**Software Methods and Functions Introduction**

This chapter will introduce the execution steps, main functions and principles behind the PX- MDsim software in detail to help users understand the architecture of the software and how to achieve polymer crosslinking in multi-step simulations. PX- MDsim is mainly used to simulate the polymer crosslinking process, especially for studying the formation process of polyamide membranes. The following are the core steps and functional analysis of the software, combined with the run.sh script and related Python script files.

**1. Preparation and preprocessing of input data**

Before starting the simulation, PX- MDsim first requires the user to input the structure file , force field file of Carboxyl (carboxyl) monomer and Amino (amino) monomer, as well as the force field file of an amide cross-linking monomer in which Carboxyl (carboxyl) monomer and Amino (amino) monomer are connected by an amide bond . Data preparation includes reading the file information of the monomer , converting the file format, checking the validity of the file, and generating the necessary parameters for the initial system. This process is completed by the interactive prompts in the run.sh script. The user needs to enter the standard residue name of the monomer (three capital letters, and the cross-linking monomer is six capital letters) . The script then automatically generates the file name based on the name and processes it .

**1.1 User Input and File Naming**

To facilitate user operation, PX- MDsim provides an interactive input prompt that requires the user to enter the standard residue name of the monomer and automatically recognizes the corresponding structure and force field file name. This process is executed by the run.sh script to ensure that the naming format of all files is consistent and avoid errors in manual operation.

“

*read -p "Enter the Carboxyl monomer name（XXX）: " CarboxylName*

*read -p "Enter the Amino monomer name（XXX）: " AminoName*

”

Based on the user input, the script automatically identifies the corresponding .pdb , .mol2 and .str file names. For example, if the user inputs the carboxyl monomer name as ABC, the identified input files include ABC.pdb ( coordinate file), ABC.mol2 ( structure file with molecular information ) and ABC.str (force field parameter file). Similar files will also be identified for amino monomers.

This automatic file recognition mechanism can effectively reduce errors in the file naming process and ensure that the required files can be found smoothly in subsequent processing steps. After identifying the input file, the system will prompt the user to confirm the input file name to ensure that the operation is correct.

**1.2 File format processing and inspection**

After receiving user input, PX- MDsim will automatically check the format of the input file to ensure that the file is valid and can be used in subsequent simulation processes. Specifically, PX- MDsim will process the .pdb , .mol2 and .str files in turn to ensure that these files contain the correct atomic, bond and molecular information.

**1.2.1 PDB Files**

The PDB (Protein Data Bank) format is a commonly used molecular structure file format that contains the atomic coordinate information of the molecule. In PX- MDsim , the PDB file provided by the user is the initial structure of the monomer molecule, which contains information such as the position of atoms, atomic numbers, and molecule names. With this information, PX- MDsim can construct the three-dimensional structure of the initial system in subsequent molecular dynamics simulations.

**1.2.2 MOL2 File**

MOL2 files are file formats that contain molecular structure information and are usually used to describe the structure, atom type, bond, and other information of small molecule compounds. In PX- MDsim , the MOL2 file provided by the user is used to generate force field parameters. This file not only describes the geometric structure of the molecule, but also contains the bond type and order in the molecule. This is crucial for the subsequent generation of force field parameters and topology files.

**1.2.3 STR File**

STR files are files generated by the CHARMM force field, which contain the molecular topology and force field parameters. In PX- MDsim , STR files need to be preprocessed to extract the core information related to atoms and bonds . During the processing, PX- MDsim uses the process\_str.py script to format the STR file ( process\_str ).

* **Parsing ATOM and BOND information** : The process\_str.py script scans each line in the STR file and extracts the parts related to atoms (ATOM) and bonds ( BOND). For the atom part, the script extracts information such as the atom number, type and position; for the bond part, the script extracts the connecting atoms of the bond and their type. All irrelevant information ( usually introduced due to different charmm force field versions ) will be removed in this step.
* **File formatting** : The script will write the processed information into a new STR file, ensuring that the file contains only the core information of atoms and bonds . The formatted STR file will be used to generate the residue topology file (RTP) for use by GROMACS in subsequent steps.

