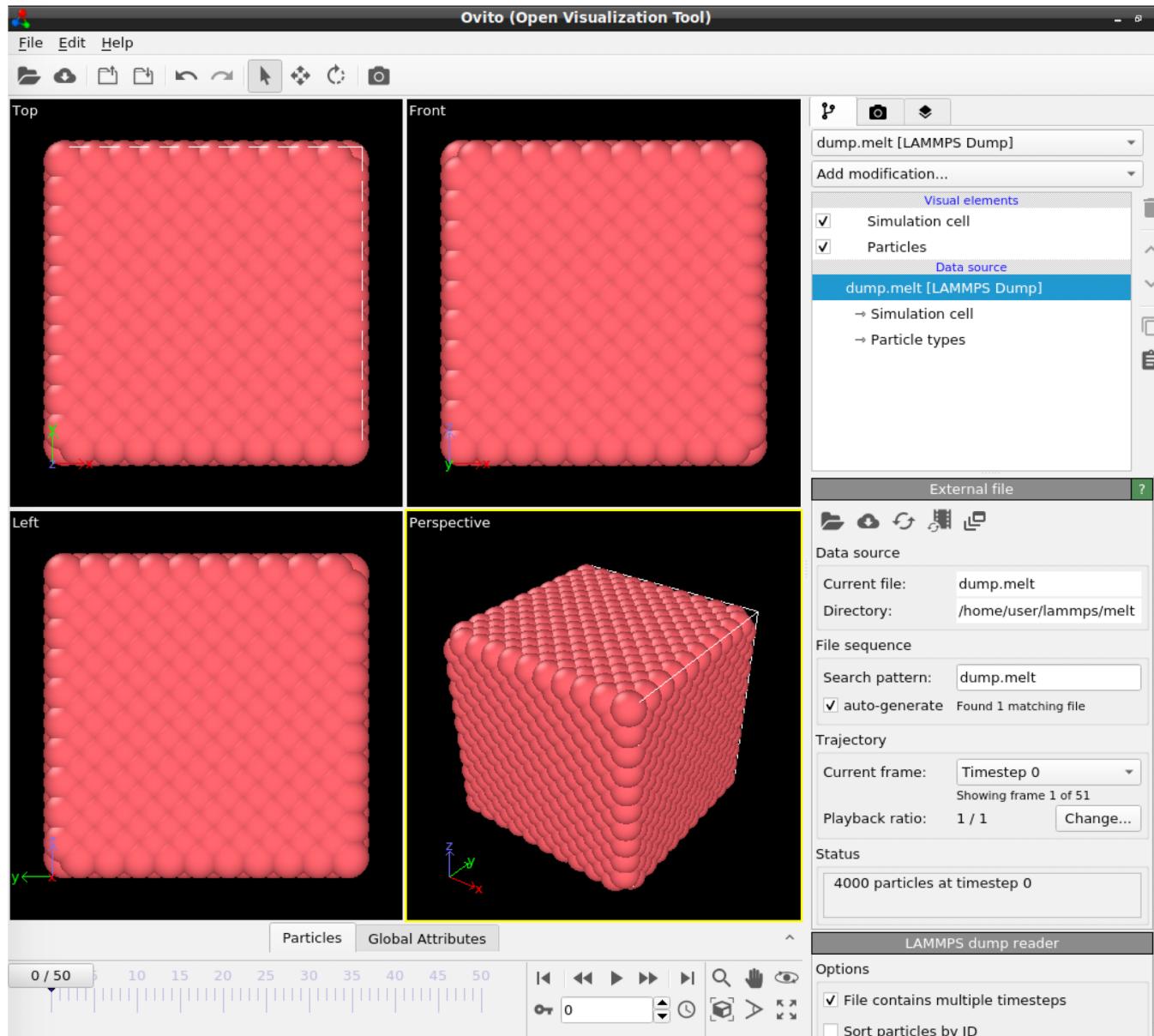
```
$ ls  
dump.melt    in.melt.viz      log.lammps
```

- MateriApps LIVE! (仮想マシン) 上で、OVITO で可視化する。

```
$ ovito dump.melt
```

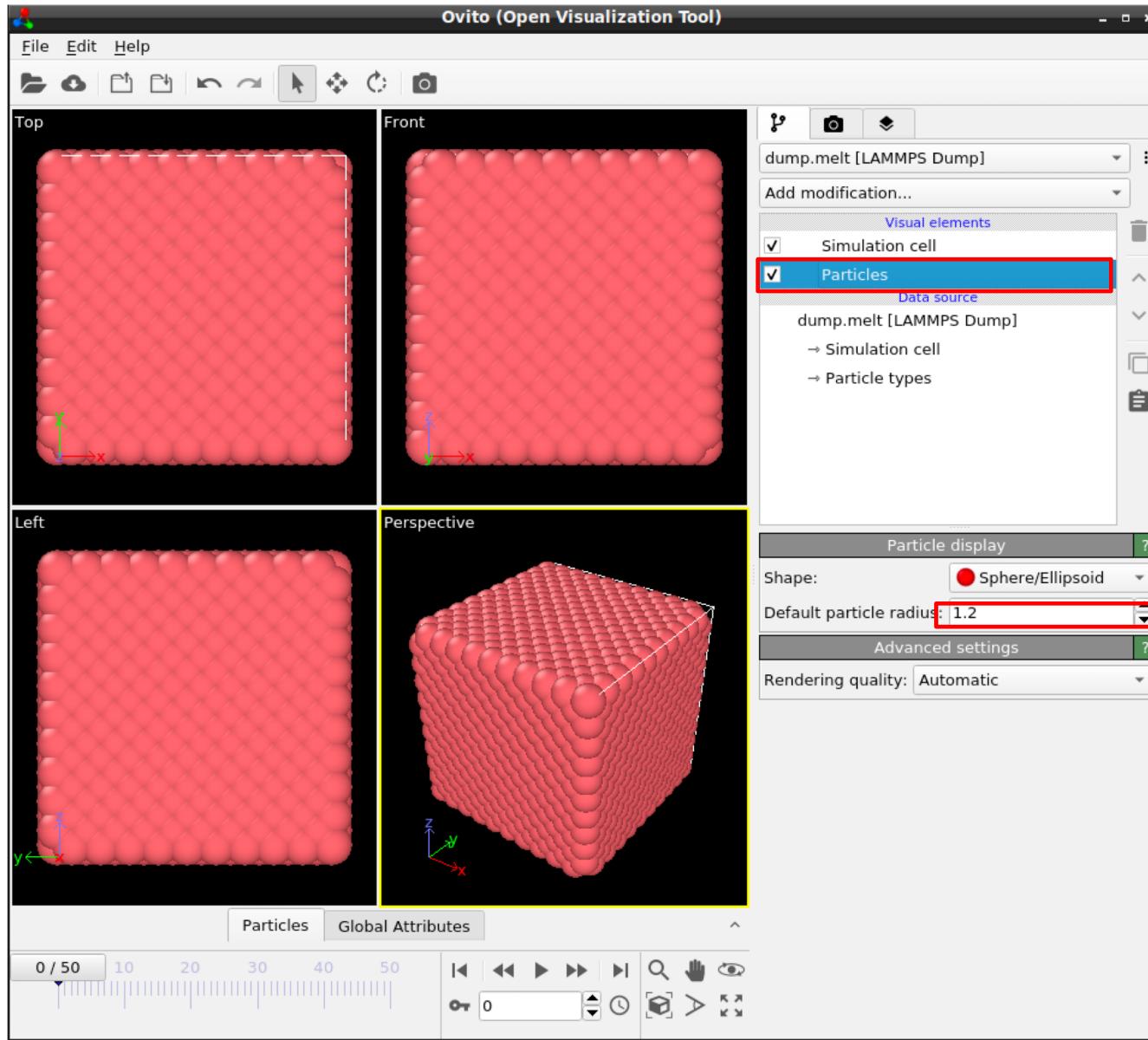
melt: 計算結果を可視化する(5)

