# SOP-GPU

## Manual

Artem Zhmurov

(zhmurov@gmail.com)

Valeri Barsegov

(Valeri_Barsegov@uml.edu)

University of Massachusetts at Lowell

November 24, 2010

# 1 The Self Organized Polymer (SOP) model

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

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

In Eq.1, the distance between any 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 the covalent bonds (both backbone and disulfide bonds) with the tolerance for changing the covalent bond distance $R_0$=2Å. We used the Lennard-Jones potential ($V_{NB}^{ATT}$) to account for the non-covalent interactions that stabilize the native state (native interactions). Two non-covalently linked residues form a native contact if, in the native state (PDB structure), the distance between their $C_\alpha$-atoms is less than the cut-off distance (simple Go definition) or the distance between the centers of heavy atoms in their side chains are within some cut-off distance (full Go definition). The most common definition is a simple Go model, in which the cut-off distance of 8Åis used. The value of $\varepsilon_h$, which is the central quantity in the SOP model, quantifies the strength of the non-bonded (non-covalent) interactions. More detailed explanation of the definition of native contacts is presented in Section 3.2. All the other pairs of atoms, that are neither connected covalently nor form a native contact, are treated using the repulsive Lennard-Jones potential $V_{REP}^{NB}$ ($\sim 1/r^6$). The strength or repulsive interactions is described by parameter $\varepsilon_r$=1$kcal/mol$, and the repulsive distance is equal the length of an amino acid $\sigma$=3.8Å(along the backbone).

## 1.1 Using SOP-GPU program

As a parameter, the name of the configuration file should be passed to the program. Starting from this configuration file, the program determines where to find information about the molecular topology (topology file) and $C_\alpha$ coordinates (coordinate file), and what simulation protocol should be used (see below). Several output files are saved during the computation. These include coordinates in the dcd format, and the energy output in the .dat (tab-separated) format. The intermediate ("restart") coordinates are also saved in the pdb format. In addition, there are separate output files for pulling parameters or indentation parameters as described in Sections 4.1.5 (4.1.7). The energy .dat output file has the following data columns:

**1.** Current simulation step.

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

**3.** Total potential energy (in $kcal/mol$).

**4.** Potential energy due to native interactions (in $kcal/mol$).

**5.** Potential energy due to repulsive Lennard-Jones interactions (in $kcal/mol$).

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

**7.** Potential energy of the covalent bonds (in $kcal/mol$).

**8.** Total number of native contacts remaining.

**9.** Gyration radius, which is not saved in a `capsid` mode (in Å).

## 1.2 Pulling simulations

Pulling simulations, are designed to mimic the actual force-ramp Atomic Force Microscopy (AFM) based dynamic force measurements. When the pulling option is enabled, some of the particles are constrained. This represent the portion of the molecule under the study, which is adsorbed on the mica surface. Some others particles are attached to the harmonic spring. This corresponds to the residues that are adsorbed on the cantilever tip in the actual AFM experiments *in vitro*. The other end of the harmonic spring, which represents the cantilever base or "chip", is pulled with a constant velocity (force-ramp). While both the spring and the biomolecule extend, the force is gradually increasing, leading to a mechanical unfolding reaction that occurs in the protein.

When the pulling mode is enabled, the program saves additional .dat file with pulling data, including the current value of the end-to-end distance and unfolding force. This file has the following data columns:

**1.** Current simulation step.

**2.** Absolute value of the end-to-end distance (in Å).

**3.** Projection of the end-to-end distance on the pulling vector (in Å).

**4.** Absolute value of the cantilever spring force $k_s \Delta x$ (in $kcal/mol$Å).

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

For pulling simulation parameters see Section 4.1.5.

## 1.3 Heating simulations

The SOP model can be used to describe, at a qualitative level, thermal unfolding of biomolecules. When the "heating" mode is enabled, the temperature of the stochastic thermostat, described by the strength of random forces, gradually increases in a linear fashion during the simulation run. In the "heating" mode, one can analyze the increase of the total energy versus the temperature increase to obtain the heat capacity of the system in question. All the heating parameters are described in Section 4.1.6.

