Computer Physics Communications 291 (2023) 108840

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

Computer Programs in Physics

PXLink: A simulation program of polymer crosslinking to study of polyamide membrane ☆,☆☆

Chi Zhang[a], Guangle Bu[a], Md Symon Jahan Sajib[b], Lida Meng[c], Shiying Xu[c], Size Zheng[c,*], Lin Zhang[a,*], Tao Wei[b,*]

[a] Engineering Research Center of Membrane and Water Treatment of MOE, College of Chemical and Biological Engineering, Zhejiang University, Hangzhou, Zhejiang 310027, PR China
[b] Department of Chemical Engineering, Howard University, Washington, D.C., 20059, USA
[c] College of Materials and Chemistry & Chemical Engineering, Chengdu University of Technology, Chengdu, Sichuan 610059, PR China

ARTICLE INFO

Article history:
Received 17 March 2023
Received in revised form 10 June 2023
Accepted 21 June 2023
Available online 7 July 2023

Dataset link: https://github.com/VlachosGroup/pMuTT

Keywords:
Atomistic molecular dynamics
Crosslinking
Polyamide membrane

ABSTRACT

Crosslinked network polymers have numerous important applications in engineering, biomedicine, and the environment. Establishing a crosslinked polymer network is an essential initial step toward studying the structure and function of polymer membranes. In this study, we developed a Python script named PXLink that integrates with GROMACS software to simulate the crosslinked aromatic polyamide (PA) membrane, which is widely used in desalination and water treatment. PXLink can create chemical bonds between the neighboring carboxyl and amine groups and relax the system during the crosslinking, utilizing energy minimization and molecular dynamics simulations implemented in GROMACS. The accuracy of our protocol has been successfully validated through a benchmark study of water diffusivity and the structural properties of a solvated membrane, including pore size distribution, membrane density, and the stacking of aromatic benzene rings (density profile and ordering). Moreover, the comparison between the dry and wet polymer membranes reveals that the swelling of the membrane surface leads to the expansion of subnanopores and the increased ordering of local structures. After the swelling of the membrane, the number of pores with a radius around the main region of 0.2 nm decreases, while more pores in the range of 0.22–0.38 nm are observed. There is also an increase in the orderliness of local structures, as evidenced by the increasing order parameter of aromatic ring pairs with a distance ranging from 0.7 to 0.8 nm. Our developed script will be a valuable tool for designing and discovering crosslinked polymers.

Program summary
Program Title: PXLink
CPC Library link to program files: https://doi.org/10.17632/yjw2ghfzmh.1
Developer's repository link: https://github.com/zchinet30/PXLink
Licensing provisions: GNU General Public License 3
Programming language: Python
Nature of problem: The formation of a crosslinked polymer network is essential for theoretical studies and numerous practical applications. However, simulating the polymer crosslinking process using conventional molecular dynamic simulations presents challenges due to its intricate nature, which involves a complex interplay of physical processes and chemical reactions within a large-scale system.
Solution method: To address this limitation, we developed PXLink, a python-based script, which integrates with the molecular dynamics software of GROMACS for the simulation of polymer crosslinking.

1. Introduction

Network polymers have numerous applications, such as engineering plastics [1–3], hydrogels [4–6], and polymer membranes [7–14]. These polymers are formed through crosslinking, which can occur via covalent bonding [15–18] and noncovalent interactions such as hydrogen bonds [19], processing various architectures

☆ The review of this paper was arranged by Prof. W. Jong.
☆☆ This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).
* Corresponding authors.
E-mail addresses: zhengsize19@cdut.edu.cn (S. Zheng), linzhang@zju.edu.cn (L. Zhang), tao.wei@howard.edu (T. Wei).

https://doi.org/10.1016/j.cpc.2023.108840
0010-4655/© 2023 Elsevier B.V. All rights reserved.

and properties. By controlling the chemical nature and architecture of the crosslinking, the mechanical and chemical properties of the polymer network can be optimized for specific applications [20–22].

Crosslinked aromatic polyamide (PA) membrane has been widely used in desalination and water treatment [14,23–29]. A selective layer of PA thin film composite (TFC) membranes above the bottom substrate is formed by interfacial polymerization of trimesoyl chloride (TMC) and m-Phenylenediamine (MPD) [30]. Despite decades of efforts to enhance and modify the membrane, the basic chemical structure of the PA separation layer has remained generally unchanged since its initial development [23]. Understanding the crosslinked structure of PA membranes and its impact on water permeation and salt rejection or pollutant filtration at the microscopic level is crucial for developing polymer membranes of higher efficacy in the future [12,31–34]. Most pores in the PA membrane have a radius of ∼ 0.2 nm, which is responsible for the water-salt separation functionalities in desalination [9]. This also requires a high simulation resolution, typically at the atomistic level.

Atomistic molecular dynamics (MD) simulation can provide insights into atomic scale details and dynamics from sub-nanoseconds to microseconds, complementing experimental development [9,10,35–45]. However, establishing a crosslinked polymer structure for MD simulations and the subsequent analyses remains a challenging issue. Conventional atomistic MD simulations are limited to presenting physical processes and are unable to simulate chemical reactions that involve the breaking and forming of chemical bonds. Although reactive forcefield (ReaxFF) MD simulations [46–50] have recently been developed to overcome this limitation, they may still be inefficient for a large-sized system and also lack accurate force field parameters to simulate the crosslinking and polymers. To construct crosslinked polymer membranes, a simplified simulation approach has been employed, which involves controlling chemical reactions through an empirical cutoff distance between active sites, followed by a heating/annealing protocol. Chemical bonds can be generated by crosslinking bridges between linear polymer chains [28,51,52] or monomers [9,53–55]. Although these models do not incorporate the solvent effect, they can still produce realistic membrane structures similar to commercial membranes under traditional synthesis conditions [11,23]. To enhance computational efficiency, in our previous work [9], we used a hierarchical crosslinking approach, initially crosslinking monomers in a small simulation cell in a vacuum and then duplicating it in 3D space for additional crosslinking and relaxation. This approach can simulate a crosslinked PA membrane with properties that match experimental measurements, such as the local structure properties, pore size distributions, water diffusivity, and water pressure-driven permeation [9,10]. Our previous study predicted perpendicular "T-shaped" packing of aromatic benzene rings in a crosslinked PA membrane [9], which was confirmed by the subsequent X-ray experiments [8]. There were also several recent simulation studies of crosslinked PA membranes about water dynamics [28,55–57], salt rejection [10,55,57], surface fouling [10,58], surface modification [59], and the composite membrane containing carbon nanotubes [60] or covalent organic framework [61].

Several open-source software packages were recently developed to perform simulation of crosslinking, such as LAMMPS (large-scale atomic/molecular massively parallel simulator) [62,63] and Polymatic (based on LAMMPS) [64]. For example, dissipative particle dynamics (DPD) simulations and coarse-grained models in LAMMPS have been widely applied to study crosslinking of different polymer systems [65–68]. Sutton et al. performed Langevin dynamics simulations with coarse-grained models in an implicit water environment to study polyamide crosslinking using their customized version of LAMMPS [69].

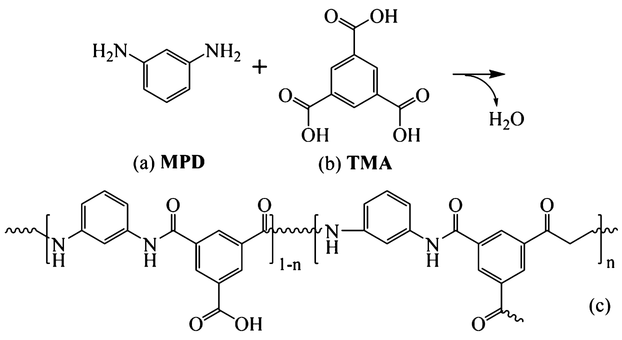

Fig. 1. Molecular structures of (a) MPD, (b) TMA, and (c) the product of condensation with the degree of polymer crosslinking $DPC = n$. Note: The definition of DPC is the ratio between the number of TMC residues that are fully bonded with three MDP residues and the initial total TMC monomers. $n = 0$ indicates a complete linear polymer, while $n = 1$ indicates a fully crosslinked polymer.

