ARMY RESEARCH LABORATORY ARL

Constructing Cross-Linked Polymer Networks Using Monte Carlo Simulated Annealing Technique for Atomistic Molecular Simulations

by Robert M Elder, Timothy W Sirk, and Jan W Andzelm

ARL-TR-7127 October 2014

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer’s or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory
Aberdeen Proving Ground, MD 21005-5066

ARL-TR-7127 **October 2014**

Constructing Cross-Linked Polymer Networks Using Monte Carlo Simulated Annealing Technique for Atomistic Molecular Simulations

Robert M Elder, Timothy W Sirk, and Jan W Andzelm
Weapons and Materials Research Directorate, ARL

REPORT DOCUMENTATION PAGE	***Form Approved*** ***OMB No. 0704-0188***

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)* October 2014	**2. REPORT TYPE** Final	**3. DATES COVERED (From - To)** January 2014–July 2014

4. TITLE AND SUBTITLE Constructing Cross-Linked Polymer Networks Using Monte Carlo Simulated Annealing Technique for Atomistic Molecular Simulations	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Robert M Elder, Timothy W Sirk, and Jan W Andzelm	**5d. PROJECT NUMBER**
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) US Army Research Laboratory ATTN: RDRL-WMM-G Aberdeen Proving Ground, MD 21005-5066	**8. PERFORMING ORGANIZATION REPORT NUMBER** ARL-TR-7127
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSOR/MONITOR'S ACRONYM(S)**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

A method for constructing atomistic models of cross-linked polymer networks using Monte Carlo simulated annealing polymerization is described in detail. The simulated annealing method is appealing because it is computationally efficient and, therefore, allows the construction of larger cross-linked networks at higher conversion than is feasible with other methods. We demonstrate the application of this method to cross-linked networks of poly(dicyclopentadiene) with varying cross-link density. The instructions are not specific to this chemistry and can easily be used to construct atomistic models of cross-linked networks with any type of chemistry. The ability to quickly and easily build computational models of cross-linked polymer networks with varying chemistry will aid in the rational design of improved materials.

15. SUBJECT TERMS

cross-linked polymer network, dicyclopentadiene, mechanical properties, thermal properties

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Robert M Elder
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	UU	42	**19b. TELEPHONE NUMBER** (*Include area code*) 410-306-1926

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

INTENTIONALLY LEFT BLANK

1. Introduction

Cross-linked polymer networks, such as epoxy thermosets and vulcanized rubber, are widely used in military and commercial applications because of their superior mechanical properties relative to linear polymers.[1–5] Numerous experimental studies of various cross-linked materials have been conducted with the overall goal of designing more effective materials.[1, 6–9] Molecular simulations of cross-linked polymers can provide information complementary to these experimental approaches, such as the atomic level details that underlie the macroscopic material properties.[6,10–16] However, molecular simulations require initial structures of cross-linked polymer networks, which are typically not available through experimental means.

Atomically detailed structures of cross-linked polymers are nontrivial to construct *de novo* because of the complex nonlinear topology of networks. To overcome this challenge, several methods for generating network structures have been proposed by other researchers.[12,17–22] Of the proposed methods, the single-step simulated annealing approach of Khare and coworkers[12] is attractive because it is computationally efficient and, thus, allows the construction of larger cross-linked networks than do the other methods.

In this report, we describe in detail the single-step simulated annealing procedure for generating cross-linked polymer network structures for atomistic molecular simulations. The procedure consists of 6 stages: generating initial structures of the components of the network; creating a physical mixture of the components of the network; generating network connectivity with a Monte Carlo simulated annealing technique; creating and relaxing cross-linking bonds; updating the force field of the cross-linked network; and relaxing the network structure using molecular dynamics simulated annealing. These stages are shown schematically in Fig.1.

