SOP-GPU

Manual

v.1.07

Artem Zhmurov

(zhmurov@gmail.com)

Valeri Barsegov

(Valeri_Barsegov@uml.edu)

April 11, 2011

1 General notes on SOP model

In SOP model, each residue is described using a single interaction center (C_α atom). The potential energy function of a protein conformation V, specified in terms of the coordinates $\{\mathbf{r}\} = \mathbf{r_1}, \mathbf{r_2}, \dots \mathbf{r_N}$, is given by

$$\begin{aligned} V = V_{FENE} + V_{NB}^{ATT} + V_{NB}^{REP} = \\ -\sum_{covalent} \frac{k}{2} R_0^2 \log \left(1 - \frac{\left(r_{ij} - r_{ij}^0\right)^2}{R_0^2}\right) \\ + \sum_{native} \varepsilon_h \left[\left(\frac{r_{ij}^0}{r_{ij}}\right)^{12} - 2\left(\frac{r_{ij}^0}{r_{ij}}\right)^{6} \right] \\ + \sum_{repulsive} \varepsilon_r \left(\frac{\sigma}{r_{ij}}\right)^{6}. \end{aligned} \tag{1}$$

In Eq.1, the distance between two residues, i and j, is r_{ij}, while r_{ij}^0 is its value in native (PDB) structure. The finite extensible nonlinear elastic (FENE) potential V_{FENE} describes covalent bonds (both backbone and disulfide bonds) with tolerance value in the change of covalent bond $R_0 = 2\AA$. We used Lennard-Jones potential (V_{NB}^{ATT}) to account for the noncovalent interactions that stabilize the native state (i.e. native interactions). Two non-covalently linked residues form native contact if, in native state (PDB structure), the distance between their C_α atoms is less then cut-off distance (simple Go definition) or/and centers of heavy atoms of amino-acids side chains are within some cut-off distance (full Go definition). Most common definition is a simple Go with a cut-off distance of $8\AA$. The value of ε_h quantifies the strength of non-bonded interactions. More detailed explanatioin on definition of native contacts can be found in Section 3.2. All other pairs of atoms (i.e. that neither connected covalently nor form a native contact) are considered to have repulsive Lennard Jones potential V_{REP}^{NB} ($\sim 1/r^6$). The strength or repulsive potential is described by parameter $\varepsilon_r = 1 kcal/mol$, repulsive distance is equal to the length of amino-acid $\sigma = 3.8\AA$.

1.1 Using SOP-GPU program

As a parameter, configuration filename should be passed to the program. From this configuration file, program determines where to find system topology and coordinates files and what simulation protocol should be performed. Several output files are saved during computation: coordinates in dcd format, energy output in .dat (tab-separated) format. Intermediate ("restart") coordinates aloso saved in pdb format. There are also separate output files for pulling (indentation) parameters as described in sections 4.1.5 (4.1.7). Energy .dat output file has the following columns:

1. Current simulation timestep.

2. Average Maxwell-Boltzmann temperature (in $kcall/mol$).

3. Total polential energy (in $kcall/mol$).

4. Potential energy of native interactions (in $kcall/mol$).

5. Potential energy of repulsive Lennard-Jones interactions (in $kcall/mol$).

6. Total potential energy of Lennard-Jones interactions (attractive and repulsive) (in $kcall/mol$).

7. Potential energy of covalent bonds (in $kcall/mol$).

8. Number of native contacts not ruptured.

9. Gyration radius (not saved in a `capsid` mode).

1.2 Pulling simulations

Pulling simulations were designed to mimick force-ramp AFM experiment. When pulling is enabled, some of the particles are fixed (to represent the part of the molecule that is adsorbed on the mica surface), some others are attached to the spring (representing residues adsorbed on the tip of the canteliver). The other end of this spring is being pulled with constant velocity (cantilever base or "chip"). While the spring is extended, the force is increasing, inducing mechanical unfolding in the protein.

When pulling is enabled, programm will save additional .dat file with pulling data, including current end-to-end distance and forces. This file has following columns:

1. Current simulation timestep.

2. Absolute value of the end-to-end distance (in $\AA$).

3. Projection of the end-to-end distance on pulling vector (in $\AA$).

4. Absolute value of the canteliver spring force ($k_s \Delta x$, in $kcal/mol\AA$).

5-7. Force vector components (x, y and z, in $kcal/mol\AA$).