In this paper, we developed a Python script called PXLink, which hybrids with the popular MD simulations software of GROMACS [70] to simulate polymer crosslinking. We used the PA membrane as a model system to compare with our previous studies [9] and validate the effectiveness of our script of PXLink for polymer crosslinking. We demonstrated that our script could create chemical bonds between neighboring carboxyl-amine pairs by editing the simulation topology file while utilizing energy minimization and MD simulations in GROMACS to relax the system to achieve the equilibrium state using the heating/annealing protocol. Our benchmark study of a solvated PA membrane validated the accuracy of our approach in modeling polymer membranes. In addition, our case study of the swelling of the PA membrane uncovered the membrane's microscopic structural changes that take place upon the solvation. Details of the simulation procedure and script architecture can be found in the Software description section. The script and two examples are available on GitHub.

2. Software description

2.1. Overview

PXLink is an object-oriented script written in Python 3.9 that automates the simulation of the crosslinking process through GROMACS 4.6.5 [70]. The script requires Python Version 3.9 or newer, as well as the following packages: Numpy [71], Pandas [72], Matplotlib [73], and NetworkX [74] using a Python package manager such as pip or anaconda. The PXLink can be easily customized to work with newer versions of GROMACS. For atomistic MD simulations, we used CHARMM36 general force field [75,76]. Atomistic partial charges of crosslinked polymer were assigned using CGenFF 2.4.0 [76] according to the patterns of different monomers and oligomers (from dimer to tetramer) (Fig. S1 in the Supplementary Information). The script is open-source and available on GitHub at https://github.com/zchinet30/PXLink.

In this work, we utilized PXLink to simulate a crosslinked network of aromatic PA polymers. To simplify the simulations, hydrolyzed TMC (trimesic acid, TMA) monomers are used instead of TMC to interact with MPD (Fig. 1) following our previous work [9]. In the paper, we present simulation results of a large system, which comprises 1822 TMA and 2733 MPD molecules in a cubic box of 9.0 × 9.0 × 11.4 nm^3. In addition, we provided a small demo system of 240 TMA and 360 MPD monomers in the folder of "*Example_crosslinking*" in GitHub. The demo system was used to illustrate the required input and output files in the "Software Description" section, although its results are not discussed in the "Result and Discussion" section.

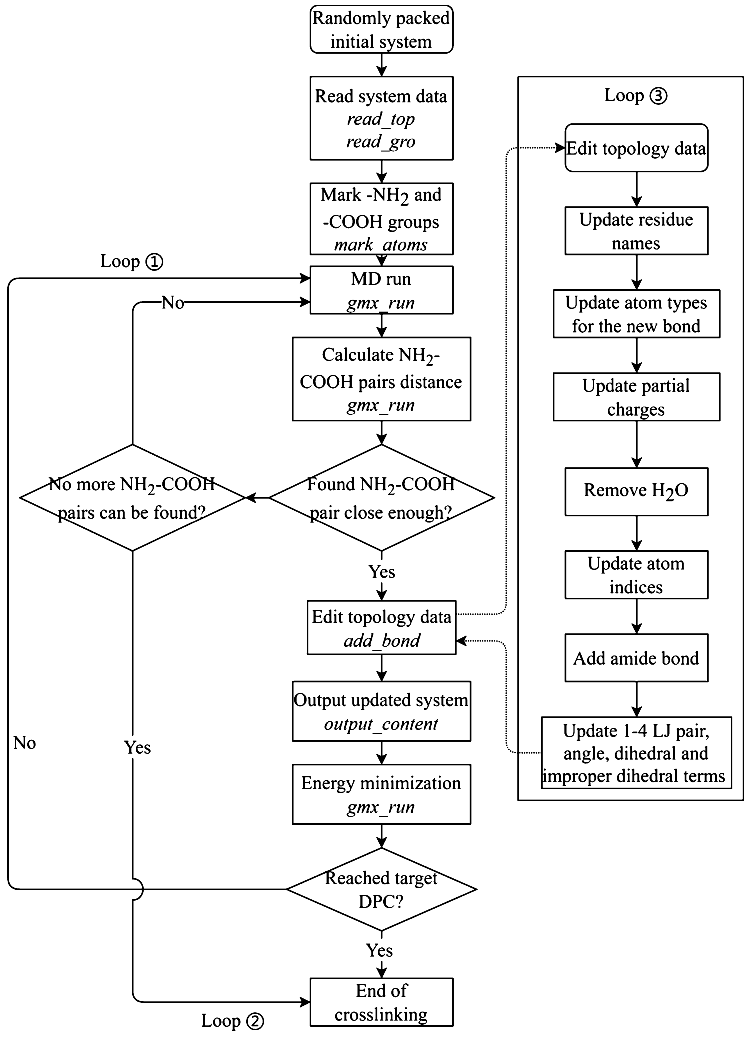

Fig. 2. Flowchart of crosslinking simulation via PXLink proceeded through three loops labeled Loop ①, Loop ②, and Loop ③. The main subroutines (i.e., class methods in Python) that execute each step are italicized.

Table 1
Major subroutines in the PXLink script (GromacsSys class).

Name	Description
read_top	Reads topology data from the "*.top*" file.
read_gro	Reads coordinate data from the "*.gro*" file.
mark_atoms	Identifies free carboxyl and amine groups that could form amide bonds.
gmx_run	Calls GROMACS to perform an MD simulation and then loads new coordinate and topology data from the simulation.
CN_dist	Calculates the distance between amine and carboxyl groups and identifies the closest amine-carboxyl pair that can form an amide bond.
add_bond	Modifies topology data to create an amide bond between a pair of amine and carboxyl groups.
output_content	Outputs topology and coordinate data into "*.top*" and "*.gro*" files.
adjust_Z	Adjust simulation system density by modifying cell size, minimizing energy, and conducting NVT MD protocol.

2.2. Protocol of crosslinking simulation

PXLink utilizes a code structure (known as a class in Python) named *GromacsSys* to record all necessary variables to store the system data and mandatory file paths (listed in Table S1). The flowchart in Fig. 2 illustrates the general protocol for the crosslinking simulation, with each step implemented as a subroutine in the PXLink script. Table 1 lists subroutines (i.e., methods in Python) that perform functions required in the simulation of the crosslinking process.

As shown in Fig. 2, the initial system is constructed by randomly packing TMA and MPD monomers into a box. The topology and coordinate information are saved in two separate files using the "*.top*" and "*.gro*" formats of GROMACS. At the beginning of the simulation, the system's topology and coordinate data are read by the *read_top* and *read_gro* subroutines. Subsequently, the *mark_atoms* subroutine identifies all free carboxyl and amine groups as available for crosslinking. The rest of the program includes three loops: Loop ① refers to the sequence of steps executed when a script successfully locates a pair of free carboxyl carbon and amine nitrogen atoms and creates an amide bond between them; Loop ② is for the case that the script does not find such pairs; Loop ③ updates the topology file.

At the beginning of the crosslinking simulation (Loop ① in Fig. 2), the *gmx_run* subroutine is executed to call GROMACS to perform an MD simulation in an NVT ensemble to relax the system to the equilibrium state. Then, the distances between free carboxyl carbon and amine nitrogen are calculated using the *CN_Dist*

Table 2
Input files in the PXLink execution script "*example_run_script.py*".