The purpose of this process is to ensure that the STR file provided by the user is in the correct format and does not contain redundant information, so as to avoid parsing errors or loss of force field parameters in subsequent steps.

**1.3 Generation of the initial system**

After the input files have been checked and processed, PX- MDsim begins to generate the initial structure of the simulation system, including setting the number of molecules, defining the system box size, etc. This step is completed by the InpGene.py script ( InpGene ).

**1.3.1 Setting the box size**

Molecular dynamics simulations are usually performed in a limited space, which is called a simulation box. PX- MDsim will generate a cube or rectangular box based on the number of amino molecules and carboxyl molecules and the box size entered by the user . The InpGene.py script calls the packmol software, which ensures that all molecules can be placed in the box without overlap or unreasonable initial arrangement based on the number of molecules and the box side length provided by the user .

In the InpGene.py script, the user can interactively input the number of monomers and the size of the box. The script will generate an input file containing these molecular coordinates, ensuring that the system is in a reasonable geometric layout when the simulation starts.

“

*box\_size = input("Please enter the side length of the cube box (unit: angstrom):")*

”

**1.3.2 Generating mixed molecular systems**

PX - MDsim requires the user to simulate multiple different carboxyl and amino monomers at the same time. In InpGene.py, the user needs to enter the number of each type of monomer, and the system will generate a mixed molecular system based on the data entered by the user. The script places each molecule in the specified area within the box to ensure a reasonable distribution between different molecular types. The output file of this step is a PDB file containing all monomers, which provides the initial state for subsequent molecular dynamics simulations.

The PDB file records the initial positions of all molecules and the initial relationships between molecules. Through this file, the subsequent topology file generation, force field parameter generation and molecular dynamics simulation can ensure the accuracy of the molecular system.

**1.4 Confirmation and verification of input data**

At the end of the input data preparation, PX- MDsim will show the user the generated file name and parameters, asking the user for final confirmation. This step is critical to ensure the correctness of the input file and reduce potential errors. Through interactive prompts, users can recheck the input monomer name, quantity, and generated file name.

”

*echo "Please confirm the names of input files."*

*echo "Carboxyl monomer files: $pdbFile1, $mol2File1, $strFile1"*

*echo "Amino monomer files: $pdbFile2, $mol2File2, $strFile2"*

*echo "Double monomer str file: $ strFileDouble "*

”

Once the input data is confirmed to be correct, PX- MDsim will enter the next stage, which is the stage of generating molecular topology and force field parameters.

**summary**

PX- MDsim uses a variety of scripts to automate the process from user input to data file processing in input data preparation and preprocessing, ensuring that the generated molecular structure file format is correct and contains the necessary information. Through a user-friendly interactive input interface, PX- MDsim provides accurate initial configuration for the simulation system, reducing the possibility of errors in manual operations. The data processing steps include file format checking, formatting, and initial system generation, all of which lay a solid foundation for subsequent molecular dynamics simulations.

**2. Molecular topology and force field settings**

PX- MDsim uses GROMACS to perform molecular dynamics simulations, so the molecular topology and force field files must be set before simulation .

The setting of molecular topology and force field is one of the key steps in molecular dynamics simulation, which ensures that the structure, interaction, bonding mode and mechanical properties of each molecule in the simulation are correctly defined and simulated. In PX- MDsim software, the generation of molecular topology and the setting of force field parameters involve multiple steps and file processing to ensure the correct simulation of carboxyl and amino monomers. The following will introduce the topology and force field setting process in PX- MDsim in detail.

**1. Generation of topology files**

The topology file describes the composition, bonds, angles, dihedral angles and non-bonded interaction parameters of each molecule in the system, which is essential for molecular dynamics simulation. PX- MDsim uses the CHARMM36 force field, which is widely used for simulation of proteins, lipids and small molecules.