For pulling simulation parameters see section 4.1.5.

1.3 Heating simulations

Although coarse-grained models are known for describing heat-induced unfolding badly, SOP-model still can provide good qualitatative results. When heating is on, temperature of water bath (i.e. strength of random force) is increased gradualy during the simulation procedure. Heating parameters are described in section **??**.

1.4 Forced indentation

Indentation mode adds a canteliver and surface model to the system. Cantilever tip is desribed as a repulsive Lennard-Jones ($1/r^6$) sphere, attached to a spring. The other end of this spring (canteliver base or "chip") is moving with constant velocity, increasing external force, acting on the tip. Tip coordinates undergo Brownian dynamics with given friction coefficient (usually about 100 times bigger then that of a single amino-acid). Resulting force, acting on a tip is a summ of a harmonic force from canteliver extension and all molecular forces of tip-protein interactions. Because it moves slowly, the equation of motion of the canteliver tip are integrated using timestep bigger then that used for amino-acids (tip is shifted only after some number of steps). This also allows one to increase the performance of the program. Mica surface is modelled as repulsive Lennard-Jones ($\sim 1/r^6$) surface. Initial position, radius and friction coefficient of the canteliver tip, initial coordinates and velocity of a canteliver chip, position of the surface should be specified in configuration file when indentation simulation protocol is used (see section 4.1.7). Indentation values will be saved into .dat file with following columns:

1. Current simulation timestep.

2. Distance traveled by cantiliver base (in $\AA$).

3. Average molecular force acting on a canteliver tip projected onto chip movement direction (in $kcal/mol\AA$).

4. Average absolute value of a molecular force, actinf on a cantaliver tip (in $kcal/mol\AA$).

5. Absolute value of the canteliver spring force at a given step ($k_s\Delta x$, in $kcal/mol\AA$).

6. Absolute value of the canteliver spring force average ($\overline{k_s\Delta x}$, in $kcal/mol\AA$).

7-9. Molecular force vector components (x, y and z, in $\AA$).

10-12. Current canteliver tip coordinates (x, y and z, in $\AA$).

13-15. Current canteliver base coordinates (x, y and z, in $\AA$).

2 Units

Characteristic time for underdamped motion of spherical particle of mass m and radius a with energy scale ε_h is $\tau_L = \sqrt{ma^2/\varepsilon_h}$. Then, $\alpha = \zeta m/\tau_L$ is the friction coefficient for the spherical particle (ζ = dimensionless friction coeff). $\alpha = 6\pi\eta a$ is the Stokes-Einstein friction for spherical particle of radius a (in A) in liquid with viscosity η. Therefore $\zeta = (6\pi\eta a^2)/\sqrt{(}m\varepsilon_h)$. Bulk water viscosity $\eta = 0.01gs^{-1}cm^{-1}$. In the program $\zeta = 50$. This was obtained for $a \sim 5\AA$ and $m \sim 3 \times 10^{-22}g$ (mass of a residue) and using bulk water viscosity. In general, a varies between $3.8\AA$ to $5\AA$, while m varies between $3 \times 10^{-22}g$ to $5 \times 10^{-22}g$. In simulations $a = 3.8\AA$. Because of the fact that ζ depends on ε_h, every time ε_h changed, valid m value shold be found. This value should give $\zeta = 50$. Example: for $\varepsilon_h = 1kcal/mol$ from the above equation for ζ we find that $m = 4.3 \times 10^{-22}g$ which is a valid value. For $\varepsilon_h = 1.5kcal/mol$, we get $m = 3 \times 10^{-22}g$ which is still a valid value. After finding the mass m, we can go back to the expression for τ_L and get its value. For example, for $\varepsilon_h = 1kcal/mol$ we get $\tau_L = 3ps$ while for $\varepsilon_h = 1.5kcal/mol$, we get $\tau_L = 2ps$.

$\tau_H = \zeta \varepsilon_h \tau_L / kT$ is the characteristic time for the overdamped Langevin dynamics. To get it in ps, both ε_h and kT need to be in the same units. Because ε_h is in $kcal/mol$, kT is also in $kcal/mol$ ($T = 300K$ corresponds to $0.6kcal/mol$). Therefore an elementary time step for the simulation is $h \times \tau_H$ in ps.