File name	File description
init.gro	Coordinate file of the initial system.
init.top	Topology file containing partial charges and force field parameters.
ffbonded.itp	Modified forcefield file, which contains additional parameters of intramolecular interactions.
merged.rtp	Modified file of residue database, which contains residue information of TMA and MPD.
init.ndx	(Optional) Index file that includes a [frozen] group of atoms to fix their positions used in the "constraint wall" method.
minim.mdp	Simulation parameter file for energy minimization after forming amide bond(s).
nvt.mdp	Simulation parameter file for an NVT run. New velocities are generated since the run is performed after topology modification and energy minimization.
nvt_cont.mdp	Simulation parameter file for a continued NVT run in case the program fails to find any appropriate C-N pair to form an amide bond.

Table 3
Main controlling variables in the PXLink execution script: "*example_run_script.py*".

Name	Type	Description
old_gmx	bool	Option of using commands of GROMACS 4.6.x (True) or GROMACS 5.0 and later (False).
run_label	string	The prefix of the names of all files created during the simulation of crosslinking process.
max_links	integer	The maximum number of crosslinking.
max_dpc	float	The maximum DPC.
dist	list	A list of the cutoff distances for amide bond formation.
Set_shift	integer	The maximum number of consecutive search failures for locating a C-N pair using each cutoff distance in the '*dist*' list.
use_zlim	bool	Control variable to determine whether to allow atoms in the middle region ($z_{min} < z < z_{max}$) to form amide bonds.
zmin, zmax	list	Boundaries along z-axis for crosslinking.
do_NPT	bool	Control variable to determine whether to run an MD simulation in the NPT to keep system density stable.
do_adjust_Z	bool	Control variable to determine whether to execute the protocol of adjusting the box size, energy minimization, and MD simulation in the NVT to correct the density of the system.

subroutine. If the distances between any pairs of free carboxyl carbon and amine nitrogen atoms are within the cutoff distance for a condensation reaction (i.e., crosslinking), the script sequentially executes the *add_bond* and *output_content* subroutines to update the topology data and generate new topology and coordinate files. This creates an amide bond between the closest pair of free carboxyl carbon and amine nitrogen atoms. In the newly generated topology files, for all atoms involved in the condensation reactions, their indices, partial charges, and parameters for bonded interactions (including bonding, angle, dihedral, and improper dihedral) and the nonbonded Lennard Jones (LJ) interactions are updated (Loop ③ in Fig. 2). After that, the script will use the *gmx_run* subroutine to perform relaxation using energy minimization. The script also checks for the expected DPC and amide bond count. If either count is reached, it outputs the crosslinked structure, removes the unreacted monomers, and ends the crosslinking process. Otherwise, it repeats Loop ① by starting a new MD simulation.

In the case that the script fails to find any pairs of free carboxyl carbon and amine nitrogen atoms that are within the cutoff distance (see Loop ② in Fig. 2), it also goes back to the beginning of the crosslinking loop and attempts to find such pairs again after another relaxation with an MD simulation in the NVT ensemble. If the script is unable to find any more pairs after multiple consecutive attempts, it concludes that no additional pairs can be found and terminates the simulation, regardless of whether the target DPC has been achieved or not. Once the aforementioned process has been completed, the MPD and TMA monomers can be crosslinked to form an extensive polymer network. The final structure can be obtained by relaxing the crosslinked groups and removing unreacted monomers.

It is worth noting that the PXLink script employs *read_top* and *read_gro* subroutines to retrieve data from the topology and coordinate files, respectively, and save the data into two Pandas data frames. These include atomic indices and names; residue indices and names; and atomic coordinates and charges. We chose Pandas as the primary data structure since it can store all data types in an intuitive and easy-to-search format. Interactive entries in the topology file are stored as lists. Moreover, *read_top* generates a NetworkX graph object to represent the system topology, which is necessary to locate neighboring atoms and connected structures and is also useful for debugging and further developing purposes. Further information on the primary subroutines presented in Fig. 2 is available in the Supplementary Information. It is worth noting that although GROMACS 4.6.5 was used in our study, it is feasible to utilize newer versions of GROMACS by modifying the "*old_gmx*" variable in the execution script.

2.3. Data processing

2.3.1. PXLink input

To initiate the crosslinking process, an execution script is needed to load the initial system and execute each subroutine in the proper order. We will use a demo system as an example to show the required input files. The demo script called "*example_run_script.py*" is provided on GitHub. Table 2 shows the required input files, while Table 3 lists the main run-control variables assigned in the PXLink script. PXLink accepts several standard GROMACS input files: the topology ("*.top*") and coordinates ("*.gro*") files of the initial system, as well as the simulation parameters ("*.mdp*") files required for the MD runs during the process of crosslinking (Table 2).

In our protocol, parameters related to MD simulations, such as MD time steps for relaxation and the simulation conditions (simulation time, temperature, PBC conditions, thermostat methods/parameters, etc.), are defined in the "*.mdp*" files. In contrast, parameters that control crosslinking, such as crosslinking distance and maximum DPC, are assigned directly in the execution script. For the initial simulation of monomers, missing parameters of intramolecular interactions (including bond, angle, dihedral and improper dihedral) were added to "*ffbonded.itp*" file. The residue topology file for TMA and MPD monomers was also included in the CHARMM file "*merged.rtp*" to generate an initial topology file ("*.top*"). This modified force field file "*ffbonded.itp*" should be placed in the same directory as the script and other input files and included in the topology files ("*.top*"). During the crosslinking process, we directly modified the topology data and output the modified topology "*.top*" files (Loop ③ in Fig. 2). All modified atom lists, partial charges, and force field parameters are available in our GitHub repository.

It is also worth noting that as the process of crosslinking further progresses, the polymer network becomes more condensed,

Table 4
Main output files in the PXLink execution script "*example_run_script.py*".

File name	File description
opt.gro	Coordinate file for energy minimization. Output at each loop.
opt.top	Topology file for energy minimization. Output at each loop.
nvt.gro	Coordinate file for the MD run in the NVT. Output at each loop.
nvt.top	Topology file for the MD run in the NVT. Output at each loop.
addbond.gro	Updated coordinate file after forming an amide bond. Output each time after a bond is formed.
topol.top	Updated topology file after forming an amide bond. Output each time after a bond is formed.
crosslink.gro	Final coordinate file. Output at the end of crosslinking.
crosslink.top	Final topology file. Output at the end of crosslinking.
crosslink_removed.gro	Final coordinate file with unreacted monomers removed. Output at the end of crosslinking.
crosslink_removed.top	Final topology file with unreacted monomers removed. Output at the end of crosslinking.
run_log.log	Detailed log file that provides information about activities of each subroutine.
scriptlog.log	Log file that provides a summary of the execution script's activities.
bonds.log	Log file that records the distance between each carboxyl carbon and amine nitrogen when an amide bond is created between them.

which makes it more difficult to locate the crosslinking sites within a given cutoff distance. To address this issue, we introduce a series of cutoff distances defined by the list of *dist*, with the corresponding maximum number of searches specified by the parameter *Set_shift* (Table 2). If a search using a particular cutoff distance fails, a new search is performed using a larger cutoff distance until an appropriate crosslinking site is identified.

2.3.2. Output files

PXLink outputs a series of GROMACS files after crosslinking with a user-defined prefix, followed by the loop numbers and a postfix describing which step they are output in. For example, "*Testrun_loop_1_opt.top*" is the topology file output in the energy minimization step in the crosslinking loop (Loop ① in Fig. 2). After completing the crosslinking process, the final topology and coordinate files are output. The script will also remove the unreacted monomers and output the resulting topology and coordinate files. Additionally, three log files are generated by PXLink to log the script's activities and amide bonds created. Details of the output files are listed in Table 4.

2.4. Simulations of polymer membrane surface

To mimic a large polymer membrane surface, the PBC is applied on two dimensions on the polymer membrane surface (i.e., X- and Y-axes in our simulations) during the simulation of crosslinking. Along the Z-axis (i.e., the surface normal direction), monomers are frozen near the top and bottom boundaries of the simulation cell without involving crosslinking. This generates two "frozen walls" on the top and bottom of the simulation cell in the Z-axis, effectively confining monomers within the designated region without interactions with their periodic images throughout the crosslinking process, preventing the polymer membrane from crosslinking with its periodic image and forming an infinitely-sized polymer chunk devoid of surfaces (Fig. 3).