- OVITO の起動



melt: 計算結果を可視化する(6)

- 粒子のサイズの変更

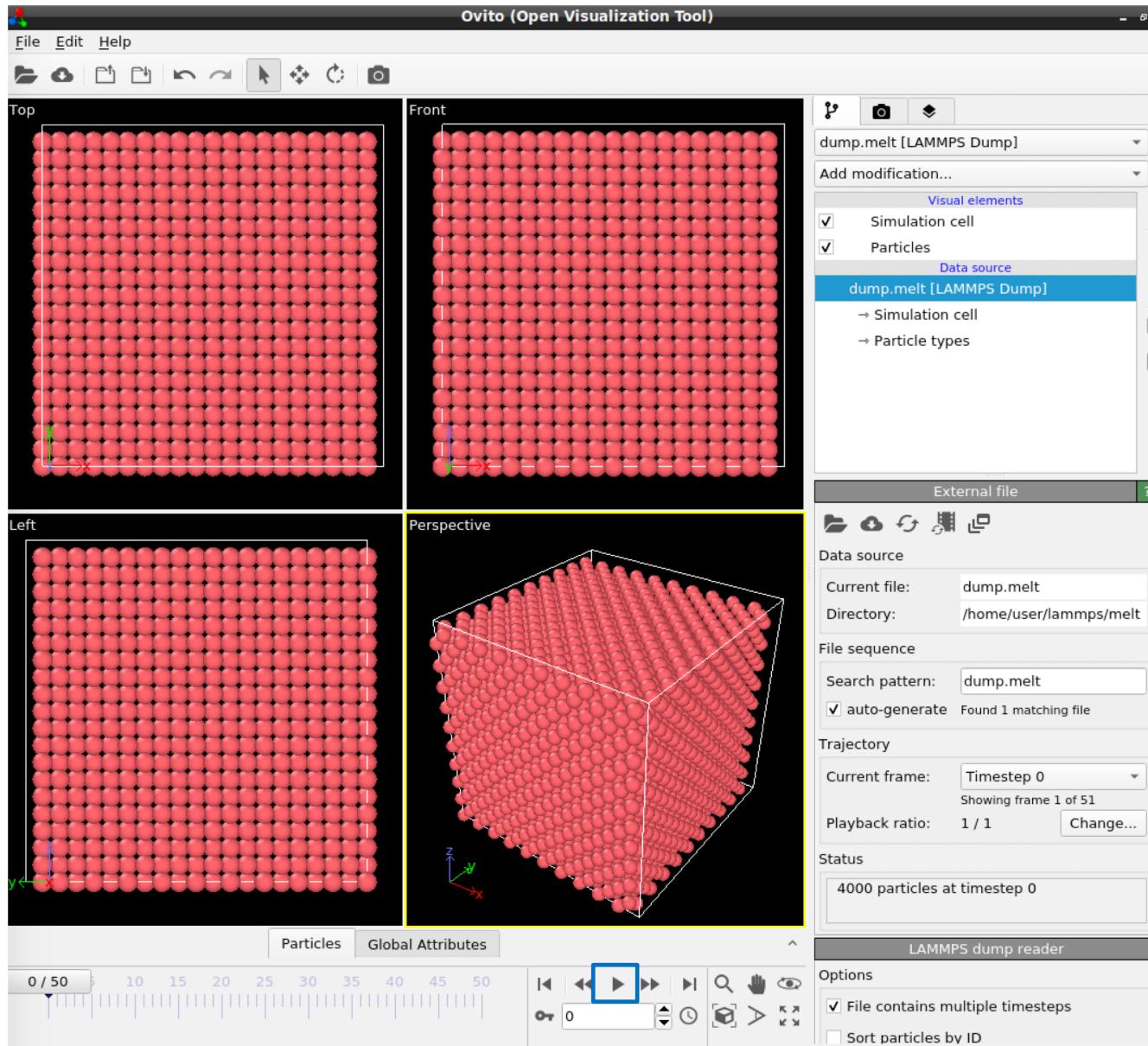


①「Particle」を選択する。

② 0.5 程度に設定する。

melt: 計算結果を可視化する(7)