### 1.4 Forced indentation

The “forced indentation” mode adds to the system a cantilever tip and an adsorptive (mica) surface. The cantilever tip is described as a repulsive Lennard-Jones ($\sim 1/r^6$) sphere, attached to a harmonic spring. The other end of the spring (cantilever base or “chip”) is moving with a constant velocity $v_f$ (force-ramp), which results in the increase of an external mechanical force, acting on the tip. The tip coordinates undergo Brownian dynamics with a given friction coefficient (typically, $\sim$100-times larger than that for the a single amino acid residue ($C_\alpha$-bead). The resulting force, acting on the tip is the sum of the harmonic force from the cantilever extension and all the molecular forces due to the tip-protein interactions. Because the tip moves slowly, the equation of motion of the cantilever tip can be integrated using longer time steps than the time step used to propagate Langevin dynamics of a biomolecule, i.e. the tip is shifted once over some number of integration time steps. This also allows to increase the performance of the SOP-GPU program. The mica surface is modeled as repulsive Lennard-Jones ($\sim 1/r^6$) surface. The initial position, radius and friction coefficient for the cantilever tip, the initial coordinates and velocity of the cantilever chip, and, finally, the position of the surface should be specified in configuration file when the indentation protocol is used (see Section 4.1.7). The indentation output is saved into the .dat file, which has the following data columns:

**1.** Current simulation step.

**2.** Distance traveled by the cantilever base (in Å).

**3.** Average molecular force acting on the cantilever tip projected onto the chip direction of motion (in $kcal/mol$Å).

**4.** Average absolute value of the molecular force, acting on the cantilever tip (in $kcal/mol$Å).

**5.** Absolute value of the cantilever spring force at a given step $k_s \Delta x$ (in $kcal/mol$Å).

**6.** Running average of the absolute value of the cantilever spring force $\overline{k_s \Delta x}$ (in $kcal/mol$Å).

**7-9.** Vector components of molecular force in $x$, $y$ and $z$ directions (in Å).

**10-12.** Current cantilever tip coordinates $x$, $y$ and $z$ (in Å).

**13-15.** Current cantilever base coordinates $x$, $y$ and $z$ (in Å).

## 2 Physical units

Characteristic time for the underdamped motion of a spherical particle of mass $m$ and radius $a$ and energy $\varepsilon_h$ is given by $\tau_L{=}\sqrt{ma^2/\varepsilon_h}$. Then, $\alpha{=}\zeta m/\tau_L$ is the friction coefficient for the particle ($\zeta$ is the dimensionless friction coefficient). Here, $\alpha{=}6\pi\eta a$ is the Stokes-Einstein friction for the spherical particle of radius $a$ (in Å) in a liquid characterized by viscosity $\eta$. Therefore, $\zeta{=}(6\pi\eta a^2)/\sqrt{(}m\varepsilon_h)$, where the bulk water viscosity is $\eta{=}0.01gs^{-1}cm^{-1}$. In the SOP-GPU program, $\zeta{=}50$. This was obtained for $a{\sim}5$Å- the average size of an amino acid residue, and

$m$~$3\times10^{-22}g$ - the average mass of a residue, and viscosity of the bulk water. In general, $a$ varies between 3.8Åand 5Å, while $m$ varies between $3\times10^{-22}g$ and $5\times10^{-22}g$. In the simulations, $a$=3.8Å. Because of the fact that $\zeta$ depends on $\varepsilon_h$, when $\varepsilon_h$ changed, $m$ should be recalculated. This value should give $\zeta$=50. For example, for $\varepsilon_h$=1$kcal/mol$ we find from the above equation for $\zeta$ that $m$=$4.3\times10^{-22}g$. For $\varepsilon_h$=1.5$kcal/mol$, we get $m$=$3\times10^{-22}g$, which is still a valid value. After finding the mass $m$, we can go back to the expression for $\tau_L$ to obtain its value. For example, for $\varepsilon_h$=1$kcal/mol$, we get $\tau_L$=3$ps$, whereas for $\varepsilon_h$=1.5$kcal/mol$, we get $\tau_L$=2$ps$.