We directly delete water molecules generated during the condensation reaction (see Fig. 1) without modeling them in our simulation. This can cause a decrease in the overall density of the system, which could lead to the deformation of the membrane for a large system. To ensure a stable density of the polymer membrane, the PXLink provides two options. The first option is to slightly reduce the simulation box's length along the Z-axis once a certain number of amide bonds have formed. It can increase the system density to the level before deleting water molecules. The system is then equilibrated through an energy minimization process followed by an MD run in the NVT ensemble. This density adjustment can be performed using the *adjust_Z* subroutine (see Table 1). The second option is to run MD simulations in the NPT ensemble after a specific number of amide bonds have formed.

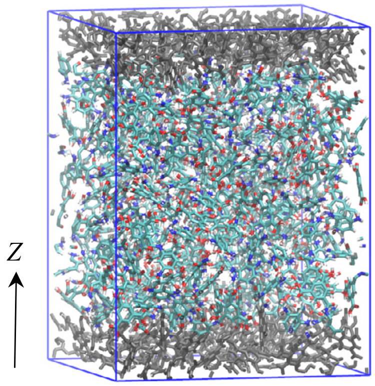

Fig. 3. Snapshot of the system of crosslinking. Two constrained regions were assigned at the bottom and top of the simulation box. Each region has a thickness of ∼ 0.6 nm. Monomers inside the constrained regions (grey color) were frozen without involving crosslinking after the initial relaxation.

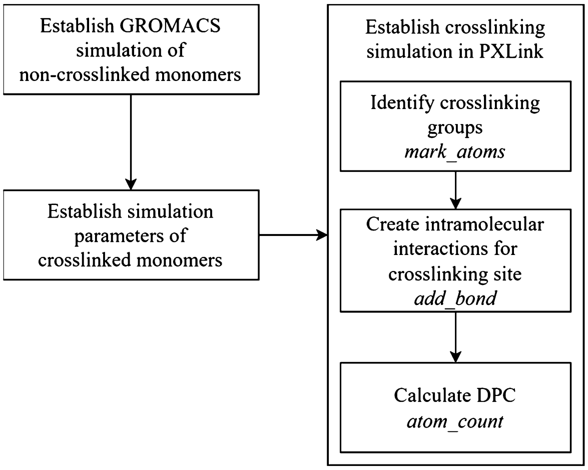
Establish GROMACS simulation of non-crosslinked monomers
Establish simulation parameters of crosslinked monomers
Establish crosslinking simulation in PXLink
Identify crosslinking groups
mark_atoms
Create intramolecular interactions for crosslinking site
add_bond
Calculate DPC
atom_count

Fig. 4. Workflow of applying PXLink on other polymers or force fields.

2.5. Pipeline of crosslinking simulation with different polymers and force fields

PXLink has been specifically developed to simulate the crosslinking process of aromatic polyamide membranes using the CHARMM36 forcefield parameters. However, it is feasible to simulate crosslinking in other polymers using a customized version of PXLink by following the workflow outlined below and making necessary modifications to the PXLink source code. It is also possible to apply different forcefield parameters. The following steps, as outlined in Fig. 4, must be completed prior to initiating the crosslinking process.

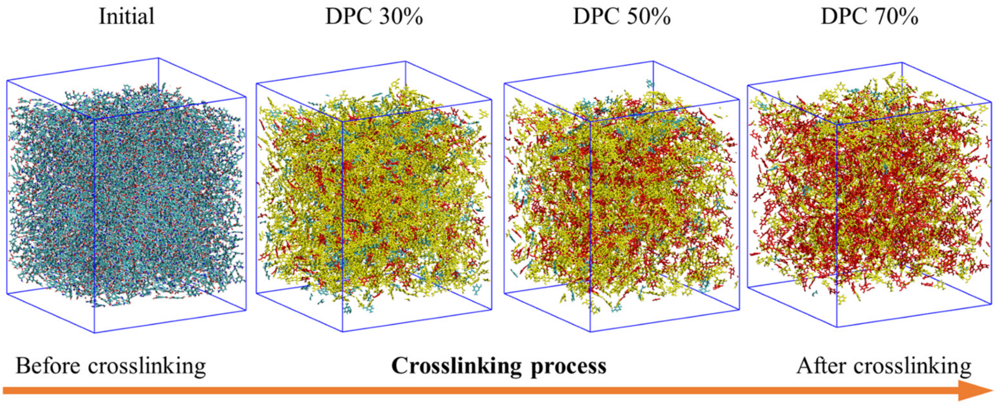

Fig. 5. Snapshots of the crosslinking process: unreacted monomers (cyan), linear groups (uncrosslinked, yellow), and crosslinked TMA-MPD groups (red). Note: according to the definition of DPC, the linear groups belong to the "uncrosslinked" part of the polymer. For clarification, the frozen walls on the top and bottom of the simulation box are not shown. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

The first step is to establish GROMACS simulation setup for the uncross-linked monomers or short polymer chains. This setup includes the force field parameters, topology data, and partial charges of the uncross-linked molecules. Additionally, an initial system of randomly packed monomers or chains should be generated. Since these aspects are associated with the regular GROMACS simulations, we will skip the detailed discussion here.

The second step is to define the simulation parameters of crosslinked monomers or chains. This includes determining the variations of partial charges after the crosslinking reaction and adjusting the parameters for intra- and inter-molecular interactions around the crosslinking sites. In both step 1 and step 2, quantum simulations can be used to estimate the missing simulation parameters.

In the third step, it is necessary to modify the source code of PXLink to simulate a different polymer system. The subroutines "*mark_atoms*", "*add_bond*" and "*atom_count*" are responsible for handling the chemistry structures and atomic properties of the crosslinking bonds and the function groups participating in these bonds. These subroutines also use force field-specific atom types to identify atoms. Therefore, modifications to these subroutines are necessary to perform crosslinking simulations on polymers other than polyamide and using force fields other than CHARMM. More detailed instructions on making these modifications are elucidated in the Supplementary Information.

3. Results and discussion

3.1. Crosslinking and solvation of PA membrane

Fig. 5 shows the crosslinking process of a relatively large PA membrane. The system comprised 1822 TMA and 2733 MPD molecules packed randomly in a cubic box measuring 9.0 × 9.0 × 11.4 nm^3, with an initial density of 1.22 g/cm^3 and a ratio of free carboxyl and amine groups at 1:1. Prior to crosslinking, the system was equilibrated using the NVT ensemble at 560 K for 20 ns. To construct a membrane surface, crosslinking bonds were not formed between different periodic images on the Z-direction. To achieve this, the positions of atoms in the ranges of $z < 1.2$ nm or $z > 10.2$ nm were fixed to exclude them from polymerization, thereby creating a wall layer that prevented movement and bond formation. In each crosslinking (Loop ① in Fig. 2) an MD run was performed for 0.1 ns at 560 K in the NVT. The temperature was elevated above the typical glass-transition temperature of PA polymer to enhance computational efficiency. The script then searched for the closest pairs of amine nitrogen and carboxyl carbon atoms with a cutoff distance of 3–4 Å and created amide bonds between the closest pair, followed by a short energy minimization.

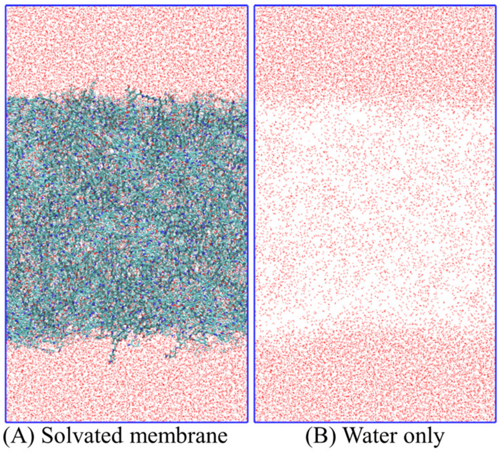

Fig. 6. (A) A snapshot of the solvated membrane after a production run of 500 ns and (B) Water distribution inside the polymer membrane. Red dots indicate water molecules.