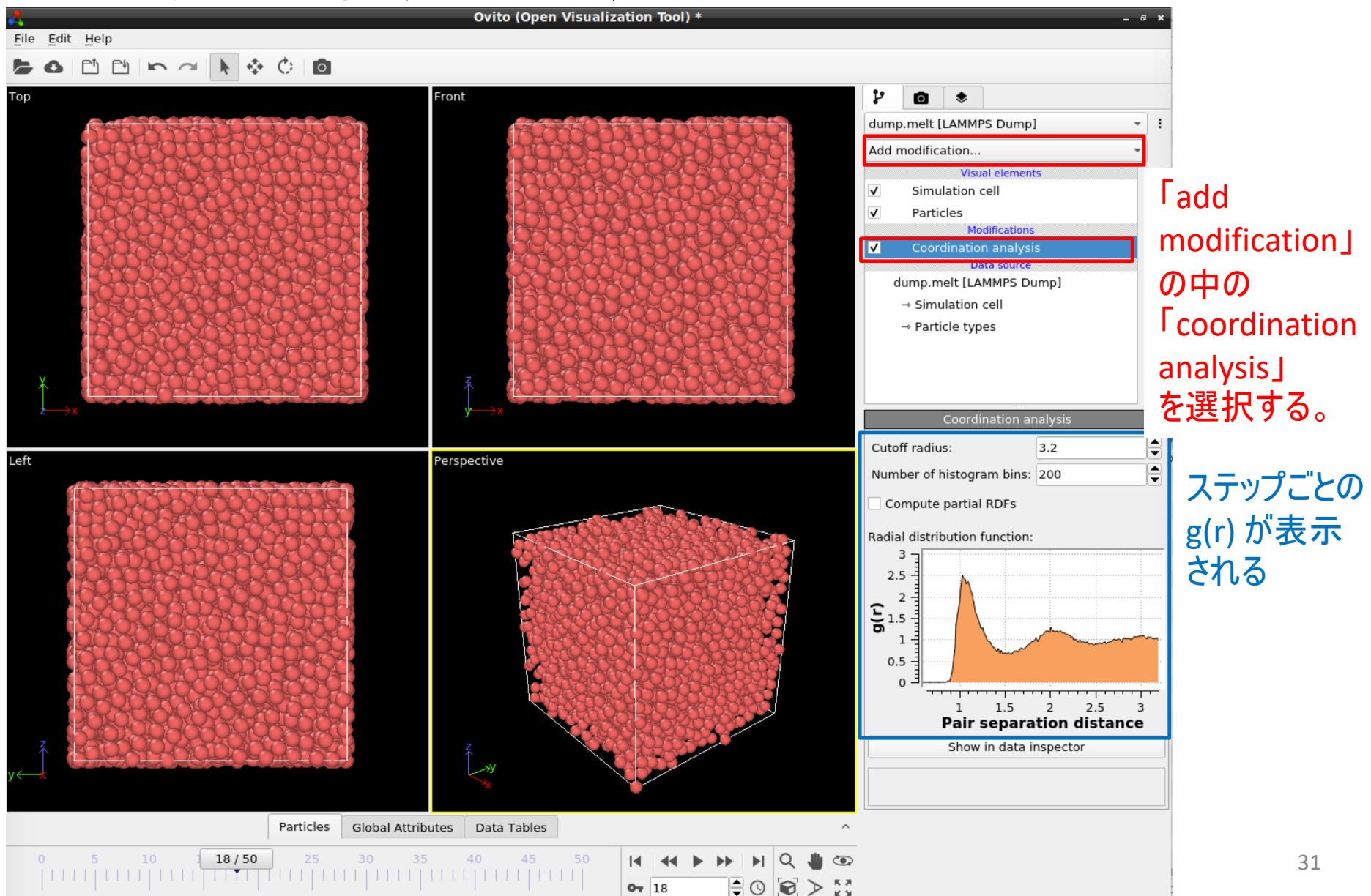
- 動画再生



スタートボタン
「▶」を押す

melt: 計算結果を可視化する(8)

- 動径分布関数(RDF)の表示



melt: 計算結果を可視化する(9)

- ホストマシンの OVITOで可視化する。
- ホストマシンに dump.melt をコピーする。
(「Shared」というディレクトリが共有化されているとする。)
\$ cp dump.melt /media/sf_Shared

melt: エネルギーをプロットする(1)

- エネルギーを出力するの入力ファイル in.melt.erg を作成する。
\$ cp in.melt.viz in.melt.erg
- エディタ(emacs)を起動する。
\$ emacs in.melt.erg &

melt: エネルギーをプロットする(2)

- melt の入力ファイル(in.melt.erg)の編集

```
# 3d Lennard-Jones melt

units          lj
atom_style     atomic

...
neighbor      0.3 bin
neigh_modify   every 5 delay 0 check no

fix           1 all nve                                         追記する

variable ST equal step          ST にステップ数を割り当てる
variable ETOT equal etotal     ETOT にトータルエネルギーを割り当てる

fix    energy all print 2 "${ST} ${ETOT}" file out.erg screen no

dump          id all atom 5 dump.melt                         out.erg というファイルを生成し、
...                                         画面には値を出力しない
thermo        50
run           1000                                         ステップ数を増やす
```

melt: エネルギーをプロットする(3)

- 編集した in.melt.erg の実行

```
$ lammps < in.melt.erg
```

エラーメッセージなしで
Total wall time が表示されれば
正常終了

- out.erg というファイルができているか確認する。

```
$ ls
```

```
out.erg      dump.melt      in.melt.erg      log.lammps
```

- gnuplot を用いてプロットする。

```
$ gnuplot
```

```
...
```

```
GNUPLOT
```

```
Versin 5.2 patchlevel 6
```