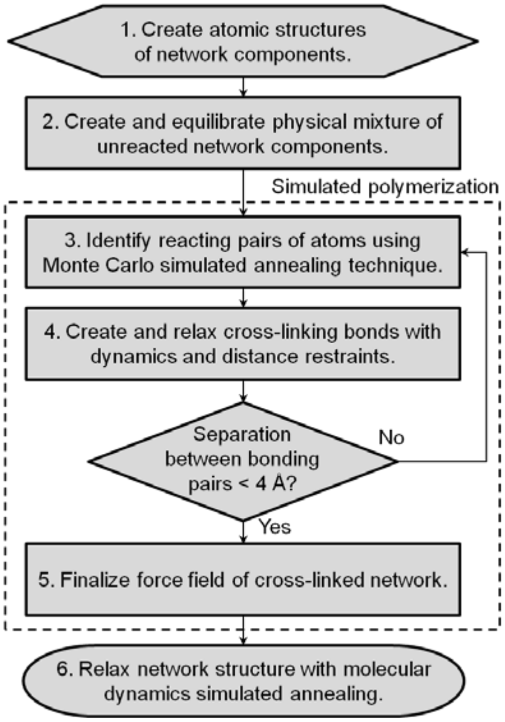
1. Create atomic structures of network components.
2. Create and equilibrate physical mixture of unreacted network components.
Simulated polymerization
3. Identify reacting pairs of atoms using Monte Carlo simulated annealing technique.
4. Create and relax cross-linking bonds with dynamics and distance restraints.
Separation between bonding pairs < 4 A?
No
Yes
5. Finalize force field of cross-linked network.
6. Relax network structure with molecular dynamics simulated annealing.

Fig. 1 Schematic of stages for building cross-linked polymer network. The schematic is based on the diagram in Lin and Khare.

2. Method for Constructing Cross-Linked Polymer Network

2.1 Stage 1. Create Atomic Structures of Network Components

Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS) is a widely used classical molecular dynamics code.[23] LAMMPS, an open source code developed and distributed by Sandia National Laboratories (http://lammps.sandia.gov), is the primary software tool used in the network building process because it efficiently performs computer simulations of large systems of interacting particles and because it includes much of the required functionality in a single convenient software package.

2.1.1 Step 1: Build Atomic Structures

Create atomic structures of the components of the polymer network: the linear segments and the cross-linkers. In this report, we use the term, linear segment, to refer to a component of the polymer network with connectivity of 2. We use the term, cross-linker, to indicate a component with connectivity greater than 2. Use molecule building software, such as CambridgeSoft ChemOffice, VegaZZ, and MarvinSketch, to generate these atomic structures in Tripos mol2 format. Appendix A lists other required software.

For example, to create an epoxy network, create a structure of an epoxy monomer (the linear segment, e.g., diglycidyl ether of bisphenol A) and a cross-linker molecule (e.g., poly[oxypropylene] diamine, 4,4'-methylenebis [cyclohexylamine]). For a cross-linked dicyclopentadiene (DCPD) polymer (Fig. 2a), the linear segment is a predefined oligomer of several DCPD units where the norbornene-like rings have opened and bonded; 2 terminal methyl groups on these oligomers are used to bond with cross-linkers (Fig. 2b). The cross-linker is a DCPD monomer where both the norbornene and the cyclopentene rings have opened; 4 terminal methyl groups on the cross-linker are used to bond with linear segments (Fig. 2c). After these methyl groups are double-bonded in the cross-linking reaction, 2 hydrogen atoms are deleted from each methyl group. Although these DCPD linear segments and cross-linker molecules are not a chemically realistic representation of DCPD cross-linking, they are convenient intermediates for the simulated polymerization procedure. Regardless of the artificial nature of the initial network components, the final cross-linked network is a chemically realistic model.