In the latter half of the simulation, the length in the Z-axis of the periodic box was adjusted every time 10 bonds were formed, reducing the box size to maintain system density stability. The process was repeated until the DPC reached 70% (Fig. 5). After polymerization, the frozen monomers were removed. To further relax the polymer structure, the system was heated to 700 K for 100 ns, and then cooled down to room temperature. Any remaining unreacted monomers were removed afterward.

Due to the slow swelling process of crosslinked PA membranes, which occurs on a timescale beyond what can be captured by atomistic MD simulations (typically microseconds), we adopted the following protocol: the system's periodic box is initially expanded to 9 × 9 × 14 nm^3, and water molecules are randomly packed in the box, including inside the membrane, using Packmol [77]. The solvated membrane system is then equilibrated with the NVT ensemble at 360 K for 50 ns. To ensure that the membrane can be sufficiently swollen in the MD time scale, this solvating process is repeated three times. The solvated system is then equilibrated with the NVT ensemble at 298.15 K for 50 ns, followed by an NPT equilibration at 298.15 K and 1 bar for 100 ns. Finally, an NVT production run is conducted at 298.15 K for 500 ns.

3.2. PA membrane in vacuum and solvation

Fig. 6 displays a snapshot of a solvated PA membrane and the distribution of water molecules in the entire system. The simulated crosslinked PA membrane is located in the middle of the

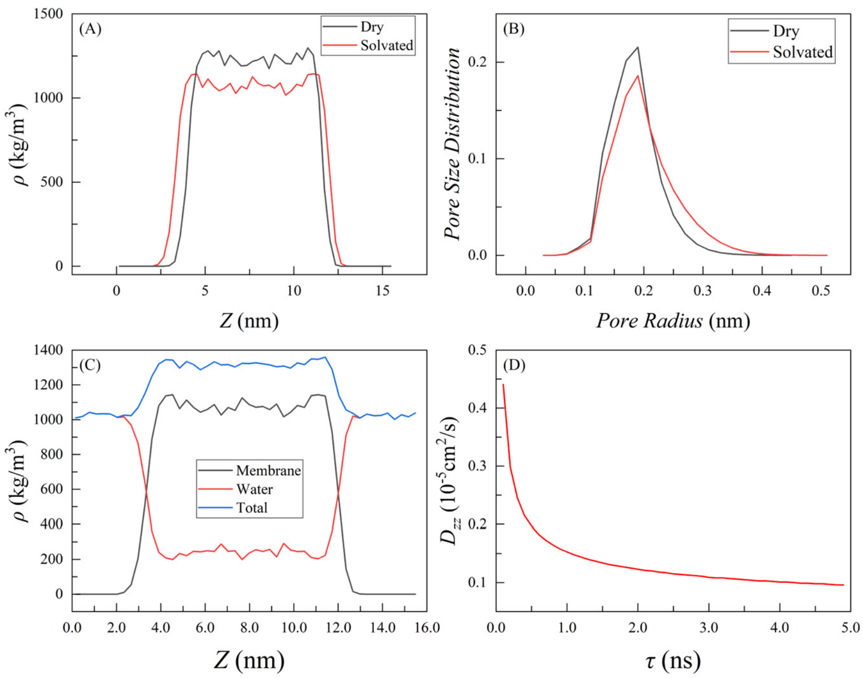

Fig. 7. (A) Density comparison of the dry and solvated PA membranes; (B) Pore size distributions of the dry and solvated membranes; (C) Density profiles of the PA membrane, water, and the entire system for a solvated membrane; (D) Z-component of diffusivity coefficient D_{zz} as a function of autocorrelation time τ.

simulation box, with two water boxes on either side representing the bulk water environment. The lack of notable large-sized water clusters inside the membrane area indicates a general homogeneous crosslinked membrane structure.

In our simulations, the dry membrane has a bulk density of ~ 1233.1 kg/m^3 (Fig. 7A). Upon solvation, the membrane bulk density decreased to ~ 1083.6 kg/m^3, while the membrane thickness expanded by 0.5 nm. The densities of both dry and solvated membranes are close to the previous studies reported in the literature [9,78]. The solvated membrane bulk contains 18.1% water by weight (Fig. 7C), which is within the range of 10–23% reported in previous experimental studies [79,80].

We calculated the pore size of the membrane using a custom C++ script, following the approach published in our previous paper [9]. Fig. 7B compares the pore size distributions of the dry and solvated membranes. Although the majority of pores have radii between 0.16 and 0.20 nm in both the dry and solvated membranes, it indicates that the membrane has expanded, and pores have become larger after solvation. In the dry membrane, the pore radius ranges from 0.06 nm to 0.44 nm, whereas in the solvated membrane, the range of pore radius increases to 0.06–0.50 nm, which is consistent with our previous work [9]. For both dry and solvated membranes, most pores have the radius around 0.2 nm (Fig. 7B). However, after the solvation, the number of pores with a radius around the main region of 0.2 nm decreases, while more pores around 0.22–0.38 nm are detected.

For the solvated membrane system, we also computed the water diffusivity within the membrane bulk. We calculated the diffusivity component perpendicular to the membrane surface (i.e., along the Z-axis) of water molecules inside the polymer thin film with the following autocorrelation functions, by solving the diffusion equation subject to the boundary conditions [9,37,81]:

$$D_{zz} = \frac{\ln(L \cdot \langle \Psi_n(z(t))\psi_n(z(0))\rangle)}{-(\frac{n\pi}{L})^2 t}, \tag{1}$$

$$\Psi_n(z) = \sqrt{2/L}\,\sin\left(\frac{n\pi(z-a)}{L}\right). \tag{2}$$

where the diffusivity of water within boundary $a < z < b$ is calculated, and L is the thickness of this region. To compute water diffusivity in the membrane bulk (6.2 nm $\le z \le$ 9.4 nm), we used the last 100 ns trajectory in the 500 ns production run. As shown in Fig. 8D, D_{zz} in the membrane bulk varied with autocorrelation time τ ranging from 0.43×10^{-5} cm^2/s to 0.10×10^{-5} cm^2/s (Fig. 8D). This result is consistent with our former work [9].

Since the PA membrane contains an abundance of aromatic rings, the local structures of the membranes are largely influenced by the interaction of aromatic rings. To characterize local structures, we calculated the profiles of order parameter $S(r)$ and the radial distribution function $g(r)$ of neighboring benzene rings,

$$S(r) = \left\langle \frac{3\cos^2(\theta(r)) - 1}{2} \right\rangle, \tag{3}$$

$$g(r) = \frac{\rho(r)}{\rho_{bulk}}. \tag{4}$$

where r is the radial distance between benzene ring centers; $\theta(r)$ is the angle between benzene rings' normal vectors at distance r; $\rho(r)$ is the density of benzene rings at r, and ρ_{bulk} is the density of benzene rings in the membrane bulk area ($5.0 < Z < 10.0$ nm). A value of $S = 1$ corresponds to a completely ordered packing structure, whereas $S = 0$ represents a random structure [9,37]. The comparison of the $S(r)$ between the dry and solvated membranes shows that after solvation, the orientation of the benzene rings in the membrane became more ordered (Fig. 9). The difference is particularly pronounced in the $r \sim 0.7$–0.8 nm region. While these structures are not dominant in the membranes, as evidenced by the absence of a $g(r)$ peak in this region (Fig. 9), they become significantly more orderly aligned after solvation. This increase in orderliness can be attributed to the polymer-water interactions. Swelling of the polymer matrix and expansion of sub-nanopores can increase the distance between the polymer chains, which reduces the steric hindrance among the aromatic rings. This decrease in hindrance can allow the benzene rings to pack in a more ordered arrangement (see Fig. 9A). Additionally, the linked aromatic benzene rings may become more elongated, resulting in