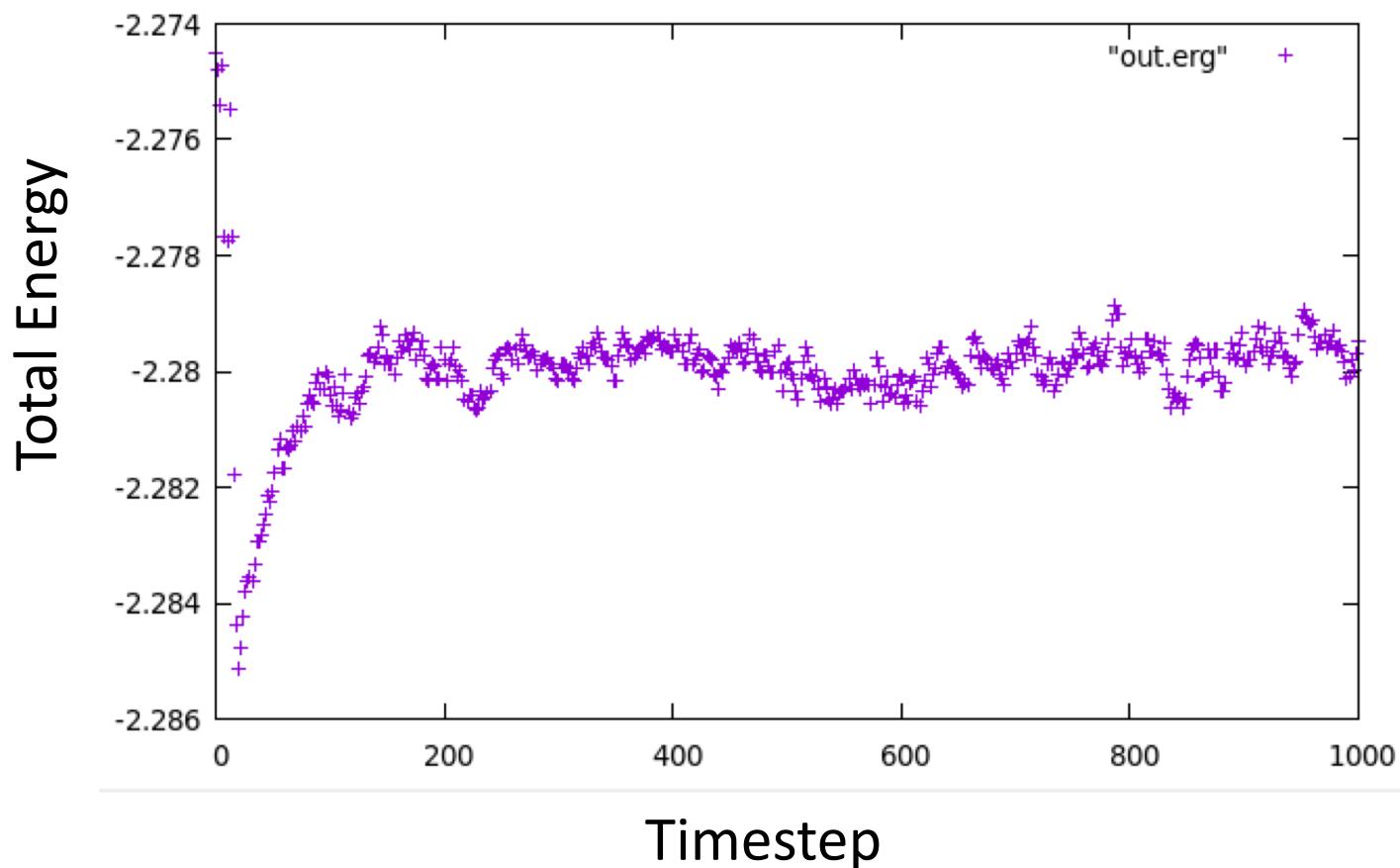
```
last modified 2019-01-01
```

```
...
```

```
Terminal type is now 'wxt'
```

```
gnuplot> plot "out.erg"
```

melt: エネルギーをプロットする(4)



micelle の実行

- micelle: 会合する分子の2次元の計算
micelle へ移動 → melt と同様の操作

\$ cd \$HOME/lammps/micelle

\$ lammps < in.micelle

(注) 21 Jul 2020 の実行では、

WARNING: Communication cutoff 1.42246 is shorter than a bond length based estimate of 1.425. This may lead to errors. (../comm.cpp:680)

の警告ができるが(カットオフを大きくすると回避できる)、この演習においては無視してください。
(examples の中にあるログファイル 29 Mar 2020 の実行においては、この警告は出ない。)

→ 自分で入力ファイルを作成する時は警告は出ないようにした方が良い。

micelle の可視化(1)

- micell の入力ファイル(in.micell)の編集

```
# 2d micelle simulation

dimension 2      2次元の計算

neighbor 0.3 bin
neigh_modify delay 5

atom_style bond

# Soft potential push-off

read_data data.micelle      データファイル data.micelle を読み込む
special_bonds fene

...
fix 1 all nve      NVEアンサンブル
...
thermo 50

dump 1 all atom 2000 dump.micelle
```

← #を取る

micelle の可視化(2)

```
#dump      2 all image 2000 image.*.jpg type type zoom 1.6
#dump_modify 2 pad 5 adiam 1 0.5 adiam 2 1.5 adiam 3 1.0
adiam 4 0.75

#dump      3 all movie 2000 movie.mpg type type zoom 1.6
#dump_modify 3 pad 5 adiam 1 0.5 adiam 2 1.5 adiam 3 1.0
adiam 4 0.75

reset timestep 0
run    100000
```

ステップ数を増やす

micelle の可視化(2)

- 編集した in.micelle の実行

```
$ lammps < in.micelle
```

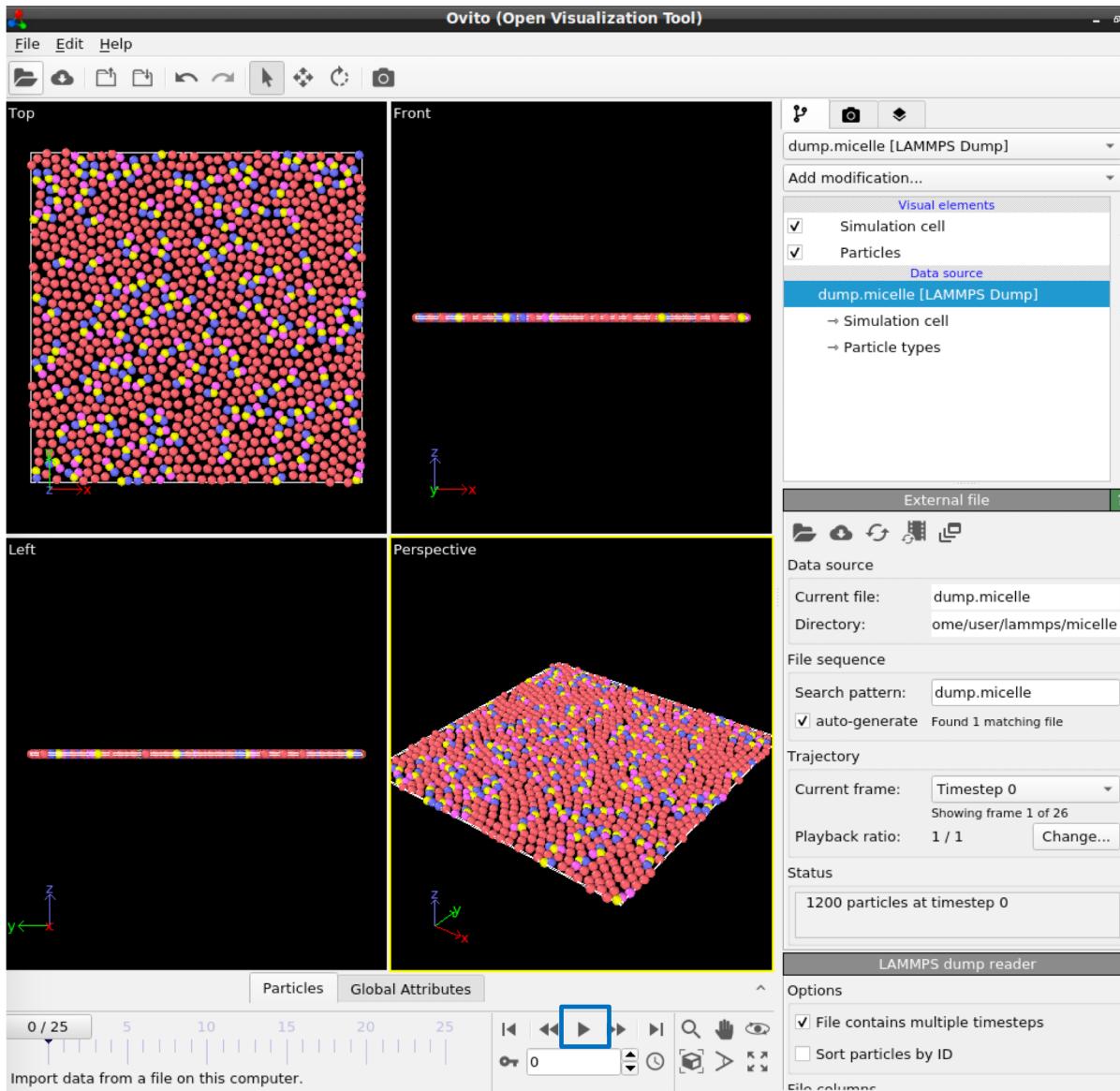
エラーメッセージなしで
Total wall time が表示されれば
正常終了