Continuing, $\tau_H$=$\zeta\varepsilon_h\tau_L/kT$ is the characteristic time for the overdamped Langevin dynamics. To get this time in $ps$, both $\varepsilon_h$ and $kT$ need to be in the same units. Because $\varepsilon_h$ is given in $kcal/mol$, $kT$ should also be given in $kcal/mol$, Here, the room temperature $T$=300$K$ corresponds to 0.6$kcal/mol$. Therefore an elementary time step for the simulation is given by $h\tau_H$, and is in $ps$. Computationally, the pulling speed is expressed as $v_f$=$\Delta x/(n_{av}h\tau_H)$, where $\Delta x$ (in Å) is the displacement of the cantilever tip in $n_{av}$ simulation steps. The mechanical force is in $kcal/(mol$Å$)$, and to obtain the force values in $pN$, the values given in $kcal/(mol$Å$)$ should be multiplied by a factor of 70.

# 3 Topology

## 3.1 File

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

### 3.1.1 `[ bonds ]` section.

As an example, the `[ bonds ]` section include the following lines:

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

```
    2     3     1             3.79319
```

The covalent bonds include the backbone interactions and the disulfide ($S{-}S$) bonds. In Eq. 1, the potential energy function term that correspond to the covalent bonds is described by $V_{FENE}$. Here, the summation runs over all the lines in `[ bonds ]` section in the topology file, and $i$ and $j$ correspond to indices of the particles listed in the line, the distance $r_{ij}$ is computed from the particles coordinates, $r_{ij}^0$ is the distance between two corresponding $C_\alpha$-atoms in the native state (in the PDB file), listed as the first parameter in the line column `c0` (see sample listing above).