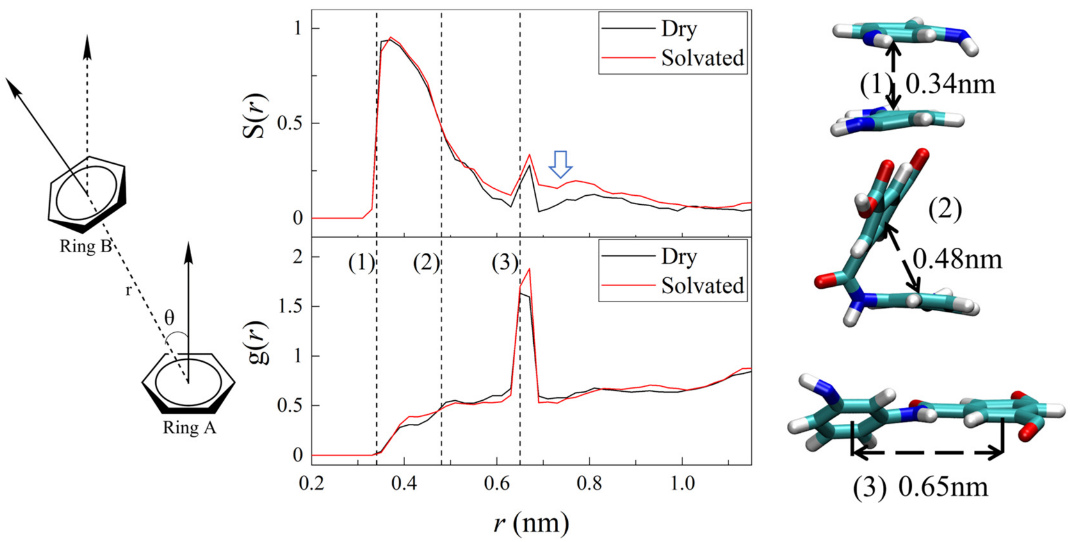

Fig. 8. The profiles of density $g(r)$ and order parameter $S(r)$ between neighboring aromatic benzene rings, where r is the distance between two benzene rings, and θ is the angle between the normal vectors between two benzene rings. Snapshots (1), (2), and (3) display the structures of parallel-stacked rings, T-shaped rings, and covalently bonded rings, respectively. The blue arrow indicates the region where the $S(r)$ increases significantly after the solvation of the membrane.

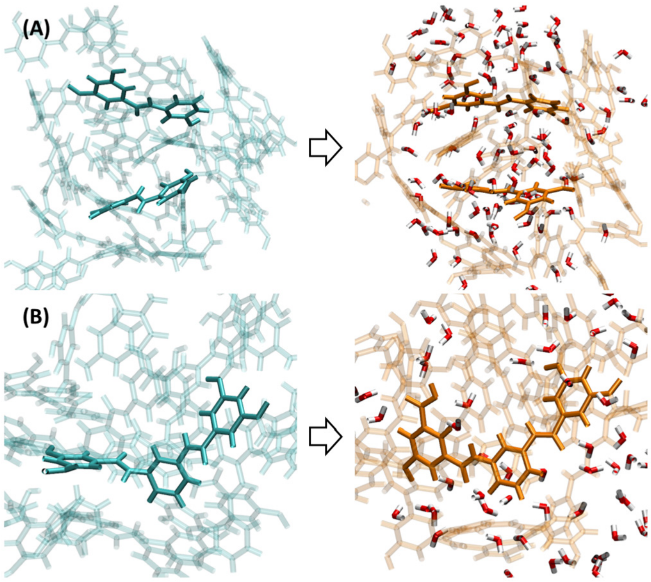

Fig. 9. Comparison of the local structure of the crosslinked membrane in the vacuum (left) and after membrane swelling (right): (A) benzene rings of two PA chains become more parallel after swelling; (B) a PA chain is more stretched, forming a more linear conformation. The structure of a dry membrane is colored cyan, while the structure of a swollen membrane is colored orange.

a less twisted linear structure comprising multiple benzene rings (Fig. 9B), although their occurrence or percentage is low at a high DPC of 70%.

4. Conclusions and future work

In this study, we developed a Python script that can be integrated with the GROMACS software to simulate the polymer crosslinking process, which has wide-ranging applications. Our script creates chemical bonds between reaction-active sites to generate a three-dimensional network while relaxing the system using energy minimization and MD simulations in GROMACS. With necessary modifications, the current version of PXLink can be applicable to different polymer systems and compatible with other force fields or other versions of GROMACS.

We applied our script to simulate a crosslinked aromatic PA membrane using initially randomly packed monomers and a heating/annealing process. Our benchmark results demonstrate that our simulated polymer membrane has similar microscopic structural properties, such as pore size distribution and benzene ring packing, to experimental measurements. After swelling of the polymer membrane in solvation, the packing of the benzene rings of the solvated membrane becomes more ordered compared to that of a dry membrane.

Notably, the functionality of our script can be expanded in the future to incorporate solvent effects and realistic Arrhenius-based reaction kinetic models. This would enable a more comprehensive analysis of the polymer crosslinking process and its practical applications.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

PXLink is a free software with the MIT license. Instructions of installation, the source code, the up-to-date document, and an example can be found on the GitHub page (https://github.com/VlachosGroup/pMuTT).

Acknowledgements

T. Wei acknowledges the grant support from National Science Foundation (NSF 2132524). L Zhang acknowledges funding support from the National Natural Science Foundation of China (22138010 and 52261145697) and the Natural Science Foundation of Zhejiang Province (LZ20B060001). S. Zheng acknowledges the funding support from the National Natural Science Foundation of China (22203011) and the Sichuan Science and Technology Program (2022NSFSC1274).

Appendix A. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cpc.2023.108840.

References

[1] D. Aili, J. Yang, K. Jankova, D. Henkensmeier, Q. Li, J. Mater. Chem. A 8 (2020) 12854–12886, https://doi.org/10.1039/d0ta01788d.

[2] P. Shieh, W. Zhang, K.E. Husted, S.L. Kristufek, B. Xiong, D.J. Lundberg, J. Lem, D. Veysset, Y. Sun, K.A. Nelson, Nature 583 (2020) 542–547, https://doi.org/10.1038/s41586-020-2495-2.

[3] G.M. Scheutz, J.J. Lessard, M.B. Sims, B.S. Sumerlin, J. Am. Chem. Soc. 141 (2019) 16181–16196, https://doi.org/10.1021/jacs.9b07922.

[4] X. Li, C. Tang, D. Liu, Z. Yuan, H.C. Hung, S. Luozhong, W. Gu, K. Wu, S. Jiang, Adv. Mater. 33 (2021) 2102479, https://doi.org/10.1002/adma.202102479.

[5] Y. Gao, K. Peng, S. Mitragotri, Adv. Mater. 33 (2021) 2006362, https://doi.org/10.1002/adma.202006362.

[6] T. Bai, J. Li, A. Sinclair, S. Imren, F. Merriam, F. Sun, M.B. O'Kelly, C. Nourigat, P. Jain, J.J. Delrow, Nat. Med. 25 (2019) 1566–1575, https://doi.org/10.1038/s41591-019-0601-5.

[7] N. Verma, L. Chen, Q. Fu, S. Wu, B.S. Hsiao, Membranes 12 (2022) 1081, https://doi.org/10.3390/membranes12111081.

[8] Q. Fu, N. Verma, H. Ma, F.J. Medellin-Rodriguez, R. Li, M. Fukuto, C.M. Stafford, B.S. Hsiao, B.M. Ocko, ACS Macro Lett. 8 (2019) 352–356, https://doi.org/10.1021/acsmacrolett.9b00077.

[9] T. Wei, L. Zhang, H. Zhao, H. Ma, M.S.J. Sajib, H. Jiang, S. Murad, J. Phys. Chem. B 120 (2016) 10311–10318, https://doi.org/10.1021/acs.jpcb.6b06560.