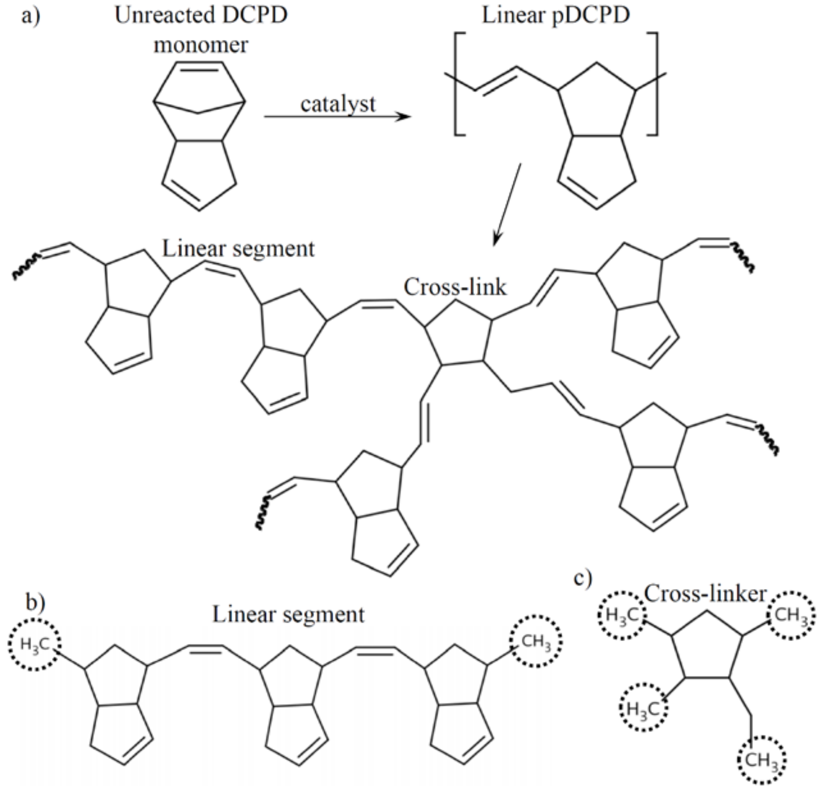

Fig. 2. a) Schematic of DCPD cross-linking reaction. b) DCPD linear segment in which 3 DCPD monomers are bonded via ring-opening of the norbornene-like ring. Atoms that will later form cross-linking bonds are circled; the number of cross-linking bonds (i.e., connectivity) is 2. c) DCPD cross-linker in which both the norbornene-like and cyclopentene-like rings have undergone ring-opening. Atoms that will later form cross-linking bonds are circled; the number of cross-linking bonds (i.e., connectivity) is 4.

2.1.2 Step 2: Assign Initial Atom Types, Charges, Force Field Parameters

Use the Antechamber program in Assisted Model Building with Energy Refinement (AMBER) Tools to assign partial charges (using the Austin Model 1 [AM1]-bond charge corrections [BCC] method[24]) and atom types (using the general AMBER force field[25] [GAFF]) to both of the molecules. An example command-line input with a typical AmberTools installation follows.

```
$AMBERHOME/bin/antechamber -i input.mol2 -fi mol2 -o output.mol2 -fo mol2 -c bcc
```

> Note: Because of differences in `.mol2` file formatting between various molecule building programs (e.g., field width, number format), Antechamber may fail to read `input.mol2`. If this failure occurs, loading the `.mol2` file in Visual Molecular Dynamics (VMD) and saving as a `.mol2` file with VMD may help. The `.mol2` file format generated by VMD is compatible with Antechamber.

Use the Tleap program in AmberTools to generate AMBER format topology and coordinate files for each molecule. An example usage of `Tleap` is given below

```
    $AMBERHOME/bin/tleap -f input.bat
```

where `input.bat` contains

```
    source leaprc.gaff
    b = loadmol2 output.mol2
    saveamberparm b output.top output.crd
    quit
```

Use the LAMMPS tool, `amber2lammps.py,` to convert the AMBER format topology/coordinate files to a LAMMPS data file. (`amber2lammps.py` version Oct 8, 2013, or newer is recommended. `amber2lammps.py` is distributed with LAMMPS in `tools/amber2lmp`/, as of LAMMPS version 1 Feb 2014). `amber2lammps.py` automatically finds all pairs of `.top/.crd` files in the current directory (e.g., `output-1.top`, `output-1.crd`, `output-2.top`, `output-2.crd`) and converts them to LAMMPS data files (e.g., `data.output-1`, `data.output-2`). The products of this step are 2 LAMMPS data files: 1 for the linear segment and 1 for the cross-linker.