The pulling speed is $v_f = \Delta x / (n_{av} \times h \times \tau_H)$ where Δx is a displacement during n_{av} timesteps in $\mathring{A}$. The force is in $kcal/(mol\mathring{A})$, to get a force in pN, value in $kcal/(mol\mathring{A})$ should be multiplied by 70.

3 Topology

3.1 File

There are three types of interactions in SOP model: covalent interactions, native interactions and repulsive pairs. All these shold be listed in Gromacs-style topology file (.top). There are four sections in SOP topology file: `[ atoms ]`, that lists all C_α atoms in the system (including information about residue ID, residue and chain name, etc.), and three sections that correspond to three types of interactions: `[ bonds ]` for covalent bonds, `[ native ]` for native interactions and `[ pairs ]` for repulsive pairs. `[ atoms ]` section simply follows the Gromacs-style atom description, including only C_α atoms. The last three sections cosist of the list of lines, each of which has indexes of interacting particles, function type and set of specific parameters. Particle indexes correspond to internal programm indexes and start from 0, function type column is set to 1 for all pairs and ignored, parameters are specific for each interaction type as described below. More details on file format can be found in Gromacs Manual.

3.1.1 `[ bonds ]` section.

Typical `[ bonds ]` section include lines, simular to the following:

```
[ bonds ]
;  ai    aj funct           c0            c1            c2            c3
     0     1     1    3.81188
     1     2     1    3.77232
     2     3     1    3.79319
```

Covalent bonds include backbone interactions and disulfide $S-S$ bonds. Potential energy function term that correspond to covalent bonds interaction is described by V_{FENE} in Eq. 1, where summation is made over all lines in `[ bonds ]` section of the topology file, i and j correspond to indexes of the particles listed in the line, distance r_{ij} is computed from particles coordinates, r^0_{ij} is the distance between two corresponding C_α atoms in native state (PDB file), listed as the first parameter in the line (column `c0`, see sample listing above).

3.1.2 `[ native ]` section.

```
[ native ]
```

```
;  ai    aj funct            c0            c1            c2            c3
    5     9     1    5.85792    1.50000
    5    10     1    7.06482    1.50000
    5    35     1    6.64479    1.50000
```

In SOP model, native interactions (V_{NB}^{ATT}) are described by full Lennard-Jones potential (Eq. 1). Each term in the summ correspond to one line in `[ native ]` section. Apart from indexes of interacting particles, equilibrium distance r_{ij}^0 (column `c0`) and strength coefficient, ε_h (column `c1`), are listed. r_{ij}^0 is the distance between C_α atoms in native state (PDB file), value of ε_h is usualy between 1.0 and $1.5 kcal/mol$ and should be chosen to provide the best fit to experimental data.

3.1.3 `[ pairs ]` section.

```
[ pairs ]
;  ai    aj funct            c0            c1            c2            c3
    0     2     1
    0     3     1
    0     4     1
```

Pairs section correspond the third term in Eq. 1. There is no pair-specific parameters in this section, only indexes are listed. Note, that this list scales as $\sim N^2$ with the system size N, and saving all possible repulsive pairs in the topology file would lead to very large files. In SOP-GPU programm, this section is used only for small systems and when trajectory massive-production is employed. When large system is simulated, dual-range cut-off algorithm is utilized and only pairs withing bigger cut-off are kept (pairlist). Pairlist is updated using exclusion principle: olny those pairs that are withing cut-off distance but not in the list of exluded pairs added. Verlet list is built from this pairlist based on smaller cut-off distance and used when potential function and forces are computed. Excluded pairs are those already listed in `[ covalent ]` and `[ native ]` sections.

3.2 Creation of topology file

3.2.1 General description

Originally, creation of the topology was included in SOP-GPU programm. However, to simlify the internal logic, it was moved to a separate programm (sop-top). As with the main programm, configuration file should be passed as the first parameter to sop-top. This configuration file can use the same features as configuration file for SOP-GPU, as described in Section 4.1. Topology is created from original (full-atomic) PDB file using `ATOM` and `SSBOND` entries of the later. All C_α atoms are added into `[ atoms ]` section of topology file generated. Backbone connectivity and disulfide bonds along with their equilibrium (PDB) distances are collected into `[ bonds]` section. Native contacts are determined based on two cut-off distances. First refer to maximum $C_\alpha - C_\alpha$

distance for two amino-acids in native contact (simple Go definition), second one is the cut-off for the minimal distance of two heavy atoms in corresponding amino-acids side-chains (full Go definition). Along with the indexes of amino-acids i and j, PDB distance r^0_{ij} and value of ε_h are saved for each pair qualify. ε_h can be specified as constant value for all native pairs or can be taken from occupancy of beta columns of original PDB. In later case, geometric average of two values listed for amino-acids i and j is taken. All pairs that did not qualify for covalent bond or native contacts are added to `[ pairs ]` section of the topology file. For the large systems, this is done based on a cut-off distance to decrease the size of the topology file. In this case list of repulsive pairs is updated during execution of the programm.