### 3.1.2 `[ native ]` section.

As an example, the `[ native ]` section include the following lines:

```
[ 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 the SOP model, native interactions ($V_{NB}^{ATT}$) are described by the full Lennard-Jones potential (see Eq. 1). Each term in the summation correspond to one line in the `[ native ]` section. Apart from the indices for interacting particles, this section of the topology file lists the equilibrium distance $r_{ij}^0$ (column `c0`) and $\varepsilon_h$ (column `c1`). Here, $r_{ij}^0$ is the distance between the $C_\alpha$-atoms in the native state (in the PDB file); the typical numerical values of $\varepsilon_h$ are between $1.0 kcal/mol$ and $1.5 kcal/mol$, and it should be chosen to provide the best agreement between the experimental force spectra and the simulated force-extension profiles.

### 3.1.3 `[ pairs ]` section.

As an example, the `[ pairs ]` section include the following lines:

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

The pairs section of the topology file correspond to the third term in Eq. 1. There are no pair-specific parameters in this section of the file, only the indices are listed. Of note, the size of the list scales as $\sim N^2$ with the system size $N$, and hence, saving all possible repulsive pairs in the topology file would lead to a very large file. In the SOP-GPU program, this section of the file is used only for small systems when multiple-run-per-GPU approach is employed. When a large system is simulated, a dual cut-off algorithm is utilized, where only pairs withing the additional larger cut-off are kept (pairlist). The pair list is updated using the exclusion principle: only those pairs that are withing the cut-off distance but not in the list of excluded pairs are added to the pair list. The Verlet list is built from the pair list based on smaller cut-off distance, and is used only when the potential energy function and forces are computed. Here, the excluded pairs of residues are those that are already already listed in `[ covalent ]` and `[ native ]` sections of the topology file.

## 3.2 Creating topology file

### 3.2.1 General description

Originally, creating the topology file was included in the SOP-GPU program. However, to simplify the internal logic, it was later moved into a separate program ("sop-top" functionality). In the main program, the configuration file should be passed as the first parameter to the "sop-top". This configuration file has similar features as the configuration file in the SOP-GPU program (see Section 4.1). The molecular topology is created from the original (full-atomic) PDB file, using the `ATOM` and `SSBOND` entries in the PDB file. All the $C_\alpha$-atoms are then added into the `[ atoms ]` section of the topology file generated. The backbone connectivity and disulfide bonds along with their equilibrium (PDB) distances are accumulated in the `[ bonds]` section of the topology file. The native contacts are determined based on two cut-off distances. The first cut-off distance refers to the maximum $C_\alpha$–$C_\alpha$ (non-covalent) distance for any two amino acid residues forming a native contact (so called "simple Go" definition). The second cut-off is the minimal distance between any two heavy atoms in the side-chains of the corresponding amino acid residues (so called "full Go" definition). Along with the indices for amino acids $i$ and $j$, the PDB distance $r_{ij}^0$, and the value of $\varepsilon_h$ are saved for each pair that passes either the first or second cut-off test. Here, $\varepsilon_h$ can be specified as constant (same) for all the native pairs in the system, or it can be taken from the occupancy of beta columns in the original PDB file. In the later case, the geometric average of two values listed for amino acids $i$ and $j$ is taken. All the pairs that do not qualify for the covalent bonds or for the native contacts are accumulated in the `[ pairs ]` section of the topology file. For a large system, this is done based on the cut-off distance to decrease the size of the topology file. In this case, a list of repulsive residue pairs is updated during the execution of the program.

### 3.2.2 Parameters for creating molecular topology

The following parameters can be used to create the topology file for the system in question, using the "sop-top" utility:

- `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 the $C_\alpha$-atomic coordinates only.

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

  **Type:** Float.

  **Status:** Optional, default value is 10Å.

  **Purpose:** Cut-off for SS-bonds. If the distance is larger then this value, residues will not be covalently linked even when they are listed in the SSBOND entry in the PDB file.

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

  **Type:** Float.

  **Status:** Required.

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

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

  **Type:** Float.

  **Status:** Required.

  **Purpose:** Cut-off for the $C_\alpha - C_\alpha$-atom distance for two amino acid residues forming 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 side-chains is less than this parameters, then the corresponding amino acid form a native contact.

- `eh` $<\varepsilon_h>$

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