- dump.micelle というファイルができているか確認する。

```
$ ls  
data.micelle dump.micelle log.lammps  
def.micelle in.micelle
```

micelle の可視化(3)

- dump.micelle を開く
\$ ovito dump.micelle



スタートボタン
「▶」を押す

rerun: 動径分布関数(RDF)をプロットする(1)



- rerun へ移動

```
$ cd $HOME/lammps/rerun
```

- 入力ファイル in.rdf.first の内容の確認

```
$ less in.rdf.first
```

(注) less の表示は、"q"ボタンを押して閉じる。

```
# 3d Lennard-Jones melt

variable      x index 1
variable      y index 1
variable      z index 1
...
fix          1 all nve

dump         1 all custom 100 lj.dump id type x y z

compute      myRDF all rdf 50 cutoff 2.5    半径 2.5の距離において、50 点計算
fix          2 all ave/time 100 10 1000 c_myRDF[*] file rdf.first mode vector
thermo      100
run          1000
```

FCC 格子状に並んだ
Lennard-Jones(LJ)粒子が
溶けて拡散
32,000原子

rerun: 動径分布関数(RDF)をプロットする(2)



- in.rdf.first の実行

```
$ lammmps < in.rdf.first
```

エラーメッセージなしで
Total wall time が表示されれば
正常終了

- rdf.first というファイルができているか確認する。

```
$ ls  
rdf.first    lj.dump    in.rdf.first    log.lammps
```

- gnuplot を用いてプロットする。

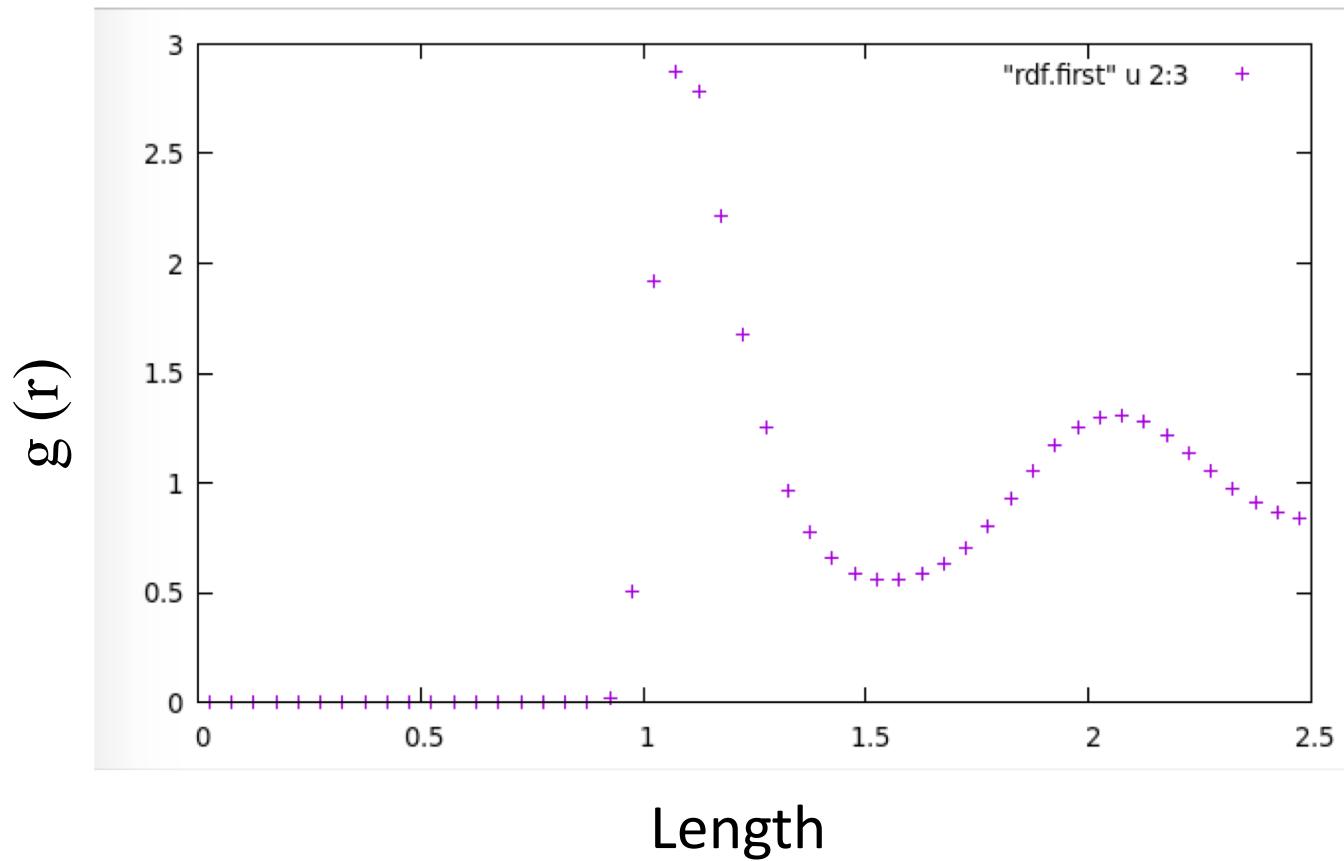
```
$ gnuplot
```

...

```
gnuplot> plot "rdf.first" u 2:3
```

rerun: 動径分布関数(RDF)をプロットする(3)

FCCから溶けたときの構造を表している



並列実行(1)

- MPIプロセス数=4 での実行

マシンに4コア以上ある場合

```
$ mpirun -oversubscribe -n 4 lammps < in.melt  
LAMMPS (21 Jul 2020)
```

...