Before starting a simulation, PX- MDsim first generates a topology file for each molecule. This process is divided into the following steps:

* **Formatting STR files** : Users need to provide CHARMM .str files for Carboxyl and Amino monomers . We recommend using the CGenFF tool to easily obtain them . This file contains the structural information of the molecule, including atomic coordinates, bonds, angles, etc. First, PX- MDsim performs preliminary processing on the .str file through the process\_str.py script, retaining only the parts related to atoms (ATOM) and bonds ( BOND) ( process\_str ). This step ensures the correct generation of the subsequent force field file and avoids interference from irrelevant information.
* **Topology and coordinate file generation** : Through the InpGene.py script, PX- MDsim generates an initial PDB file for the system and defines the box size and number of monomers contained in the system. This PDB file is then used in the GROMACS pdb2gmx tool to generate a preliminary topology (TOP) file. The pdb2gmx command automatically generates a topological description of the system based on the atomic information in the PDB file and parameterizes it using the specified force field.

**2. Generation and appending of RTP files**

RTP files (residue topology files) are an important part of the CHARMM force field, which are used to define the internal connections and interactions of each residue in a molecule. In PX- MDsim , generating and appending RTP files is an important step to ensure that the residues for cross-linking reactions can be correctly identified and processed by GROMACS.

* **Generate Residue Topology File (RTP)** : Using the appendrtp4.py script, PX- MDsim extracts molecular information from the .str file entered by the user and generates the corresponding RTP file. The script generates ATOM and BOND blocks based on the input molecular residue names, describing the structural information such as the atom type, bonds, and dihedral angles of each residue. At this point, the RTP file records the topological information of each molecule, including atoms, bonds, angles, dihedral angles, etc. in the residue.
* **Append to force field file** : The generated RTP file needs to be appended to the CHARMM36 force field to ensure that GROMACS can correctly recognize these new residues in the simulation. Through the appendrtp4.py script, the RTP file is appended to the merged.rtp file of the CHARMM36 force field and finally copied to the GROMACS force field directory to ensure that the user-input Carboxyl monomer and Amino monomer are correctly recognized in the simulation .

**3. Parameterization using CGenFF tool**

For molecules containing special functional groups, PX- MDsim uses the CHARMM General Force Field ( CGenFF ) tool for parameterization to ensure that the special atoms and bonds of the molecule can be correctly handled. This process uses the cgenff\_charmm2gmx.py script .

* **Calling the CGenFF tool** : The user-provided .mol2 file (containing the coordinates of the molecule) and the .str file (containing the CHARMM parameters) are passed to the CGenFF tool. The tool automatically generates a GROMACS-compatible topology file (. itp ) for the molecule, as well as additional parameter files (. prm ) that record the force field parameters for special bonds, angles, and dihedrals in the molecule.
* **Integration of topology files** : The generated .itp file will be integrated into the main topology file of the system. Users can use the topology file to control how these parameters are applied to the simulation, ensuring that the interaction parameters of each molecule in the system are correctly described.

**4. Add force field parameters such as bonds, angles, and dihedrals**

After the initial topology file is generated, PX- MDsim also requires the manual addition of new parameters for specific molecules or cross-linking processes to the force field. These parameters include special bond types, angles, and dihedral parameters that are usually not preset in the charmm force field library. In particular, when cross-linking occurs, newly formed bonds need to be introduced into the simulation.

* **Additional force field parameters** : Through the append\_ffbonded.py script, PX- MDsim will add special bonds, angles, dihedrals and other mechanical parameters in the molecule to the ffbonded.itp file. The script will copy and append specific bond types in the source file to the target file to ensure that these bonds can be correctly identified in the simulation and participate in the mechanical calculation of the molecule.

**5. Final topology and coordinate file generation**

When the RTP file and force field parameters have been set, PX- MDsim generates the final topology and coordinate files of the system through the GROMACS tool:

* **pdb2gmx command** : Use the pdb2gmx command to generate a system topology file (TOP) and a coordinate file (GRO) containing all molecules and force field parameters. In this step, all molecules in the system will be mapped to the CHARMM36 force field, and combined with the previously generated topology and force field parameters, the system can accurately perform subsequent molecular dynamics simulations.
* **Attached NDX file** : The ndx\_generater.py script is used to generate a molecular index file ( NDX ), which records the numbers of specific atoms or residues in the molecule, making it easy to perform selective operations in subsequent simulations, such as freezing specific atoms .