3.2.2 Parameters for topology creation

- `structure` <*filename*>

  Type: Name of .pdb file.

  Status: Required.

  Purpose: Initial (full-atomic) PDB file.

- `topology` <*filename*>

  Type: Name of .top file.

  Status: Required.

  Purpose: Output filename for topology file.

- `coordinates` <*filename*>

  Type: Name of .pdb file.

  Status: Required.

  Purpose: Output filename for corse-grained .pdb file with C_α coordinates only.

- `covalent_cutoff` <*cut-off distance in Å*>

  Type: Float.

  Status: Optional, dafault value is $10\AA$.

  Purpose: Cut-off for SS-bonds. If the distance is larger then this value, residues will not be covalently linked even if listed in SSBOND pdb entry.

- `SS_cutoff` <*cut-off distance in Å*>

  Type: Float.

  Status: Required.

Purpose: If the minimal distance between two heavy atoms in two amino-acids side-chains is less then this parameters, amino-acids are in a native contact.

- `R_limit_bond` <*cut-off distance in Å*>

  Type: Float.

  Status: Required.

  Purpose: Cut-off for $C_\alpha - C_\alpha$ distance for two amino-acids to be in a native contact.

- `SC_limit_bond` <*cut-off distance in Å*>

  Type: Float.

  Status: Required.

  Purpose: If the minimal distance between two heavy atoms in two amino-acids side-chains is less then this parameters, amino-acids are in a native contact.

- `eh` <ε_h >

  Type: Float or 'O'/'B'.

  Status: Required.

  Purpose: Strength of native interactions (ε_h). If number is specified, ε_h is constant for all native interactions in the system. If 'O' ('B') is set, value will be taken from occupancy (beta) column of initial PDB-file and geometric average will be saved for interacting pair.

- `pairs_threshold` <*cut-off distance in Å*>

  Type: Float.

  Status: Optional.

  Purpose: Cut-off distance for repulsive pairs. If distance is larger then this value, pair will not be listed in `[ pairs ]` section. If not specified, all pairs will be listed. Should not be used with many-runs-per GPU approach.

3.2.3 Parameters to create tandems

- `createTandem` <*yes/no*>

  Type: "yes" or "no".

  Default: "no".

  Status: Optional.

  Purpose: If yes, tandem of proteins will be created.

- `linkerLength` <*number of residues*>

  Type: Integer.

  Status: Required if createTandem is "yes".

  Purpose: Number of residues to be added between monomers in tandem.

- `monomerCount` <*number of monomers*>

  Type: Integer.

  Status: Required if createTandem is "yes".

  Purpose: Number of monomers in tandem.

- `tandemDirection` <*direction*>

  Type: "endToEnd" or "vector".

  Status: Required if createTandem is "yes".

  Purpose: Specify direction in which monomers are connected to form a tandem.

- `fixedEnd` <*residue ID*>

  Type: Integer.

  Status: Required if tandemDirection is "endToEnd".

  Purpose: First end for end-to-end direction.

- `pulledEnd` <*residue ID*>

  Type: Integer.

  Status: Required if tandemDirection is "endToEnd".

  Purpose: Second end for end-to-end direction.

- `tandemVectorX, tandemVectorY, tandemVectorZ` <*vector coordinates*>

  Type: Float.

  Status: Required if tandemDirection is "vector".

  Purpose: Direction in which monomers are connected to form a tandem, specified by vector coordinates.

3.2.4 Parameters to add covalent linkers

- `covLinkersCount` <*number of different linkers*>

  Type: Integer

  Default: 0.

  Status: Optional.

  Purpose: How many linker should be added to the system.

- `covLinker1`, `covLinker2`, ... <*linker definition*>

  Type: <ResName1><ResID1>-<ResName2><ResID2>.

  Status: Required if covLinkersCount is not 0.