Note: Older versions of `amber2lammps.py` (prior to 2011) generated LAMMPS data files that required a small correction to the dihedral coefficients section to make the third term (periodicity) an integer. The recommended version of `amber2lammps.py` does not have this issue.

Note: Older versions of `amber2lammps.py` (prior to 8 Oct 2013) require corrections to the top files created by newer versions of Tleap (AmberTools12 and newer): the sections ATOMIC_NUMBER, SCEE_SCALE_FACTOR, and SCNB_SCALE_FACTOR in the `top` files must be removed for older versions of `amber2lammps.py` to work. The recommended version of `amber2lammps.py` does not have this issue.

Note: You can visualize LAMMPS data files in VMD with this TopoTools command.

```
topo readlammpsdata /path/to/data.file
```

2.1.3 Step 3: Reassign Atom Types and Adjust Charges

Assign unique atom types to all unique atoms (unique by charge, force field, or position in the molecule). Later, this unique signature will allow atoms to be identified unambiguously for the purpose of assigning new force field parameters after forming the cross-linked network. Without unique atom types, it is more difficult to unambiguously identify these atoms. The unique atom types may be assigned using knowledge of which atoms are chemically equivalent (e.g., hydrogen atoms in a methyl group). An alternate approach is to assign all atoms a unique type regardless of their chemical nature, which eliminates all ambiguity in the following steps.

To reassign the atom types, manually edit the data files containing the linear segment and cross-linker. In the `Atoms` section of the data files, change the third column (which is the atom type identification [ID]) of each row to a unique ID number. Change the `# atom types` entry near the top of the data files to indicate the new number of atom types. In the `Masses and Pair Coeffs` sections, create an entry for all atom types, and copy the parameters of the new types from the corresponding old types (e.g., copy the `Pair Coeffs` entry for Type 1 atoms in the old data file to all of the new atom types that were previously Type 1 atoms in the old data file).

Typically, the net charge of the 2 components (linear segment and cross-linker) is not exactly zero because of small rounding errors (e.g., a net charge of –0.002). Change the atomic charges slightly so that both data files have zero total charge (or possibly integer total charge, if the network is electrically charged). One way of changing charges is by evenly distributing the opposite of the erroneous charge over all atoms in the molecule. Another way of changing charges is to round the charge of several atoms by a small amount so that the overall effect of rounding is to make the total charge zero. Either of these methods is acceptable because the change in any individual atomic charge will be very small.

The products of this stage are 2 LAMMPS data files: 1 data file contains the linear segment molecule and 1 data file contains the cross-linker molecule.

2.2 Stage 2. Create and Equilibrate Physical Mixture of Unreacted Network Components

The reaction mixture is a large mixture of linear segments and cross-linkers, which will subsequently be cross-linked into a polymer. Starting from the 2 data files created in the previous step, apply 1 of the 2 methods described below to create the reaction mixture.

2.2.1 Step 1: Create Reaction Mixture

Option 1 for creating reaction mixture: custom combining code.

Use `CombinedReactants4Epoxy.cpp` to generate a LAMMPS data file with a stoichiometric mixture of linear segments and cross-linkers named `data.Combined`. `CombinedReactants4Epoxy` asks the user for the names of 2 input LAMMPS data files (1 containing the linear segment and 1 containing the cross-linker) and the name of the output LAMMPS data file (`data.Combined`). Currently, this program creates a data file that is stoichiometric for epoxy: 2 linear segments and 1 cross-linker.