**6. Update of functional group identification and force field parameters**

Before the cross-linking reaction simulation, PX- MDsim prepares for cross- linking by identifying the functional groups in the molecule . Use Identify\_atom\_C.py and Identify\_atom\_N.py to identify the atoms of the carboxyl and amino functional groups, respectively ( Identify\_atom\_C ) and ( Identify\_atom\_N ). This functional group information is updated in PXLink.py and applied to the formation of amide bonds during the cross-linking process.

In addition, the get\_charge.py script will obtain the charge changes of the relevant bonding atoms after cross-linking and update the charge distribution information in the cross-linked molecule based on the charge information in the force field file input by the user , which is composed of a Carboxyl monomer and an Amino monomer connected by an amide bond. This information is crucial for the charge calculation of the force field, ensuring that the effect of charge transfer on molecular interactions during the cross-linking process can be correctly simulated.

**summary**

The topology and force field settings in PX- MDsim are implemented through a series of automated scripts, from input file processing, force field parameter generation to functional group identification, ensuring that the structure and interaction of each molecule can be accurately displayed in molecular dynamics simulation. In particular, through the CGenFF tool and force field additional scripts, PX- MDsim can flexibly handle complex cross-linking reactions and the generation of special bonds, laying a solid foundation for subsequent polymer cross-linking simulations.

**3. Functional group identification and parameter update**

PX- MDsim software, the identification and parameter update of functional groups are key steps to achieve molecular cross-linking reactions. By identifying specific functional groups (carboxyl and amino groups) in molecules, the software can determine which molecules can react, as well as the atomic sites where cross-linking reactions occur, and provide the required force field parameters and charge information for cross-linking simulations. This step ensures that the chemical bond formation, bond length, angle, and charge changes during the cross-linking process can be correctly simulated and recorded. The following is a detailed introduction to the process.

**1. Purpose of functional group labeling**

Functional group identification refers to identifying the functional groups in a molecule that can participate in chemical reactions by parsing the molecular structure file. In PX- MDsim , the two main functional groups involved are carboxyl and amino . They are the reaction centers for generating amide bonds, and the cross-linking process is achieved by forming a covalent bond (CN bond) between the carbon atom in the carboxyl group and the nitrogen atom in the amino group.

Functional group labeling helps in the following aspects:

* **Locating reaction sites** : By identifying the carboxyl and amino atoms in the molecule, the software can accurately locate the sites that may participate in the cross-linking reaction.
* **Force field parameter update** : Based on the identified functional groups, the software needs to update the relevant force field parameters to ensure the accuracy of chemical bond formation and charge transfer during the cross-linking process.
* **Generate charge transfer information** : Functional group labeling also involves charge redistribution. After cross-linking, the charge of atoms in the molecule will change, which needs to be accurately reflected in the simulation.

**2. Labeling of carboxyl functional groups**

carboxyl group is the carbon atom provider in the cross-linking reaction. PX- MDsim first identifies the atoms in the carboxyl group through the Identify\_atom\_C.py script, including carbon (C), oxygen (O) and other atoms (R) connected to the carboxyl group. These functional group atoms play a key role in the subsequent cross-linking process and are also the parts that need to be focused on, such as the determination of reaction sites, changes in bond angle parameters, charge transfer, and calculation of cross-linking degree. The script uses the following steps ( Identify\_atom\_C ):

* **Parsing BOND and ATOM lines : The script reads information** related to atoms and bonds from the input .str file , parses the bond information using the parse\_bond\_lines () function, and generates a connection map of atoms for subsequent functional group identification.
* **Functional group identification algorithm** : By traversing the generated atomic connection graph, the identify\_carboxy\_groups () function specifically looks for pairing relationships between carbon atoms (C) and oxygen atoms (O), determining which carbon atoms are part of the carboxyl group and which are not . At the same time, it also identifies hydrogen atoms and other atoms (R) connected to the carboxyl group.
* **Output functional group type** : After identification, the script will output the type of each atom, such as Type\_Carboxy\_C (carbon atom in carboxyl), Type\_Carboxy\_O (oxygen atom in carboxyl), Type\_Hydroxy\_O (oxygen atom in hydroxyl) and Type\_Hydroxy\_H (hydrogen atom in hydroxyl). This information will be further passed on In the subsequent PXLink module, it is used to set the force field for cross-linking reactions.