  **Status:** Required.

  **Purpose:** Strength of native interactions ($\varepsilon_h$). If a numerical value is specified, then $\varepsilon_h$ will be constant for all native interactions in the system. However, when 'O' ('B') is chosen, the $\varepsilon_h$ value will be taken from the occupancy (beta) column in the initial PDB file, and the geometric average will be saved for each interacting pair of residues.

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

  **Type:** Float.

  **Status:** Optional.

  **Purpose:** Cut-off distance for repulsive pairs of residues. If the distance is larger than this value, then this pair of residues will not be listed in the `[ pairs ]` section of the topology file. When not specified, all the pairs will be listed. This parameter should not be used when the many-runs-per GPU approach is employed.

### 3.2.3 Parameters for creating multimeric proteins (protein tandems)

Since the AFM-based dynamic force experiments are also utilized to perform force measurements on protein tandems, the "sop-top" utility has the functionality to create linear tandems of proteins from individual protein domains. This can be readily done by using the following parameters:

- `createTandem` <*yes/no*>

  **Type:** "yes" or "no".

  **Default:** "no".

  **Status:** Optional.

  **Purpose:** If yes, a tandem of proteins will be created, where protein domains are separated by flexible linkers.

- `linkerLength` <*number of residues*>

  **Type:** Integer.

  **Status:** Required if createTandem is "yes".

  **Purpose:** Length of a linker: the number of residues to be added between adjacent monomers (proteins) forming the tandem.

- `monomerCount` <*number of monomers*>

**Type:** Integer.

**Status:** Required if createTandem is “yes”.

**Purpose:** Total number of monomers, or protein domains $D$, $n$ forming the tandem $(D)_n$.

- `tandemDirection` <*direction*>

  **Type:** “endToEnd” or “vector”.

  **Status:** Required if createTandem is “yes”.

  **Purpose:** Specify the direction in which individual monomers are connected to form the initial tandem structure.

- `fixedEnd` <*residue ID*>

  **Type:** Integer.

  **Status:** Required if tandemDirection is “endToEnd”.

  **Purpose:** First end of the molecule (protein tandem) for the calculation of the end-to-end direction and distance.

- `pulledEnd` <*residue ID*>

  **Type:** Integer.

  **Status:** Required if tandemDirection is “endToEnd”.

  **Purpose:** Second end of the molecule (protein tandem) for the calculation of the end-to-end direction and distance.

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

  **Type:** Float.

  **Status:** Required if tandemDirection is “vector”.

  **Purpose:** Direction in which monomers are connected to form the initial tandem structure, specified by vector coordinates.

### 3.2.4 Parameters for adding covalent linkers

The following parameters can be used to add covalent linkers to a protein system. Note, that the disulfide bonds are added automatically if the corresponding “SSBOND” entries are present in the initial (all-atom) .pdb file.

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

  **Type:** Integer

  **Default:** 0.

**Status:** Optional.

**Purpose:** Specifies how many linkers should be added to the system (protein tandem).

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

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

  **Status:** Required if covLinkersCount is not 0.

  **Purpose:** Contains three-letter codes of two residues to be linked and their residue id's. Three-letter codes and IDs should be the same as in the PDB file. For example, 'LYS169-ASN356' will create linkers between residue LYS169 and ASN356 if they are within the cut-off distance (see next parameter).

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

  **Type:** Float.

  **Status:** Optional. Default value is 10Å

  **Purpose:** Defines the cut-off distance for covalent linkers.

# 4 Input parameters

## 4.1 General description

The file with the input parameters contains all the simulation parameters listed as tab- or space-separated pairs of the parameter name and value. Writing remarks is allowed using the '#' character. To simplify a process of creation of multiple configuration files, parameter values support macros. This is useful, e.g., when specifying the output files to avoid overwriting when multiple independent trajectories are produced on a single GPU device concurrently (many-runs-per-GPU approach). Any parameter name in the configuration file can be used as macros, and, when needed, additional macros can be added using same name-value syntax for parameters. To use macros, parameter values should include other parameter name (or macros name) inside the < and > characters. For example, the following will result in the name for the output file ending with '3.*dcd*':

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

### 4.1.1 Device parameters

The following parameters can be used to select the GPU device as well as to control thread block size and memory usage:

- `device` <*device ID*>

  **Type:** Integer.

**Status:** Required.

**Purpose:** The ID of the NVidia graphics card used in simulations. 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, the SOP-GPU program will set its value to 128. This parameter can be specified for every potential energy term in the force field individually, 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 the covalent bonds, native contacts, and for pairs and possiblePairs interactions.

### 4.1.2 Main parameters

The general information about the system in question and the simulation protocol can be specified using the following parameters:

- `name` <*protein name*>