[10] M.S. Jahan Sajib, Y. Wei, A. Mishra, L. Zhang, K.-I. Nomura, R.K. Kalia, P. Vashishta, A. Nakano, S. Murad, T. Wei, Langmuir 36 (2020) 7658–7668, https://doi.org/10.1021/acs.langmuir.0c01308.

[11] T. Wei, C. Ren, in: Polymer Science and Innovative Applications, Elsevier, 2020, pp. 207–228.

[12] Y.L. Xue, J. Huang, C.H. Lau, B. Cao, P. Li, Nat. Commun. 11 (2020) 1461, https://doi.org/10.1038/s41467-020-15038-w.

[13] Z. Tan, S. Chen, X. Peng, L. Zhang, C. Gao, Science 360 (2018) 518–521, https://doi.org/10.1126/science.aar6308.

[14] S. Han, J. Zhu, A.A. Uliana, D. Li, Y. Zhang, L. Zhang, Y. Wang, T. He, M. Elimelech, Nat. Commun. 13 (2022) 7954, https://doi.org/10.1038/s41467-022-35681-9.

[15] N.J. Van Zee, R. Nicolaÿ, Prog. Polym. Sci. 104 (2020) 101233, https://doi.org/10.1016/j.progpolymsci.2020.101233.

[16] P. Chakma, D. Konkolewicz, Angew. Chem. Int. Ed. 58 (2019) 9682–9695, https://doi.org/10.1002/anie.201813525.

[17] T. Ube, Polym. J. 51 (2019) 983–988, https://doi.org/10.1038/s41428-019-0224-1.

[18] J. Brunet, F. Collas, M. Humbert, L. Perrin, F. Brunel, E. Lacôte, D. Montarnal, J. Raynaud, Angew. Chem. Int. Ed. 58 (2019) 12216–12222, https://doi.org/10.1002/anie.201904559.

[19] P. Song, H. Wang, Adv. Mater. 32 (2020) 1901244, https://doi.org/10.1002/adma.201901244.

[20] A. Bagheri, C.M. Fellows, C. Boyer, Adv. Sci. 8 (2021) 2003701, https://doi.org/10.1002/advs.202003701.

[21] Y. Gao, D. Zhou, J. Lyu, S. A, Q. Xu, B. Newland, K. Matyjaszewski, H. Tai, W. Wang, Nat. Rev. Chem. 4 (2020) 194–212, https://doi.org/10.1038/s41570-020-0170-7.

[22] T. Katashima, Polym. J. 53 (2021) 1073–1082, https://doi.org/10.1038/s41428-021-00505-y.

[23] K.P. Lee, T.C. Arnot, D. Mattia, J. Membr. Sci. 370 (2011) 1–22, https://doi.org/10.1016/j.memsci.2010.12.036.

[24] M. Qasim, M. Badrelzaman, N.N. Darwish, N.A. Darwish, N. Hilal, Desalination 459 (2019) 59–104, https://doi.org/10.1016/j.desal.2019.02.008.

[25] C.F. Carolin, P.S. Kumar, A. Saravanan, G.J. Joshiba, M. Naushad, J. Environ. Chem. Eng. 5 (2017) 2782–2799, https://doi.org/10.1016/j.jece.2017.05.029.

[26] D.M. Warsinger, S. Chakraborty, E.W. Tow, M.H. Plumlee, C. Bellona, S. Loutatidou, L. Karimi, A.M. Mikelonis, A. Achilli, A. Ghassemi, L.P. Padhye, S.A. Snyder, S. Curcio, C.D. Vecitis, H.A. Arafat, J.H. Lienhard, Prog. Polym. Sci. 81 (2018) 209–237, https://doi.org/10.1016/j.progpolymsci.2018.01.004.

[27] S. Kim, K.H. Chu, Y.A.J. Al-Hamadani, C.M. Park, M. Jang, D.-H. Kim, M. Yu, J. Heo, Y. Yoon, Chem. Eng. J. 335 (2018) 896–914, https://doi.org/10.1016/j.cej.2017.11.044.

[28] M. Ding, A. Szymczyk, A. Ghoufi, J. Membr. Sci. 501 (2016) 248–253, https://doi.org/10.1016/j.memsci.2015.12.036.

[29] D. Lu, Z. Yao, L. Jiao, M. Waheed, Z. Sun, L. Zhang, Adv. Membr. 2 (2022) 100032, https://doi.org/10.1016/j.advmem.2022.100032.

[30] J.E. Cadotte, R.J. Petersen, R.E. Larson, E.E. Erickson, Desalination 32 (1980) 25–31, https://doi.org/10.1016/S0011-9164(00)86003-8.

[31] U. Merten, Ind. Eng. Chem. Fundam. 2 (1963) 229–232, https://doi.org/10.1021/i160007a013.

[32] D. Paul, J. Membr. Sci. 241 (2004) 371–386, https://doi.org/10.1016/j.memsci.2004.05.026.

[33] H.F. Ridgway, J. Orbell, S. Gray, J. Membr. Sci. 524 (2017) 436–448, https://doi.org/10.1016/j.memsci.2016.11.061.

[34] X. Ma, D. Lu, J. Lu, Y. Qian, S. Zhang, Z. Yao, L. Liang, Z. Sun, L. Zhang, Desalination 548 (2023) 116293, https://doi.org/10.1016/j.desal.2022.116293.

[35] P. Sarker, G.T. Chen, M.S.J. Sajib, N.W. Jones, T. Wei, Colloids Surf. A, Physicochem. Eng. Asp. 653 (2022) 129943, https://doi.org/10.1016/j.colsurfa.2022.129943.

[36] H. Huang, C. Zhang, R. Crisci, T. Lu, H.-C. Hung, M.S.J. Sajib, P. Sarker, J. Ma, T. Wei, S. Jiang, J. Am. Chem. Soc. 143 (2021) 16786–16795, https://doi.org/10.1021/jacs.1c08280.

[37] T. Wei, M.S.J. Sajib, M. Samieegohar, H. Ma, K. Shing, Langmuir 31 (2015) 13543–13552, https://doi.org/10.1021/acs.langmuir.5b03603.

[38] T. Wei, T. Huang, B. Qiao, M. Zhang, H. Ma, L. Zhang, J. Phys. Chem. B 118 (2014) 13202–13209, https://doi.org/10.1021/jp508549m.

[39] A. Choubey, R.K. Kalia, N. Malmstadt, A. Nakano, P. Vashishta, Biophys. J. 104 (2013) 2429–2436, https://doi.org/10.1016/j.bpj.2013.04.036.

[40] C.L. Rountree, R.K. Kalia, E. Lidorikis, A. Nakano, L. Van Brutzel, P. Vashishta, Annu. Rev. Mater. Res. 32 (2002) 377–400, https://doi.org/10.1146/annurev.matsci.32.111201.142017.

[41] N.P. Van Der Munnik, M.S.J. Sajib, M.A. Moss, T. Wei, M.J. Uline, J. Chem. Theory Comput. 14 (2018) 2696–2704, https://doi.org/10.1021/acs.jctc.7b01057.

[42] T. Zhang, T. Wei, Y. Han, H. Ma, M. Samieegohar, P.-W. Chen, I. Lian, Y.-H. Lo, ACS Cent. Sci. 2 (2016) 834–842, https://doi.org/10.1021/acscentsci.6b00217.

[43] Z. Yuan, P. McMullen, S. Luozhong, P. Sarker, C. Tang, T. Wei, S. Jiang, Chem. Sci. 14 (2023) 2033–2039, https://doi.org/10.1039/D2SC07047B.

[44] J. Yang, L. Tao, J. He, J.R. McCutcheon, Y. Li, Sci. Adv. 8 (2022) eabn9545, https://doi.org/10.1126/sciadv.abn9545.