After identification, these atom types are dynamically replaced in the PXLink.py script so that they can be used for bond generation in subsequent simulations.

**3. Labeling of amino functional groups**

Similarly, the nitrogen atom in the amino group is another reactive site in the CN bond during crosslinking. PX- MDsim uses the Identify\_atom\_N.py script to identify the nitrogen atom (N), hydrogen atom (H) and other atoms ( R ) connected to the amino group .

* **Parsing of amino atoms** : The identify\_atom\_N.py script first reads the bond and atom information in the .str file and builds an atom connection diagram. The identify\_amine\_groups () function identifies the nitrogen atom in the amino group and its surrounding hydrogen atoms and other atoms (R).
* **Functional Group Identification** : This function will find the hydrogen atoms attached to the nitrogen atoms and further identify the other atoms (R) attached to the amino group. The script will then output the types of amino atoms identified, including Type\_Amine\_N (nitrogen atom in the amino group ), Type\_Amine\_H (hydrogen atom in the amino group), and Type\_Amine\_R (other atoms attached to the amino group).
* **Updating PXLink.py** : Once identified, these atom types are also dynamically updated into PXLink.py via the sed command for use in cross-linking reactions.

**4. Identification of the bonding atoms of amide bonds in amide cross-linking monomers**

In addition to amino monomers and carboxyl monomers, the software also needs to mark the bonding atoms that form amide bonds in amide cross-linking monomers to obtain the atom type after the cross-linking reaction occurs, which is used to obtain the charge change and cross-linking degree . The identification of amide bond bonding atoms is completed by the identify\_double.py script.

* **Parsing amide bonds** : The script reads the atom and bond information in the .str file and identifies the carbon atoms (C) and nitrogen atoms (N) that are connected by amide bonds. This amide bond information is then used to record new atom types, called NewType\_C and NewType\_N , representing the types of carbon and nitrogen atoms that form amide bonds after the cross-linking reaction .
* **Updated crosslink information** : These new atom types will also be updated in PXLink.py to ensure that appropriate parameters are used for simulations during crosslink reactions.

**5. Update of charge parameters**

During the cross-linking process, carboxyl and amino functional groups react chemically to form amide bonds (CN bonds), which will lead to changes in the atomic charges in the molecule. To ensure the accuracy of the simulation, PX- MDsim will recalculate these charges and update them to the force field.

* **Charge difference calculation** : The get\_charge.py script extracts the charge information of each residue by reading the .str file provided by the user, and calculates the difference in charge before and after the reaction through the previously identified functional group atoms and amide bond atoms ( get\_charge ). For example, the charge of the carbon atom in the carboxyl group and the nitrogen atom in the amino group may change slightly after the amide bond is formed, and these charge changes will affect the accuracy of subsequent simulations.
* **Charge Update** : The get\_charge.py script outputs these charge differences and applies them to the self.delta\_charge\_C and self.delta\_charge\_N variables in the PXLink.py script. These charge differences are used to adjust the charge distribution of the relevant atoms during the reaction to ensure that the charge transfer and electrostatic interactions in the simulation can be accurately described.

**6. Atom counting and topology updating**

To ensure the consistency of the number of atoms and the correctness of the topology file during the reaction, PX- MDsim counts the atoms in the molecule before and after the reaction through the count\_atom.py script ( count\_atom ). This script reads the atomic information in each .str file, calculates the number of atoms in each monomer, and updates this information to the force field.

* **Atom counting** : count\_atom.py will count the number of atoms in the carboxyl monomer and the amino monomer respectively, which is very important for the subsequent application of force field parameters and topology file generation.
* **Topology file update** : The script will replace these atomic numbers into the topology settings section in PXLink.py to ensure that the molecular structure described in the force field is consistent with the molecules actually involved in the reaction.

**summary**

Through the identification of carboxyl and amino functional groups, the recognition of double bond structures, the redistribution of charges and the processing of atomic counts, PX- MDsim achieves the precise positioning of functional groups and the dynamic update of force field parameters in molecular cross-linking simulations. This process ensures the correctness of the chemical reactions in the simulation and provides accurate molecular structures and interaction parameters for subsequent cross-linking reaction simulations.