  Purpose: Description of the linkers should be added to the system. Contains of three-letter codes of two residues to link and their residue id's. Three-letter codes and IDs should be the same as in pdb file. For instance, value of 'LYS169-ASN356' will create linkers between residue LYS169 and ASN356 if those are within certein cutoff distance (see next parameter).

- `covLinkersCutoff` <*cut-off distance*>

  Type: Float.

  Status: Optional. Default value is 10 *Å*

  Purpose: Defines the cut-off for covalent linkers.

4 Input parameters file

4.1 General features

Input parameters file contains all the simulation parameters listed as tab or space separated pairs of name and value. Remarks are allowed using '#' character. To simplify creation of multiple configuration files, parameters values support macroses. This is usefull, e.g. when specifying output files to avoid overwriting when multiple trajectories are prodused. Any parameter name in the file can be used as macros, additional macroses can be added using same name-value sintaxis as for parameters. To use macros, parameters value should include other parameter name (or macros name) inside < and > characters. For example, the following will result in value for the output file name equal to '3.*dcd*':

```
run 3
DCDfile <run>.dcd
```

4.1.1 Device parameters

- `device` <*device ID*>

  Type: Integer.

  Status: Required.

  Purpose: Id of NVidia card to run simulations on. Use "deviceQuerry" from NVidia SDK to check devices.

- `block_size` <*integer*>

  Type: Integer.

  Status: Optional.

  Purpose: Set the number of threads per block. By default, set to 128. Can be specified for every potential individualy, using `block_size_covalent`, `block_size_native`, `block_size_pairs`, `block_size_pairlist` and `block_size_possiblepairs`.

- `max_covalent`, `max_native`, `max_pairs` and `max_possiblePairs` <*integer*>

  Type: Integer.

  Status: Required.

  Purpose: Set the maximum number of pairs per residue for covalent, native, pairs and possiblePairs interactions.

4.1.2 Common parameters

- `name` <*protein name*>

  Type: String.

  Status: Optional.

  Purpose: Name, assigned to the protein. Used mostly for files naming.

- `stage` <*stage name*>

  Possible values: `equil`, `pull`, `heat`, `minim`, `indent`.

  Status: Optional.

  Purpose: Specifying stage will assign a simulation protocol. `equil` stage stands for equilibrium (quenching) simulation. `pull` will force programm to fix some residues, pulling others and will require additional parameters to be set as it is described in section 4.1.5. `heat` stands for heating simulation and described in section 4.1.6. Using `minim` value will turn off the random force. `indent` stands for indentation and further described in **??**. `stage` parameter can be omitted and simulation protocol can be assigned using `pulling on`, `heating on` and/or `minimization on` lines as it is described in corresponding sections. If neither is specifyed, equilibrium simulations will be helded.

- `topology` <*filename*>

  Type: Path to the file.

  Status: Required.

  Purpose: Path to topology file in Gromacs-like format.

- `coordinates` <*filename* >

  Type: Path to the file.

  Status: Required.

  Purpose: Path to coordinates file in PDB format.

- `numsteps` <*steps count* >

  Type: Integer.

  Status: Required.

  Purpose: Length of the simulation in timesteps.

- `timestep` <*time* >

  Type: Float.

  Status: Required.

  Purpose: Length of the timestep.

- `seed` <*random seed* >

  Type: Long integer.

  Status: Required.

  Purpose: Initial random seed used for random force. Actual seed is computed by adding `run` or `firstrun` (whichever is defined) to this value.

- `run` <*trajectory number* >

  Type: Integer.

  Status: Required if `firstrun` and `runnum` are not specified.

  Purpose: Trajectory number when running only one trajectory per GPU ("one-run-per-GPU approach"). Usually used for files naming. Alternatively, `firstrun` and `runnum` can be used.

- `firstrun` <*trajectory number* >

  Type: Integer.

  Status: Required if `run` is not specified.

  Purpose: Number of first trajectory when "using many-runs-per-GPU" approach.

- `runnum` <*number of trajectories* >

  Type: Integer.

  Status: Required if `firstrun` is specified.

  Purpose: Quantity of trajectories to run simultaniously on one GPU when using "many-runs-per-GPU" approach. Trajectories from `firstrun` to `firstrun` + `runnum` will be started. Note, that in this case all output files requre "`<run>`" macros, so that the output data will be saved into different files for different trajectories.

- `mode` <*output mode* >

  Type: Default or "capsid".