1 by 2 by 2 MPI processor grid

...

Performance: 90089.282 tau/day, 208.540 timesteps/s
22.1% CPU use with 4 MPI tasks x 1 OpenMP threads

...

並列実行(2)

- スレッド数=4 での実行

```
$ OMP_NUM_THREADS=4 lammps < in.melt
```

LAMMPS (21 Jul 2020)

using **4** OpenMP thread(s) per MPI task

...

Performance: 115561.777 tau/day, 267.504 timesteps/s

97.6% CPU use with 1 MPI tasks x **4** OpenMP threads

...

マシンに4コア以上ある場合

サイズを大きくした計算(1)

- 作業ディレクトリを作る。(例: \$HOME/lammps/melt2)
\$ cd \$HOME/lammps
\$ mkdir melt2
- 作成した作業ディレクトリへ移動する。(\$HOME/lammps/melt2)
\$ cd melt2
- meltディレクトリから in.melt をコピーする
\$ cp ../melt/in.melt ./
- melt の入力ファイルを開く
\$ emacs in.melt

サイズを大きくした計算(2)

```
# 3d Lennard-Jones melt

units          lj
atom_style     atomic

lattice        fcc 0.8442
region         box block 0 40 0 40 0 40      256,000原子になる
create_box     1 box
create_atoms   1 box
mass           1 1.0

velocity       all create 3.0 87287

pair_style     lj/cut 2.5
pair_coeff     1 1 1.0 1.0 2.5

neighbor       0.3 bin
neigh_modify   every 20 delay 0 check no

fix            1 all nve
```

サイズを大きくした計算(3)

- 編集した in.melt の実行

```
$ lammps < in.melt
```

- ログファイルの確認

```
$ cat log.lammps
```

...

Created 256000 atoms

...

サイズを大きくした計算(4)

examples/melt

実行時間(時:分:秒)

iMac:
2.8 GHz Intel Core i7

	手持ちのPC(iMac)	
プロセス数	1	4
4000原子1,000ステップ	0:00:02	0:00:01
4000原子10,000ステップ	0:00:29	0:00:11
256,000原子1,000ステップ	0:03:06	0:01:13
256,000原子10,000ステップ	0:31:15	0:12:48
2,048,000原子1,000ステップ	0:27:04	0:10:06
2,048,000原子10,000ステップ		1:46:38

フラットMPI

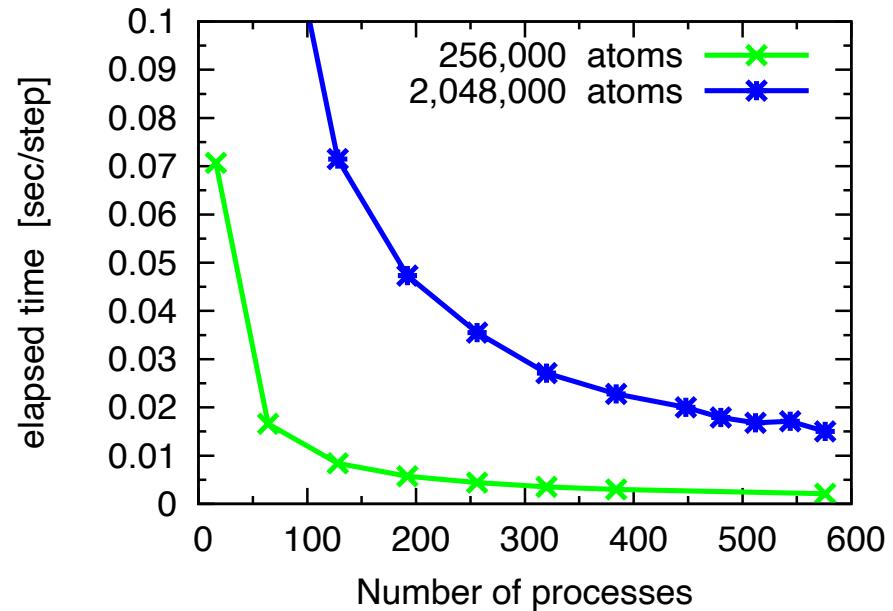
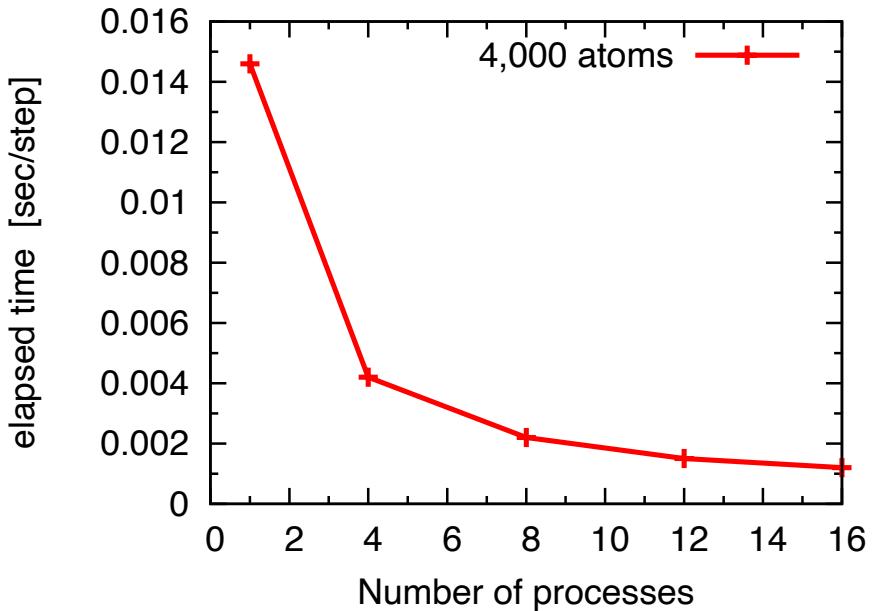
FX10		
16	256	576
0:00:01	0:00:05	
0:00:12	0:00:03	
0:01:11	0:00:04	0:00:02
0:11:47	0:00:44	0:00:21
0:08:56	0:00:36	0:00:15
1:28:35	0:05:55	0:02:30