**4. Molecular dynamics simulation**

Molecular dynamics simulation (MD) is the core component of PX- MDsim software. By performing energy minimization, equilibrium and dynamic simulation of cross-linking processes on molecular systems, PX- MDsim can simulate the detailed process of polymer cross-linking and generate cross-linking structures that meet actual physical and chemical conditions. The following will introduce in detail the various steps of molecular dynamics simulation in PX- MDsim software and the implementation principles behind them.

**1. Objectives of Molecular Dynamics Simulation**

PX- MDsim is to simulate the cross-linking reaction of polymers. By performing energy minimization, equilibrium simulation and kinetic simulation on the system, the software can simulate the formation process of chemical bonds in the actual cross-linking process and ensure that the molecular system achieves reasonable physical stability before and after the reaction.

During the cross-linking reaction, two molecules (usually carboxyl and amino groups) are linked together by forming an amide bond. This process involves not only the formation of new chemical bonds, but also the readjustment of structure and charge. Therefore, molecular dynamics simulations must not only ensure that the reaction proceeds, but also maintain the overall equilibrium and thermodynamic stability of the system.

**2. System initialization and energy minimization**

Before the molecular dynamics simulation begins, PX- MDsim will initialize the system and ensure that the molecular structure in the system reaches a low energy state through energy minimization (EM).

* **Generation of initial coordinate and topology files** : Before the cross-linking reaction begins, PX- MDsim first generates an initial coordinate file ( .gro ) and a topology file (.top) based on the monomer molecular structure provided by the user. These files contain the molecular and atomic positions and force field parameters in the system. Through these files generated by the pdb2gmx command, PX- MDsim provides the initial state for subsequent simulations.
* **Energy minimization** : After generating the initial system, PX- MDsim will first perform energy minimization (EM) on the system. This process is completed by the mdrun command combined with the energy minimization parameter file ( minim.mdp ) . The purpose of energy minimization is to eliminate unnatural high-energy configurations in the system, such as too short bonds or too large molecular overlaps, so that subsequent dynamics simulations can be performed under physically reasonable initial conditions. When performing energy minimization, PX- MDsim can choose to freeze certain molecules to prevent them from causing infinite extension of the structure in subsequent simulations due to the influence of periodic boundary conditions .

After energy minimization, the molecules in the system will be in a relatively stable state, thus laying the foundation for subsequent equilibrium simulations.

**3. NVT balance simulation**

After energy minimization, PX- MDsim will perform NVT (constant temperature and constant volume) equilibrium simulation on the system to make the system reach thermodynamic equilibrium at the specified temperature.

* **Temperature control** : NVT simulation is a molecular dynamics simulation of a molecular system at a constant volume and constant temperature. In PX- MDsim , this step is implemented by the GROMACS mdrun command and the nvt\_frozen.mdp parameter file ( example\_run\_script ). The parameter file defines the simulation temperature (usually 300K) and the time evolution of the system at this temperature. By controlling the temperature, PX- MDsim ensures that the system gradually reaches a thermal equilibrium state under the specified conditions.
* **Generate velocity** : In NVT equilibrium simulation, PX- MDsim needs to generate initial velocity for each atom in the system. The gen-vel option in the parameter file is set to yes, indicating that at the beginning of the simulation, a velocity that conforms to the specified temperature distribution is generated for each atom, thereby simulating the thermal motion in the system. At this time, the molecular system will go through a period of thermal equilibrium to ensure that the temperature and structure of the molecules have reached a stable state.
* **NVT Continue Simulation** : In some cases, if no suitable reaction pair (formation of C-N bonds) is found during the simulation, PX- MDsim will continue the NVT simulation using the nvt\_frozen\_continue.mdp parameter file without regenerating the velocities. This process is intended to ensure that the system can continue to evolve stably when no reaction occurs ( example\_run\_script ).

**4. Cross-linking reaction simulation**

PX- MDsim is to simulate the cross-linking reaction process of polymers, that is, to form amide bonds (CN bonds) through chemical reactions between carboxyl groups and amino groups. After performing NVT equilibrium simulation, the software enters the simulation stage of the cross-linking reaction.