  Status: Optional.

  Purpose: Slightly changes .dat output format for the capsid.

4.1.3 Force-field parameters

- `temperature` <*temperature*>

  Type: Float.

  Units: $kcal/mol$.

  Default: $0.6 kcal/mol \sim 300K$.

  Status: Required.

  Purpose: Set the temperature to heat bath (random force).

- `zeta` <ζ *value*>

  Type: Float.

  Units: Unitless.

  Default: 50.0.

  Status: Required.

Purpose: Friction coefficient for amino-acid. For the spherical particle: $\zeta = 6\pi\eta a^2/\sqrt{m\varepsilon_h}$, where $\eta = 0.01g\cdot(s\cdot cm)^{-1}$ - bulk water viscosity, $m \sim 3\cdot 10^{-22}g$ - average mass of an amino-acid residue, $a = 3.8\mathring{A}$ - length of amino-acid amide bond, ε_h - average native contacts strength factor (hydrophobisity). ε_h is taken from topology file and usually between 0.9 and 1.5.

- `R_limit` <*R_0 value*>

  Type: Float.

  Units: $\mathring{A}$.

  Default: 2.0.

  Status: Optional.

  Purpose: R_0 parameter for FENE potential (Eq. 1).

- `kspring_cov` <*spring constant*>

  Type: Float.

  Units: $kcal/mol\mathring{A}$.

  Default: 20.0.

  Status: Optional.

  Purpose: Spring constant, k, for FENE potential (Eq. 1).

- `a` <*distance*>

  Type: Float.

  Units: $\mathring{A}$.

  Default: 3.8.

  Status: Optional.

  Purpose: Default distance between atoms. This is a value that used in repulsive LJ potential for amino-acids (Eq. 1).

- `el` <*energy factor*>

  Type: Float.

  Units: $kcal/mol$.

  Default: 1.0.

  Status: Optional.

  Purpose: Energy factor for repulsive Lennard-Jones potential (Eq. 1).

4.1.4 Pairs lists

- `pairs_cutoff` <*distance in Å* >

  Type: Float.

  Units: *Å*.

  Default: 20*Å*.

  Status: Optional.

  Purpose: Pairs cut-off for repulsive Lennard-Jones. If distance between particles is more then this value, force is not computed.

- `pairlist_cutoff` <*distance in Å* >

  Type: Float.

  Units: *Å*.

  Default: 20*Å*.

  Status: Optional.

  Purpose: Pairlist cut-off. If the distance between particles is less then this value, pair is added to pairs (Verlet) list.

- `pairlist_threshold` <*distance in Å* >

  Type: Float.

  Units: *Å*.

  Default: 200*Å*.

  Status: Optional.

  Purpose: Cut-off value for generatind the list of possible pairs. This list is generated based on exclusion principle - if the pair form covalent bond or native contact it does not qualify.

- `pairlist_freq` <*number of steps*>

  Type: Float.

  Units: Unitless.

  Default: 1000.

  Status: Optional.

  Purpose: How often pirs (Verlet) list will be renewed.

- `possiblepairs_freq` <*number of steps*>

Type: Float.

Units: Unitless.

Default: 10000.

Status: Optional.

Purpose: How often list of possible pairs is to be renewed.

4.1.5 Pulling parameters

- `pulling` <*on/off* >

  Type: "on"/"off".

  Status: Optional.

  Default: "off".

  Purpose: Wheather or not external forcec should be applied. Same as using "pull" stage.

- `fixedEnd`, `pulledEnd` <*residue ID* >

  Type: Integer.

  Status: Required.

  Purpose: Residue IDs to determine end-to-end distance.

- `fixed_beads` <*number of fixed residues* >

  Type: Integer.

  Status: Required.

  Purpose: Number of residues to be fixed.

- `pulled_beads` <*number of pulled residues* >

  Type: Integer.

  Status: Required.

  Purpose: Number of beads to be pulled.

- `fixed1`, `fixed2`, ... <*residue IDs* >

  Type: Integer.

  Status: Required if `fixed_beads` > 0.

  Purpose: IDs of fixed residues. Should more or equal to `fixed_beads`. If more, only `fixed_beads` first residues listed will be fixed in simulation.

- `pulled1`, `pulled2`, … <*residue IDs* >

  Type: Integer.

  Status: Required if `pulled_beads` > 0.