  **Type:** String.

  **Status:** Optional.

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

- `stage` <*stage name*>

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

  **Status:** Optional.

  **Purpose:** Specifying the name of the stage will assign the simulation protocol used in the SOP-GPU program. Here, `equil` stage is for equilibrium (quenching) simulation (production runs); `pull` will force the program to fix some residues, while pulling some other residues, and it will require additional parameters to be specified as described in Section 4.1.5; `heat` is for heating simulation described in Section 4.1.6; using `minim` value will turn off the generation of random forces (deterministic dynamics);

`indent` stands for the simulations of forced indentation described in Section 4.1.7; `stage` parameter can be omitted and the simulation protocol can be assigned using `pulling on`, `heating on` and/or `minimization on` lines as described in the corresponding sections of this manual. If neither stage is specified, the SOP-GPU program will run equilibrium simulations.

- `topology` <*filename*>

  **Type:** Path to the file.

  **Status:** Required.

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

- `coordinates` <*filename* >

  **Type:** Path to the file.

  **Status:** Required.

  **Purpose:** Path to the coordinate file in the PDB format.

- `numsteps` <*steps count* >

  **Type:** Integer.

  **Status:** Required.

  **Purpose:** Total length of the simulation trajectory in steps.

- `timestep` <*time* >

  **Type:** Float.

  **Status:** Required.

  **Purpose:** Length of the simulation timestep in *ps*.

- `seed` <*random seed* >

  **Type:** Long integer.

  **Status:** Required.

  **Purpose:** Initial random seed used to generate the distribution of random forces. The 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:** Number of the trajectory when running only one trajectory on a single GPU device (so called "one-run-per-GPU" approach). It is used for file naming. Alternatively, `firstrun` and `runnum` can be used.

- `firstrun` <*trajectory number* >

  **Type:** Integer.

  **Status:** Required if `run` is not specified.

  **Purpose:** Number of the first trajectory when running multiple trajectories on a single GPU device (so called "many-runs-per-GPU" approach).

- `runnum` <*number of trajectories* >

  **Type:** Integer.

  **Status:** Required if `firstrun` is specified.

  **Purpose:** Total number of independent trajectories obtained simultaneously on one GPU when using the "many-runs-per-GPU" approach. Trajectories ranging from `firstrun` to `firstrun` + `runnum` will be started at the same time. Note, that in this case all the output files requre "`<run>`" macros, so that the output data could be saved into different files for each trajectory.

- `mode` <*output mode* >

  **Type:** Default or "capsid".

  **Status:** Optional.

  **Purpose:** Changes the .dat output format for the viral capsid.

### 4.1.3 Force-field parameters

Numerical constants for the various energy terms entering Eq. (1) for the SOP force-field are specified using the following parameters:

- `temperature` <*temperature*>

  **Type:** Float.

  **Units:** $kcal/mol$.

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

  **Status:** Required.

  **Purpose:** Set the temperature of the heat bath (stochastic thermostat).

- `zeta` <$\zeta$ *value*>

**Type:** Float.

**Units:** Unitless.

**Default:** 50.0.

**Status:** Required.

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

- `R_limit` <$R_0$ *value*>

  **Type:** Float.

  **Units:** Å.

  **Default:** 2.0.

  **Status:** Optional.

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

- `kspring_cov` <*spring constant*>

  **Type:** Float.

  **Units:** $kcal/mol$Å.

  **Default:** 20.0.

  **Status:** Optional.

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

- `a` <*distance*>

  **Type:** Float.

  **Units:** Å.

  **Default:** 3.8.

  **Status:** Optional.

  **Purpose:** Default distance between the $C_\alpha$-atoms. This is a value used in the repulsive Lennard-Jones potential (Eq. 1).

- `el` <*energy factor*>

  **Type:** Float.

**Units:** *kcal/mol*.

**Default:** 1.0.

**Status:** Optional.

**Purpose:** Energy scale for the repulsive Lennard-Jones potential (Eq. 1).

### 4.1.4 Pair lists

Use the following parameters to manage pair list and Verlet list generation.

- `pairs_cutoff` <*distance in Å*>

  **Type:** Float.

  **Units:** Å.

  **Default:** 20 Å.

  **Status:** Optional.

  **Purpose:** Cut-off distance for pairs of residues described by the repulsive Lennard-Jones potential (Eq. 1). If the distance between the $C_\alpha$-particles is larger than this value, the resulting force is not computed.

- `pairlist_cutoff` <*distance in Å*>

  **Type:** Float.

  **Units:** Å.

  **Default:** 20 Å.

  **Status:** Optional.

  **Purpose:** Cut-off distance used to generate a pairlist. If the distance between the $C_\alpha$-particles is shorter than this value, the pair of residues is added to the pairlist (Verlet list).

- `pairlist_threshold` <*distance in Å*>

  **Type:** Float.

  **Units:** Å.

  **Default:** 200 Å.

  **Status:** Optional.

  **Purpose:** Cut-off distance for generating the list of all possible pairs of residues. This list is generated based on the exclusion principle: if a pair of the $C_\alpha$-particles forms a covalent bond or a native contact, it does not enter the list of all possible pairs.

- `pairlist_freq` <*number of steps*>