- 主にプロセス数当たりの粒子数減少に伴う実行時間の減少
→ 計算規模(原子数など)が小さくて、プロセス数を増やすと、通信等により実行時間が増加する場合がある。
- 計算規模が大きくなる → [スパコン\(HPCI\)の利用へ](#)

サイズを大きくした計算(5)

examples/melt

フラットMPI(原子数:4,000、256,000、2,048,000)



4,000原子: ~16プロセス
 256,000原子: ~600プロセス
 2,048,000原子: 600プロセス以上

restartを行う(1)

コマンドの紹介

- write_restart/restart: 計算途中ファイル出力する。
- read_restart: 計算を再開する。
- clear: リセット

このコマンドを使用すると、1つの入力ファイルで複数のジョブを順番に実行できる。

(例)

(commands for 1st simulation)

clear

(commands for 2nd simulation)

- log: LAMMPSのログファイル(log.lammps)をクローズして、指定されたログファイルにログを取る。

restartを行う(2)

- examples の hugoniostat を実行する

exmapleファイルのコピー(hugoniostat)

```
$ cp -r /usr/share/lammps/examples/hugoniostat ./
```

(注)in.hugoniostatの実行には、SHOCKパッケージが必要。

SHOCKパッケージがないと、

```
ERROR: Unknown fix style nphug (./modify.cpp:854)
```

```
Last command: fix myhug all nphug temp 1.0 1.0 10.0 z 40.0 40.0 70.0 drag 0.0  
tchain 1 pchain 0
```

のエラーが出る。

- hugoniostatディレクトリの中を確認する

```
$ ls
```

```
in.hugoniostat      log.5Oct16.hugoniostat.g++.1
```

```
log.27Nov18.hugoniostat.g++.1 log.5Oct16.hugoniostat.g++.4
```

```
log.27Nov18.hugoniostat.g++.4
```

restartを行う(3)

in. hugoniostat(入力ファイル)

```

# This script reproduces stress trajectories from Fig. 1 in
# Ravelo, Holian, Germann, and Lomdahl, PRB 70 014103 (2004)
...
units          lj
boundary       p p p
atom_style     atomic
...
#
# Define initial velocity
velocity       all create 0.01 87287 mom yes rot yes dist gaussian
write_restart  restart.equil      restartファイルの出力

# Start Run #1
clear          clear によってリセットを行い、
               read_restart で、保存しておいた restart.equil の構造から計算を再開する
read_restart   restart.equil      restartファイルの入力
...

```

restartを行う(4)

- in.hugoniostat の実行

```
$ lammps < in.hugoniostat > log.hugoniostat
```

(log.hugoniostat には、標準出力の結果が保存される。)

- 出力されているログファイルを確認する。

```
$ ls
```

```
in.hugoniostat      log.5Oct16.hugoniostat.g++.1 log.lammps
```

```
log.27Nov18.hugoniostat.g++.1 log.5Oct16.hugoniostat.g++.4 restart.equil
```

```
log.27Nov18.hugoniostat.g++.4 log.hugoniostat      stress_vs_t.dat
```

- log.hugoniostat には、Run1～3の結果がまとめて出力されている。

【実習】

- Run 1～3ごとのログファイルを出力する

restartを行う(5)

in. hugoniostat(入力ファイル)の編集

```
...  
# Start Run #1  
  
log log.nodrag  
  
clear  
read_restart restart.equil  
  
...  
  
# Start Run #2  
  
log log.drag  
  
clear  
read_restart restart.equil  
  
...  
  
# Start Run #3  
  
log log.nhchains  
  
clear  
read_restart restart.equil  
  
...
```

赤文字の部分を追記すると、

保存しておいた restart.equil
(write_restart restart.equilで出力した)の
構造を
Run1～3の計算条件で計算とき、
ログファイルとして
Run1: log.nodrag
Run2: log.drag
Run3: log.nhchains
にそれぞれ保存

restartを行う(6)

- 編集した in.hugoniostat の実行
\$ lammmps < in.hugoniostat > log.hugoniostat
- log.nodrag, log.drag, log.nhchains というファイルができているか確認する。
\$ ls
in.hugoniostat log.5Oct16.hugoniostat.g++.1 log.hugoniostat
log.nodrag
log.27Nov18.hugoniostat.g++.1 log.5Oct16.hugoniostat.g++.4 log.lammmps
restart.equil
log.27Nov18.hugoniostat.g++.4 **log.drag** **log.nhchains**
stress_vs_t.dat
- それぞれのログファイルの中身を確認する。
\$ cat log.nodrag
など
- log.hugoniostat には、Run1～3(log.nodrag, log.drag, log.nhchains)の結果がoutputされている。

「富岳」を中心としたHPCIシステムで利用できるアプリケーション(1/4)

◎RISTで整備

シミュレーション	ソフトウェア名	理研 「富岳」	北大 <i>Grand Chariot</i>	北大 Polaire	東北大 AOBA-A (SX)	東北大 AOBA-B (LX)	筑波大 Cygnus	JCAHPC Oakforest-PACS	東大 Reedbush-H	東大 Reedbush-L	東大 Oakbridge-Cx	東工大 TSUBAME3.0	名大 FX1000	名大 CX2570	京大 CRAY XC40	阪大 OCTOPUS	九大 ITTO A	九大 ITTO B
分子動力学	AMBER									○						○	○	
	CHARMM															○	○	
	feram						○			○								
	GENESIS		◎		○		○	○			○		○		○	○	○	
	GROMACS	○	○	○				○	○	○	○	○	○	○	○	○	○	
	LAMMPS	○				○				○	○	○	○	○	○	○	○	
	MODYLAS		○			○		○			○	○	○	○	○	○	○	
	MyPresto		○	○														
	N2P2	○																
	NAMD		○	○														
	Tinker																	
量子化学	ABINIT-MP	○			○		○			○	○	○		○	○	○	○	
	BoltzTrap													○				
	GAMESS		○	○							○		○	○	○	○	○	
	Gaussian		○			○					○	○	○	○	○	○	○	
	GRRM				○													
	Molpro														○	○	○	
	NTChem	○			○			○		○	○	○		○	○	○	○	
	NWChem													○				
	Scigress														○			
	SMASH	○			○			○		○	○	○		○	○	○	○	

RISTでは、「京」及び「富岳」ではOSSを中心に、第二階層のHPCIでは国プロを中心に整備してきた。

ソフトウェア検索ページ（https://www.hpci-office.jp/pages/hardware_software?tab=software）に登録されている情報を編集