  Purpose: IDs of pulled residues. Should more or equal to `pulled_beads`. If more, only `pulled_beads` first residues listed will be pulled in simulation.

- `deltaX` <*pulling speed* >

  Type: Float.

  Status: Required.

  Purpose: Pulling speed. Position of the spring base ("cantilever chip") will be shifted by `deltaX` every `nav` steps. Actual pulling speed can be computed as `deltaX`/`nav` × `timestep`.

- `pullDirection` <*direction* >

  Type: "`endToEnd`" or "`vector`"

  Status: Required.

  Purpose: Direction of pulling. If "`endToEnd`", spring base ("cantilever chip") will move toward end-to-end vector, computed from positions of `fixedEnd` and `pulledEnd` residues. If "vector" is set, `pullVector` should be specified (see below).

- `pullVector` <*vector* >

  Type: Vector.

  Status: Required if `pullDirection` is "`vector`".

  Purpose: Vector of pulling. x, y and z coordinates should be space or tab separated.

- `pullOutput` <*filename* >

  Type: Name of a dat file.

  Status: Optional.

  Default: "pull.<name>_<author><run>_<stage>.dat"

  Purpose: Filename for pulling output.

4.1.6 Heating parameters

- `heating` <*on/off* >

  Type: "on"/"off".

Status: Optional.

Default: “off”.

Purpose: Wheather or not temperature should be changed on the course of simulations. Same as using “heat” stage.

- `initialT` <*temperature*>

  Type: Float.

  Units: *kcal/mol*.

  Status: Required.

  Purpose: System temperature at the beginning of simulations.

- `deltaT` <*temperature increment*>

  Type: Float.

  Units: *kcal/mol*.

  Status: Required.

  Purpose: Value of the temperature increment that will be added to the initial temperature every `tempFreq` steps (see below).

- `tempFreq` <*number of steps*>

  Type: Integer.

  Status: Required.

  Purpose: Frequency of changing the temperature.

4.1.7 Indentation parameters

- `indentation` <*on/off* >

  Type: “on”/“off”.

  Status: Optional.

  Default: “off”.

  Purpose: Indentation switch. Same can be done by using “indent” stage.

- `indentationChip` <*position* >

  Type: Vector.

  Status: Required.

Purpose: Initial position of canteliver spring base (i.e. canteliver "chip").

- `indentationTip` <*position* >

  Type: Vector.

  Status: Required.

  Purpose: Initial position of canteliver tip center.

- `indentationDirection` <*direction* >

  Type: Vector.

  Status: Required.

  Purpose: Direction of canteliver base movement.

- `indentationTipR` <*radius* >

  Type: Float.

  Status: Required.

  Purpose: Radius of canteliver tip.

- `indentationTipKs` <*spring constant* >

  Type: Float.

  Units: *kcal/mol*.

  Status: Required.

  Purpose: Spring constant of a canteliver.

- `indentationTipA`, `indentationTipB`, `indentationTipSigma`, `indentationTipEl` <*dimentionless* >

  Type: Float.

  Units: Dimentionless.

  Status: Optional.

  Purpose: Shape of the Lennard-Jones potential for the cantilever tip. The potential will be computed as $\varepsilon_l(A \times (\frac{\sigma}{r})^{12} + B \times (\frac{\sigma}{r})^6)$. Standart Lennard-Jones potential is $A = 1$, $B = -2$. Will override `indentationEl` and `indentationSigma`

- `indentationSurfA`, `indentationSurfB`, `indentationSurfSigma`, `indentationSurfEl` <*dimentionless* >

  Type: Float.

  Units: Dimentionless.

Status: Optional.

Purpose: Shape of the Lennard-Jones potential for the surface.

- `indentationDeltaX` <*speed* >

  Type: Float.

  Status: Required.

  Purpose: Speed of the canteliver movement. Every `nav` steps, canteliver base will be shifted on this distance.

- `indentationFixTrans` <*yes/no* >

  Type: Boolean.

  Status: Required.

  Purpose: If enabled, tip will only move along the indentation direction (i.e. it will be fixed transversaly).

- `indentationEl` <*energy factor* >

  Type: Float.

  Status: Required.

  Purpose: Repulsive energy factor for Lennard-Jones potential.

- `indentationSigma` <*distance* >

  Type: Float.

  Status: Required.

  Purpose: Repulsive distance for Lennard-Jones potential. Note that potential is shifted to the surface of the canteliver tip sphere.