* **Finding reactive pairs** : PX- MDsim uses distance criteria to find carboxyl and amino groups that may participate in the reaction. Using the CN\_dist () function, the software calculates the distance between each pair of carboxyl carbon atoms and amino nitrogen atoms in the system ( example\_run\_script ). If the distance is less than a user-defined threshold (controlled by the dist parameter, usually between 0.3 and 0.5 nanometers), the pair of molecules will be considered a candidate pair that can react.
* **Generate amide bonds** : Once a suitable reaction pair is found, PX- MDsim will generate a new chemical bond between the CN pair through the add\_bond () function. At this time, the molecular structure and topology files will be updated to reflect the newly generated amide bonds and the interconnections between the molecules ( example\_run\_script ). Each time a new bond is generated, the degree of crosslinking (DPC) in the system will increase, and PX- MDsim will record the details of these reactions in the log file .
* **Dynamically adjust reaction conditions** : If no suitable reaction pair is found in a simulation, the software will adjust the reaction conditions, such as increasing the maximum allowed reaction distance (by increasing the dist value) or adjusting the Z-axis range (by zmin and zmax ) ( example\_run\_script ). This allows the software to dynamically adjust reaction conditions according to the specific situation to maximize the possibility of crosslinking.

**5. NPT simulation and Z-axis adjustment**

During the cross-linking reaction, the volume and density of the system may change, so PX- MDsim provides two methods to stabilize the system density: NPT simulation and Z-axis adjustment.

* **NPT simulation** : NPT (constant temperature and pressure) simulation allows the system to be simulated kinetically at constant temperature and pressure. PX- MDsim can perform NPT simulations at specific stages of the crosslinking reaction according to the user's settings ( do\_NPT option) to adjust the volume and density of the system. This step is usually performed after a certain number of crosslinks have been formed. The conditions of the NPT simulation are controlled using the mdp\_NPT file ( example\_run\_script ).
* **Z-axis adjustment** : For some systems, users may want to adjust the Z-axis length to keep the system density stable. PX- MDsim provides the function of adjusting the Z-axis length, which can be activated by setting the do\_adjust\_Z option. The process of Z- axis adjustment involves reducing the Z-axis length of the box, followed by energy minimization and NVT simulation ( example\_run\_script ). This method is particularly suitable for simulations where you want to control the density by reducing the system thickness.

**6. Termination conditions for cross-linking simulation**

PX - MDsim provides a variety of termination conditions, and users can set the end criteria of cross-linking simulation according to specific needs.

* **Maximum number of crosslinks** : The user can set the maximum number of crosslinks in the system via the max\_links parameter, and the simulation will automatically end when this number is reached. This ensures that the simulation does not run indefinitely and that the user can control the final degree of crosslinking ( example\_run\_script ).
* **Maximum degree of crosslinking (DPC)** : The degree of crosslinking is an important indicator to measure the progress of the crosslinking reaction in the system, indicating the density of crosslinking bonds in the system. By setting the max\_dpc parameter, the user can define the end of the simulation when the system reaches a certain degree of crosslinking ( example\_run\_script ). PX- MDsim calculates the DPC of the system after each reaction and terminates the simulation when the maximum value is reached.

**7. Output and analysis of simulation results**

When the crosslink simulation is completed, PX- MDsim will output the final topology and coordinate files of the system and record all the generated crosslinks.

* **Topology and coordinate file output** : Through the output\_contents () function, PX- MDsim will generate the final topology file (.top) and coordinate file (.gro ) , which reflect the system state after all cross-links are generated ( example\_run\_script ). These files can be used for subsequent molecular dynamics simulation or analysis.
* **Handling of unreacted monomers** : PX- MDsim also provides the function of removing unreacted monomers. Through the remove\_residue () function, the software will detect and delete those monomer molecules that did not participate in the reaction ( example\_run\_script ). This helps to generate more accurate cross-linked structures for further physical and chemical analysis.

PX - MDsim automates the entire process from monomer molecule input, force field generation, functional group identification to cross-linking simulation through a series of Python scripts and GROMACS commands, aiming to study the formation of polymer cross-linking structures.