> Note: To use `CombinedReactants4Epoxy.cpp`, each line in the `Atoms` section of the data file should have 10 entries. If these lines only have 7 entries, simply add 3 zeros at the end of each line.

Next, generate initial coordinates of the reaction mixture using LAMMPS functionality. Read the data file containing stoichiometric ratio of linear segments and cross-linkers (e.g. `read_data data.Combined`). Replicate the small mixture on a lattice to create a larger mixture (e.g., `replicate 3 5 10`). You should choose the lattice dimensions to create a roughly cubic box. The lattice dimensions that will create a cubic box will depend on the size of `data.Combined`. The total system size should also be large enough to generate statistically significant results. A system size of approximately 200,000 atoms produces reasonable results. Appendix B shows generic header for LAMMPS input files.

An example LAMMPS input is provided below.

```
# Read file containing stoichiometric mixture of network components
read_data       data.Combined
# Minimize the structure to eliminate overlaps
min_style       sd
minimize        0 0 100000 100000
min_style       cg
minimize        0 0 100000 100000
# Replicate stoichiometric mixture on a lattice
# Pick lattice size so box is roughly cubic system is large enough
replicate       10 10 14
# Then run dynamics to equilibrate reaction mixture density
```

Option 2 for creating reaction mixture: Moltemplate.

Use the program Moltemplate (distributed with LAMMPS in `tools/moltemplate/src/`, as of LAMMPS version 1 Feb 14) to generate a large reaction mixture.

First, convert the data files for the individual network components to the Moltemplate input format (`.lt`) using the `ltemplify.py` utility. Example:

```
ltemplify.py -name xlinker data.xlinker > xlinker.lt
```

Then, use `moltemplate.sh` to load these data files and replicate them on a lattice in the desired ratio as shown below.

```
moltemplate.sh combine.lt
```

where `combine.lt` contains commands, such as the following, which creates a 2:1 mixture of linear segments and cross-linkers on a 10 × 10 × 10 lattice.

```
import "xlinker.lt" # xlinker.lt in the current directory
import "linear.lt"
# Create a 10x10x10 lattice of linear segments, with 20 Å between molecules
a = new linear [10].move(20, 0, 0)
                [10].move(0, 20, 0)
                [10].move(0, 0, 20)
# Create a second 10x10x10 lattice of linear segments, offset from the first
lattice by 10 Å in the x direction to avoid overlaps with the first lattice
b = new linear.move(10, 0, 0)    [10].move(20, 0, 0)
                                 [10].move(0, 20, 0)
                                 [10].move(0, 0, 20)
# Create a 10x10x10 lattice of cross-linkers, offset from the previous two
lattices by an amount that avoids overlaps
c = new xlinker.move(5, 8, 8)    [10].move(20, 0, 0)
                   [10].move(0, 20, 0)
                   [10].move(0, 0, 20)
# End of script – all molecules instantiated by “new” are written by
moltemplate.sh to a LAMMPS data file
```

The size of the lattice should be chosen to generate a system that is roughly cubic and to yield a system of the desired size. The spacing between molecules should be chosen to avoid atomic overlap. The primary output of `moltemplate.sh` is a LAMMPS data file (e.g., `combine.data` if using the commands above).

Edit the LAMMPS data file created by `Moltemplate` (`combine.data`) to match the format required in following steps. Delete a header line similar to the following `impropers 0`. Edit the box size information (i.e., *xlo*, *xhi*, etc.) in `combine.data` to match the size of the system. The size of the system can be determined with VMD using the following command after reading `combine.data`.

```
measure minmax [atomselect top all]
```

2.2.2 Step 2: Equilibrate Reaction Mixture With Minimization and Dynamics

After using one of the above options for creating the reaction mixture, equilibrate the reaction mixture by running minimization and dynamics in LAMMPS. The potential energy and density of the reaction mixture should reach a stable value. After equilibration, create a LAMMPS data file (`data.Soup`) of the final configuration of the equilibrated reaction mixture.