「富岳」を中心としたHPCIシステムで利用できるアプリケーション(2/4)

◎RISTで整備

シミュレーション	ソフトウェア名	理研 「富岳」	北大 Grand Chariot	北大 Polaire	東北大 AOBA-A (SX)	東北大 AOBA-B (LX)	筑波大 Cygnus	JCAHPC Oakforest-PACS	東大 Reedbush-H	東大 Reedbush-L	東大 Oakbridge-CX	東工大 TSUBAME3.0	名大 FX1000	名大 CX2570	京大 CRAY XC40	阪大 OCTOPUS	九大 ITo A	九大 ITo B
物性物理	ABINIT														○			
	AkaiKKR									○								
	ALPS									○								
	CP2K	○																
	HΦ		○		○						○					○	○	○
	Materials Studio										○							
	OpenMX		○		○		○				○							
	PHASE/0		○		○		○				○							
	Quantum ESPRESSO	○	○	○	○						○							
	SALMON		○		○						○							
	VASP			○	○											○		○
	xTAPP							○			○						○	○
計算生物学	Discovery Studio											○						
	rDock	○																
	Relion														○			
	Schrodinger Small Drug Discovery Suite												○					

ソフトウェア検索ページ (https://www.hpci-office.jp/pages/hardware_software?tab=software) に登録されている情報を編集

「富岳」を中心としたHPCIシステムで利用できるアプリケーション(3/4)

©RISTで整備

データサイエンス	ソフトウェア名	理研 「富岳」	北大 Grand Chariot	北大 Polaire	東北大 AOBA-A (SX)	東北大 AOBA-B (LX)	筑波大 Cygnus	JCAHPC Oakforest-PACS	東大 Reedbush-H	東大 Reedbush-L	東大 Oakbridge-CX	東工大 TSUBAME3.0	名大 FX1000	名大 CX2570	京大 CRAY XC40	阪大 OCTOPUS	九大 ITo A	九大 ITo B
機械学習	Caffe		○	○					○	○		○	○	○	○	○	○	
	Chainer	○	○	○					○	○		○	○	○	○	○	○	
	CNTK																○	
	cuDNN					○						○				○		
	DNNL	○					○											
	Horovod	○						○	○									
	intel DAAL		○	○														
	Keras	○							○	○			○	○	○			
	MXNet												○	○	○			
	ONNX												○	○	○			
	PyTorch	○											○	○	○			
	Scikit-Learn	○													○			
	TensorFlow	○	○	○					○	○	○	○	○	○	○	○	○	
	Theano								○	○		○	○	○	○	○		
	Torch								○	○						○		
統計解析	pandas	○													○			
/データ分析	PyMD	○																
	R	○	○	○						○	○			○	○	○	○	
	SAS													○	○			

ソフトウェア検索ページ (https://www.hpci-office.jp/pages/hardware_software?tab=software) に登録されている情報を編集

「富岳」を中心としたHPCIシステムで利用できるアプリケーション(4/4)

©RISTで整備

データサイエンス	ソフトウェア名	理研 「富岳」	北大 <i>Grand Chariot</i>	北大 <i>Polaire</i>	東北大 <i>AOBA-A (SX)</i>	東北大 <i>AOBA-B (LX)</i>	筑波大 <i>Cygnus</i>	JCAHPC <i>Oakforest-PACS</i>	東大 <i>Reedbush-H</i>	東大 <i>Reedbush-L</i>	東大 <i>Oakbridge-CX</i>	東工大 <i>TSUBAME3.0</i>	名大 <i>FX1000</i>	名大 <i>CX2570</i>	京大 <i>CRAY XC40</i>	阪大 <i>OCTOPUS</i>	九大 <i>ITO A</i>	九大 <i>ITO B</i>	産総研 <i>ABCI</i>
バイオ	bcftools	○																	
インフォマティクス	bedtools2	○ ○ ○																	
	biobambam	○ ○ ○																	
	bioconductor	○ ○ ○																	
	BioPerl	○ ○ ○																	
	BioRuby	○ ○ ○																	
	BLAST		○ ○	○ ○															
	BWA	○ ○ ○																	
	dssp	○ ○ ○																	
	GATK																		
	mapslice2	○ ○ ○																	
	Picard	○ ○ ○																	
	Pysam	○ ○ ○																	
	SAMtools	○ ○ ○																	
	Star	○ ○ ○																	
	TopHat	○ ○ ○																	
分散データ処理	Apache Hadoop																		○ ○
	Apache Spark		○ ○ ○																○ ○
	Gfarm		○ ○ ○																

ソフトウェア検索ページ (https://www.hpci-office.jp/pages/hardware_software?tab=software) に登録されている情報を編集

ソフトウェア検索システム@HPCIポータルサイト

http://www.hpci-office.jp/pages/project_categories_hardware

▼ HPCIシステムで利用可能なソフトウェアの検索

この条件で検索する >

検索範囲
 過去に利用可能であったソフトウェアと計算機資源の情報を含める

ソフトウェア名 = *入力がない場合はソフトウェア名での絞り込みは行いません。
 すべてを含む(AND) いずれかを含む(OR)

ソフトウェア分類

すべての分類

シミュレーション
 分子動力学 量子化学 物性物理 計算生物学 流体解析 構造／衝突解析
 電磁界解析 マルチフィジックス 粒子系 気象／気候 地震動

データサイエンス
 機械学習 統計解析/データ分析 バイオインフォマティクス 分散データ処理

プリボスト
 汎用可視化ソフトウェア メッシュ操作

ライブラリ等
 MPI通信 数値計算 数式処理 画像処理 データ圧縮 データ形式 Python関連
 その他のライブラリ等

開発環境
 コンパイラ／インタプリタ 性能分析 デバッガ その他の開発環境

システム基盤
 OS ジョブ管理 クラウド基盤

- **検索機能**

- ソフトウェア名による検索 [NEW]
- ソフトウェアの分類を選択してソフトウェア情報を絞り込み [NEW]
- 過去に利用可能であったソフトウェアと計算機資源の情報を含めた検索 [NEW]

- 特定の資源提供機関及び計算機資源を選択したソフトウェア情報の絞り込み

- ソフトウェアに関する情報を追加

- ソフトウェアの概要説明[NEW]
- 計算機資源毎のインストール情報や利用条件等
- 利用課題の多いソフトウェアを中心に関連情報を案内[NEW]
- 各ソフトウェアを利用した課題の利用報告書へのリンク[NEW]
- 各資源提供機関のソフトウェア関連ページへのリンク[NEW]