**Type:** Float.

**Units:** Unitless.

**Default:** 1000.

**Status:** Optional.

**Purpose:** Specifies how often the pairlist (Verlet list) is regenerated.

- `possiblepairs_freq` <*number of steps*>

  **Type:** Float.

  **Units:** Unitless.

  **Default:** 10000.

  **Status:** Optional.

  **Purpose:** Specifies how often the list of all possible pairs of residues is regenerated.

### 4.1.5 Parameters for pulling simulations

To control pulling simulations, one should use the following parameters:

- `pulling` <*on/off* >

  **Type:** "on"/"off".

  **Status:** Optional.

  **Default:** "off".

  **Purpose:** Define whether or not an external mechanical force should be applied to the system. Same as using "pull" stage.

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

  **Type:** Integer.

  **Status:** Required.

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

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

  **Type:** Integer.

  **Status:** Required.

  **Purpose:** Total number of the $C_\alpha$-residues to be fixed.

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

**Type:** Integer.

**Status:** Required.

**Purpose:** Total number of $C_\alpha$-beads to be pulled.

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

  **Type:** Integer.

  **Status:** Required if `fixed_beads` $> 0$.

  **Purpose:** Residue IDs of the fixed $C_\alpha$-residues. It should greater or equal to `fixed_beads`. If it is greater, only the `fixed_beads` first residues listed will be constrained in the course of simulations.

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

  **Type:** Integer.

  **Status:** Required if `pulled_beads` $> 0$.

  **Purpose:** Residue IDs of the pulled $C_\alpha$-residues. It should greater or equal to `pulled_beads`. If greater, only the `pulled_beads` first residues listed will be pulled in the course of simulations.

- `deltaX` <*pulling speed* >

  **Type:** Float.

  **Status:** Required.

  **Purpose:** Pulling speed. The position of the spring base ("cantilever chip") will be shifted by the amount specified by `deltaX` every `nav` steps of iteration. The actual pulling speed can be computed as `deltaX`/`nav`×`timestep`.

- `pullDirection` <*direction* >

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

  **Status:** Required.

  **Purpose:** Direction of application of an external mechanical force. If "`endToEnd`" is chosen, then the spring base ("cantilever chip") will move in the direction of the end-to-end vector, computed from positions of the `fixedEnd` and `pulledEnd` residues. If "vector" is used, `pullVector` should be specified as well (see below).

- `pullVector` <*vector* >

  **Type:** Vector.

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

**Purpose:** Vector specifying the direction of application of a pulling force. Here, the $x$, $y$ 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 the output from the pulling simulations.

### 4.1.6 Parameters for heating simulations

To perform the heating simulations (i.e. thermal denaturation), one should use the following parameters:

- `heating` <*on/off* >

  **Type:** "on"/"off".

  **Status:** Optional.

  **Default:** "off".

  **Purpose:** Specify whether or not the temperature of the stochastic (Langevin) thermostat should be changed in the course of simulations. Same as using the "heat" stage.

- `initialT` <*temperature*>

  **Type:** Float.

  **Units:** *kcal/mol*.

  **Status:** Required.

  **Purpose:** Initial temperature of the system at the beginning of heating simulations.

- `deltaT` <*temperature increment*>

  **Type:** Float.

  **Units:** *kcal/mol*.

  **Status:** Required.

  **Purpose:** Value of the temperature increment used to raise temperature from the initial temperature to some final temperature every `tempFreq` simulation steps (see below).

- `tempFreq` <*number of steps*>

  **Type:** Integer.

  **Status:** Required.

  **Purpose:** Frequency for changing the temperature from the initial temperature to some final temperature.

### 4.1.7 Parameters for indentation simulations

The forced indentation simulations utilize a Lennard-Jones tip-sphere and a surface to model the action of a cantilever tip and a mica surface. The following parameters should be used to set up the indentation simulations:

- `indentation` <*on/off* >

  **Type:** "on"/"off".

  **Status:** Optional.

  **Default:** "off".

  **Purpose:** Specifies whether the indentation force is switched on or off. This can also be achieved by using the "indent" stage.

- `indentationChip` <*position* >

  **Type:** Vector.

  **Status:** Required.

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

- `indentationTip` <*position* >

  **Type:** Vector.

  **Status:** Required.

  **Purpose:** Initial position of the center of a canteliver tip.

- `indentationDirection` <*direction* >

  **Type:** Vector.

  **Status:** Required.

  **Purpose:** Direction of movement of the canteliver base.

- `indentationTipR` <*radius* >

  **Type:** Float.

  **Status:** Required.

  **Purpose:** Radius of the cantilever spherical tip.

- `indentationTipKs` <*spring constant* >

  **Type:** Float.

  **Units:** $kcal/mol$.

**Status:** Required.

**Purpose:** Spring constant of the cantilever.

- `indentationDeltaX` <*speed* >

  **Type:** Float.

  **Status:** Required.

  **Purpose:** Velocity of the cantilever. Every `nav` steps, the cantilever base is shifted by this amount.

- `indentationEl` <*energy factor* >