> Note: A generic header file for all LAMMPS simulations is located at the end of this document. Include commands similar to those in the header file prior to the commands listed in each example LAMMPS input instructions in this document.

Example LAMMPS input is provided below.

```
# Minimize the structure to eliminate overlaps
min_style       sd
minimize        0 0 100000 100000
min_style       cg
minimize        0 0 100000 100000
# Generate initial velocities
velocity        all create 700.0 12345 dist gaussian loop local
# NPT dynamics at high pressure to squish box quickly
fix             1 all npt temp 700.0 700.0 100.0 iso 100.0 100.0 1000.0
run             100000 #fs
# NPT dynamics at 1 atm to equilibrate density
fix             1 all npt temp 700.0 700.0 100.0 iso 1.0 1.0 1000.0
run             1000000 #fs
unfix   1
write_restart   restart.Soup
```

Then, convert `restart.Soup` to `data.Soup` with `restart2data` by entering the following instruction.

```
/path/to/restart2data restart.Soup data.Soup
```

> Note: In future versions of LAMMPS, the `restart2data` tool will be replaced by the native LAMMPS command `write_data`, which is analogous to the existing `write_restart` command.

The product of this stage is a data file (`data.Soup`) containing the equilibrated reaction mixture.

2.3 Stage 3. Identify Reacting Pairs of Atoms Using Monte Carlo Simulated Annealing Technique

The reaction mixture may now be cross-linked using the simulated annealing procedure,[12] which is available as an in-house code called `simann` (or `simann_dcpd`, depending on the type of network).

> Note: The simann program may need to be customized to account for different types of network connectivity. For example, in epoxy networks, each cross-linker atom forms 2 bonds (1 bond each with 2 atoms in a linear segment); whereas in DCPD networks, each cross-linker atom forms only 1 bond with 1 atom in a linear segment. Currently, this connectivity (i.e., the number of bonds per type of atom) is hard-coded.

> Note: Several parameters affecting the Monte Carlo simulated annealing method, such as the initial temperature, rate of temperature decrease, and number of MC moves per temperature, are hard-coded. Although the hard-coded values appear to work well for DCPD and epoxy networks, it is possible they will need to be changed for other applications.

The simulated annealing program (`simann`) takes as input 1) the equilibrated reaction mixture LAMMPS data file (`data.Soup`) and 2) lists of atom types that will form bonds. The output of the simulated annealing program is a LAMMPS data file (`data.React`) with bonds formed between the linear segments and cross-linkers via a Monte Carlo method.[12]

An example input file for simulated annealing of DCPD, which must be named `SimulatedAnnealingConfig` and must be in the current directory, is shown below.

```
InputFilename data.Soup
OutFilename data.React
NumberofcrosslinkerTypes 4
NumberofMonomerTypes 2
AtomTypesofcrosslinker 76 77 78 80
AtomTypesofMonomer 9 30
```

The atom types of the cross-linker and linear segment are the atoms that will form cross-linking bonds. In this example, the atom types of the cross-linker (76, 77, 78, 80) correspond to the carbon atoms in the methyl groups in the DCPD cross-linker, and the atom types of the linear segment (9, 30) correspond to the carbon atoms in the methyl groups in the linear DCPD units (see Fig. 2). The number of cross-linker/linear segment types refers to the number of atoms in each cross-linker/linear segment that will form bonds.

In the primary version of the code, cross-linkers are allowed to form bonds with linear segments, and vice versa. Cross-linkers are not allowed to form bonds with cross-linkers, and linear segments are not allowed to form bonds with linear segments. Each cross-linker or linear segment may only form 1 bond with another particular linear segment or cross-linker, which prevents very short loop structures from forming.

Note: Alternative versions of the code allow 1) bond formation between any 2 cross-linking atoms, regardless of cross-linker or linear segment status; and 2) short loop formation by removing the requirement that a cross-linker or linear segment may only form 1 bond with another particular cross-linker or linear segment. These codes can be used to create networks with different types of topology.