- `indentationTipZeta` <*friction coefficient* >

  Type: Float.

  Status: Required.

  Purpose: Canteliver tip friction coefficient.

- `indentationSurfaceR0` <*position* >

  Type: Vector.

  Status: Required.

  Purpose: Position of mica surface.

- `indentationSurfaceN` <*direction* >

  Type: Vector.

  Status: Required.

  Purpose: Mica plane normal vector.

- `indentationDiscreteSurf` <*yes/no* >

  Type: Boolean.

  Status: Optional.

  Purpose: If enabled, mica will be represented as a set of interacting beads, positioned according to the surface representation (parameters `indentationSurfaceSize` and `indentationSurfaceStep`). Otherwise, potential will be continuous (the function fill be computed using the normal vector).

- `indentationPairsCutoff` <*cut-off distance* >

  Type: Float.

  Status: Optional.

  Purpose: Cut-off distance for the pairs list if the surface is represented as a set of discreete beads.

- `indentationRetractionStep` <*step* >

  Type: Integer.

  Status: Optional.

  Purpose: If specified, direction of indentation will be reversed on this step.

- `indentationMoveSurface` <*yes/no* >

  Type: yes/no.

  Status: Optional. Default value is "no"

  Purpose: If "yes", surface will be moving rather than cantilever.

- `indentationOutput` <*filename* >

  Type: Name of a dat file.

  Status: Optional.

  Default: "indentation.<name>_<author><run>_<stage>.dat"

  Purpose: Filename for indentation output.

- `indentationOutputFreq` <*number of steps* >

Type: Integer.

Status: Required.

Purpose: Number of steps before indentation output is printed into the file and on the screen.

- `indentationShowTipSurf` <*yes/no* >

  Type: "yes" or "no".

  Status: Optional.

  Default: "no".

  Purpose: If "yes", programm will save coordinates of canteliver tip and chip as well as some points representing mica surface. Usefull for representation purposes. Tip will be represented as two particles with chain 'T' in pdb file, surface particles will have chain 'M'.

- `indentationSurfaceSize` <*number of points* >

  Type: Integer.

  Status: Optional.

  Default: 51.

  Purpose: Number of surface points to show in each direction. Total number of points saved will be a square value of this.

- `indentationSurfaceStep` <*distance* >

  Type: Float.

  Status: Optional.

  Default: 10.

  Purpose: Distance between points on mica representation.

- `indentationSurfConnectFile` <*filename* >

  Type: String, filename.

  Status: Optional.

  Purpose: Filename to dump "connect" script that can be used in VMD to show the mica as a surface rather than set of points.

- `indentationCantLength` <*distance* >

  Type: Float.

  Status: Required.

  Purpose: Length of the cantilever in its representation. Only needed when `indentationShowTipSurf` is enabled.

4.1.8 Output parameters

- `nav` <*number of steps*>

  Type: Integer.

  Status: Required.

  Purpose: Number of steps to take average on. Used when temperature is computed. Also when pulling is enabled, every `nav` steps external force will be incremented. Also used when canteliver tip is propagated during indentation simulations.

- `reffilename` <*filename*>

  Type: Path to pdb file.

  Status: Required.

  Purpose: Name of the reference file that can be used to load structure into VMD.

- `outputtiming` <*number of steps*>

  Type: Integer.

  Status: Required.

  Purpose: Number of steps before dumping energy output.

- `outputname` <*filename*>

  Type: Path to dat file.

  Status: Required.

  Purpose: Name of dat file to write energy output in. If file exists, it will be overwritten.

- `dcdfreq` <*number of steps*>

  Type: Integer.

  Status: Required.

  Purpose: Number of steps before dumping coordinates into dcd file.

- `DCDfile` <*filename*>

  Type: Path to dat file.

  Status: Required.

  Purpose: Name of dcd file to write coordinates output in. If file exists, it will be overwritten.

- `restartfreq` <*number of steps*>

Type: Integer.

Status: Required.

Purpose: Number of steps before dumping restart pdb/conf files.

- `restartname` <*filename*>

  Type: Path to file.

  Status: Required.

  Purpose: Extensionless name of restart files. .pdb and .coord extension will be added for restart coordinates/configuration parameters.

- `finalcoord` <*filename*>

  Type: Path to pdb file.

  Status: Required.

  Purpose: Filename for the final coordinates.