  **Type:** Float.

  **Status:** Required.

  **Purpose:** Repulsive energy for the Lennard-Jones potential, which specifies the repulsive interactions between the cantilever tip and the $C_\alpha$-particles of the system (viral capsid).

- `indentationSigma` <*distance* >

  **Type:** Float.

  **Status:** Required.

  **Purpose:** Repulsive distance for the Lennard-Jones potential, which specifies th repulsive interactions between the cantilever tip and the $C_\alpha$-particles of the system (viral capsid). The values of this potential are identical on the surface of the cantilever tip-sphere, but change with the distance between the tip surface and the $C_\alpha$-particles of the system.

- `indentationTipZeta` <*friction coefficient* >

  **Type:** Float.

  **Status:** Required.

  **Purpose:** Friction coefficient for the cantilever tip.

- `indentationSurfaceR0` <*position* >

  **Type:** Vector.

  **Status:** Required.

  **Purpose:** Position of the adsorptive (mica) surface underneath the system of interest (viral capsid).

- `indentationSurfaceN` <*direction* >

  **Type:** Vector.

  **Status:** Required.

**Purpose:** Normal vector for the adsorptive surface (mica plane).

- `indentationRetractionStep` <*step* >

  **Type:** Integer.

  **Status:** Optional.

  **Purpose:** If specified, the direction of indentation (approach) will be reversed in this step (retraction).

- `indentationMoveSurface` <*yes/no* >

  **Type:** yes/no.

  **Status:** Optional. Default value is "no"

  **Purpose:** If "yes"is chosen, the adsorptive surface will be moving upward to indent the system of interest rather than the cantilever tip.

- `indentationOutput` <*filename* >

  **Type:** Name of a dat file.

  **Status:** Optional.

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

  **Purpose:** Filename for the indentation output.

- `indentationOutputFreq` <*number of steps* >

  **Type:** Integer.

  **Status:** Required.

  **Purpose:** Number of steps before the next indentation output (energies, coordinates, etc.) is printed into the file and on the computer screen.

- `indentationShowTipSurf` <*yes/no* >

  **Type:** "yes" or "no".

  **Status:** Optional.

  **Default:** "no".

  **Purpose:** If "yes" is chosen, the SOP-GPU program will save the coordinates of the cantilever tip and chip as well as the coordinates representing the adsorptive plane (mica surface). It is useful only for the purposes of representation and visualization. If "yes" is chosen, the cantilever tip will be represented by just two particles within the chain 'T' in the output (.pdb) file, whereas the surface particles will be denoted by the chain 'M'.

- `indentationSurfaceSize` <*number of points* >

  **Type:** Integer.

  **Status:** Optional.

  **Default:** 51.

  **Purpose:** Number of points on the adsorptive surface to be shown in each ($x$ and $y$) direction. This is only for the purposes of representation and visualization. The total number of points saved will be given by the square of this value.

- `indentationSurfaceStep` <*distance* >

  **Type:** Float.

  **Status:** Optional.

  **Default:** 10.

  **Purpose:** Distance between the points representing the adsorptive surface (mica plane). It is useful only for the purposes of representation and visualization.

- `indentationSurfConnectFile` <*filename* >

  **Type:** String, filename.

  **Status:** Optional.

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

### 4.1.8 Parameters for simulation output

The following parameters should be used to control the simulation output:

- `nav` <*number of steps*>

  **Type:** Integer.

  **Status:** Required.

  **Purpose:** Number of steps to obtain the average quantities. It is used when the system temperature is computed. It is also used when pulling simulations are enabled, where every `nav` steps external the mechanical force is incremented, and in the indentation simulations.

- `reffilename` <*filename*>

  **Type:** Path to pdb file.

  **Status:** Required.

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

- `outputtiming` <*number of steps*>

  **Type:** Integer.

  **Status:** Required.

  **Purpose:** Number of steps of the simulation before the next energy output is printed.

- `outputname` <*filename*>

  **Type:** Path to dat file.

  **Status:** Required.

  **Purpose:** Name of the output .dat file to store the computed values for the energy output. If this file already exists, it will be overwritten each time when a new simulation run is (re)started.

- `dcdfreq` <*number of steps*>

  **Type:** Integer.

  **Status:** Required.

  **Purpose:** Number of steps of the simulation before the next output for coordinates is printed into the .dcd file.

- `DCDfile` <*filename*>

  **Type:** Path to dat file.

  **Status:** Required.

  **Purpose:** Name of the .dcd file for the output of coordinates of the $C_\alpha$-particles. If this file already exists, it will be overwritten each time when a new simulation run is (re)started.

- `restartfreq` <*number of steps*>

  **Type:** Integer.

  **Status:** Required.

  **Purpose:** Number of steps before dumping the restart coordinates (.pdb) and configuration files.

- `restartname` <*filename*>

  **Type:** Path to file.

  **Status:** Required.

  **Purpose:** Extension-less name for the restart files. Here, the .pdb and .coord extension will be added by the program for the restart coordinates and configuration parameters.

- `finalcoord` <*filename*>

  **Type:** Path to pdb file.

  **Status:** Required.

  **Purpose:** Filename for reporting the final coordinates of the $C_\alpha$-particles.