[45] L. Tao, J. He, T. Arbaugh, J.R. McCutcheon, Y. Li, J. Membr. Sci. 665 (2023) 121131, https://doi.org/10.1016/j.memsci.2022.121131.

[46] M. Samieegohar, F. Sha, A.Z. Clayborne, T. Wei, Langmuir 35 (2019) 5029–5036, https://doi.org/10.1021/acs.langmuir.8b03951.

[47] M.S.J. Sajib, M. Samieegohar, T. Wei, K. Shing, Langmuir 33 (2017) 11102–11108, https://doi.org/10.1021/acs.langmuir.7b03102.

[48] K. Chenoweth, S. Cheung, A.C. Van Duin, W.A. Goddard, E.M. Kober, J. Am. Chem. Soc. 127 (2005) 7192–7202, https://doi.org/10.1021/ja050980t.

[49] X. Lu, X. Wang, Q. Li, X. Huang, S. Han, G. Wang, Polym. Degrad. Stab. 114 (2015) 72–80, https://doi.org/10.1016/j.polymdegradstab.2015.02.004.

[50] A. Shekhar, K.-i. Nomura, R.K. Kalia, A. Nakano, P. Vashishta, Phys. Rev. Lett. 111 (2013) 184503, https://doi.org/10.1103/PhysRevLett.111.184503.

[51] M. Ding, A. Szymczyk, F. Goujon, A. Soldera, A. Ghoufi, J. Membr. Sci. 458 (2014) 236–244, https://doi.org/10.1016/j.memsci.2014.01.054.

[52] Y. Xiang, Y. Liu, B. Mi, Y. Leng, Langmuir 30 (2014) 9098–9106, https://doi.org/10.1021/la501811d.

[53] E. Harder, D.E. Walters, Y.D. Bodnar, R.S. Faibish, B. Roux, J. Phys. Chem. B 113 (2009) 10177–10182, https://doi.org/10.1021/jp902715f.

[54] K. Li, S. Li, W. Huang, C. Yu, Y. Zhou, J. Comput. Chem. 40 (2019) 2432–2438, https://doi.org/10.1002/jcc.26015.

[55] M. Shen, S. Keten, R.M. Lueptow, J. Membr. Sci. 506 (2016) 95–108, https://doi.org/10.1016/j.memsci.2016.01.051.

[56] F. Cui, W. Chen, X. Kong, L. Liu, C. Shi, Y. Li, J. Phys. Chem. B 123 (2019) 3086–3095, https://doi.org/10.1021/acs.jpcb.9b01491.

[57] Y. Luo, E. Harder, R.S. Faibish, B. Roux, J. Membr. Sci. 384 (2011) 1–9, https://doi.org/10.1016/j.memsci.2011.08.057.

[58] Y.-L. Liu, K. Xiao, A.-Q. Zhang, X.-M. Wang, H.-W. Yang, X. Huang, Y.F. Xie, J. Membr. Sci. 577 (2019) 285–293, https://doi.org/10.1016/j.memsci.2019.02.017.

[59] Q.-A. Gu, L. Liu, Y. Wang, C. Yu, Phys. Chem. Chem. Phys. 23 (2021) 6623–6631, https://doi.org/10.1039/d0cp06383e.

[60] Q.-A. Gu, K. Li, S. Li, R. Cui, L. Liu, C. Yu, Y. Wang, Y. Zhou, G. Xiao, Phys. Chem. Chem. Phys. 22 (2020) 22324–22331, https://doi.org/10.1039/d0cp03864d.

[61] Y. Song, M. Wei, F. Xu, Y. Wang, Phys. Chem. Chem. Phys. 21 (2019) 26591–26597, https://doi.org/10.1039/c9cp05026d.

[62] J.R. Gissinger, B.D. Jensen, K.E. Wise, Polymer 128 (2017) 211–217, https://doi.org/10.1016/j.polymer.2017.09.038.

[63] A.P. Thompson, H.M. Aktulga, R. Berger, D.S. Bolintineanu, W.M. Brown, P.S. Crozier, P.J. in't Veld, A. Kohlmeyer, S.G. Moore, T.D. Nguyen, Comput. Phys. Commun. 271 (2022) 108171, https://doi.org/10.1016/j.cpc.2021.108171.

[64] L.J. Abbott, K.E. Hart, C.M. Colina, Theor. Chem. Acc. 132 (2013) 1–9, https://doi.org/10.1007/s00214-013-1334-z.

[65] A.A. Gavrilov, P.V. Komarov, P.G. Khalatur, Macromolecules 48 (2015) 206–212, https://doi.org/10.1021/ma502220k.

[66] M. Akhukov, D. Guseva, A. Kniznik, P. Komarov, V. Rudyak, D. Shirabaykin, A. Skomorokhov, S. Trepalin, B. Potapkin, in: Supercomputing: 7th Russian Supercomputing Days, RuSCDays 2021, Moscow, Russia, September 27–28, Springer International Publishing, 2021, pp. 174–185.

[67] D.V. Guseva, P.V. Komarov, A.V. Lyulin, J. Polym. Sci., Part B, Polym. Phys. 54 (2016) 473–485, https://doi.org/10.1002/polb.23928.

[68] D.V. Guseva, V.Y. Rudyak, P.V. Komarov, A.V. Sulimov, B.A. Bulgakov, A.V. Chertovich, J. Polym. Sci., Part B, Polym. Phys. 56 (2018) 362–374, https://doi.org/10.1002/polb.24548.

[69] J. Muscatello, E.A. Müller, A.A. Mostofi, A.P. Sutton, J. Membr. Sci. 527 (2017) 180–190, https://doi.org/10.1016/j.memsci.2016.11.024.

[70] B. Hess, C. Kutzner, D. Van Der Spoel, E. Lindahl, J. Chem. Theory Comput. 4 (2008) 435–447, https://doi.org/10.1021/ct700301q.

[71] C.R. Harris, K.J. Millman, S.J. Van Der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith, R. Kern, M. Picus, S. Hoyer, M.H. Van Kerkwijk, M. Brett, A. Haldane, J.F. Del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T.E. Oliphant, Nature 585 (2020) 357–362, https://doi.org/10.1038/s41586-020-2649-2.

[72] W. Mckinney, in: Proceedings of the 9th Python in Science Conference, vol. 445, 2010, pp. 51–56.

[73] J.D. Hunter, Comput. Sci. Eng. 9 (2007) 90–95, https://doi.org/10.1109/MCSE.2007.55.

[74] A. Hagberg, P. Swart, D.S. Chult, in: Proceedings of the 7th Python in Science Conference, SciPy, 2008, pp. 11–15.

[75] K. Vanommeslaeghe, E. Hatcher, C. Acharya, S. Kundu, S. Zhong, J. Shim, E. Darian, O. Guvench, P. Lopes, I. Vorobyov, A.D. Mackerell, J. Comput. Chem. 31 (2009) 671–690, https://doi.org/10.1002/jcc.21367.

[76] W. Yu, X. He, K. Vanommeslaeghe, A.D. Mackerell, J. Comput. Chem. 33 (2012) 2451–2468, https://doi.org/10.1002/jcc.23067.

[77] L. Martínez, R. Andrade, E.G. Birgin, J.M. Martínez, J. Comput. Chem. 30 (2009) 2157–2164, https://doi.org/10.1002/jcc.21224.

[78] V. Kolev, V. Freger, Polymer 55 (2014) 1420–1426, https://doi.org/10.1016/j.polymer.2013.12.045.

[79] M. Kotelyanskii, J. Membr. Sci. 139 (1998) 1–16, https://doi.org/10.1016/S0376-7388(97)00220-2.

[80] X. Zhang, D.G. Cahill, O. Coronell, B.J. Mariñas, J. Membr. Sci. 331 (2009) 143–151, https://doi.org/10.1016/j.memsci.2009.01.027.

[81] P. Liu, E. Harder, B.J. Berne, J. Phys. Chem. B 108 (2004) 6595–6602, https://doi.org/10.1021/jp0375057.