Note: The current versions of the simulated annealing programs, and the instructions in the steps that follow, assume that only 1 new type of bond forms during cross-linking. That is, it is assumed that all cross-linking bonds are identical.

Run the Simann program in a directory containing `data.Soup` and `SimulatedAnnealingConfig` by entering an instruction such as the following.

```
/path/to/simann
```

While Simann is running, the program will output the current state of the simulated annealing procedure, including the current dimensionless temperature and the length of the path of the cross-linking bonds.

The product of this stage is a LAMMPS data file (`data.React`) with new bonds formed between the linear segment atoms and cross-linker atoms according to the Monte Carlo simulated annealing protocol.

2.4 Stage 4. Create and Relax Cross-Linking Bonds with Dynamics and Distance Restraints

The new bonds created in the previous stage are typically much longer (approximately 10 Å) than the equilibrium value of a chemical bond (approximately 1.5 Å). In this step, these bonds are gradually relaxed to their equilibrium state by using LAMMPS functionality to gradually change the parameters that define the bonds.

Consider the example LAMMPS input below, where the most important line is the following.

```
bond_coeff  8 ${kb} ${db}
```

This input instructs LAMMPS to use the variables $kb and $db as the bond parameters of bond Type 8, which is the new cross-linking bond. The ID of the new cross-linking bond is the largest bond type ID in `data.React` (i.e., it is at the bottom of the `Bond Coeffs` section of `data.React`). The equilibrium bond length (db) starts at a large value (10 Å) and decreases in stages to the actual equilibrium bond distance (1.324 Å); the bond spring constant (kb) starts at a very small value (1 kcal/mol/Å^2) and increases in stages to the actual spring constant (589.7 kcal/mol/Å^2).

Example LAMMPS input:

```
# Gradually bring bond coefficients to final value
# Here, the final value is a C=C bond, so the final values are
# kb=589.7 and db=1.324
# Get these new bond parameters from the general Amber force field
# Must run this like: lmp -in input.in
# Use the following parameters for kb and db
variable    kb index 1 3 15 589.7
variable    db index 10 8 3 1.324
 # Loop over bond coefficient variables
label   nextbondcoeff
# Update bond coefficients of the new bonds (here, bondID = 8)
bond_coeff  8 ${kb} ${db}
# Minimize energy to begin equilibration to new bond parameters
min_style   sd
minimize        0 0 100000 100000
min_style   cg
minimize        0 0 100000 100000
# Run dynamics
fix             1 all npt temp 773.0 773.0 100.0  iso 1.0 1.0 1000.0
run             100000 #fs
unfix   1
next    db # update dbnext    kb # update kb
jump    SELF nextbondcoeff       # loop
# Write restart file
write_restart   restart.BondsRelaxed
```

Note: At the beginning of this simulation, LAMMPS may terminate with an error message similar to, “Bond atom missing in image check”, which means that 2 bonded atoms are very far apart. This error can often be overcome by increasing the ghost atom cutoff by adding a command to the LAMMPS input script similar to the following.

```
communicate single cutoff 10.0
```

Increase the cutoff value (10 Å, in this example) until the error is eliminated. After the bonds are closer to their equilibrium length, this command is unnecessary and should be removed from the LAMMPS input script to avoid negatively impacting performance.

Note: To determine the equilibrium bond length and bond spring constant (the final values of db and kb), either examine the GAFF (located at `$AMBERHOME/dat/leap/parm/gaff.dat` in a standard AmberTools installation), or see Stage 5, Step 2.

Note: To use the `jump SELF` functionality, this script must be passed to the LAMMPS executable by using the `-in flag`.

After relaxing the bond lengths, check that the distance between all pairs of reacting atoms is less than 4 Å. If this condition is met, proceed to the next stage. If this condition is not met, return to the previous stage and repeat the simulated annealing calculation using the final structure of the simulations in this stage. The bond lengths can be monitored in LAMMPS with instructions similar to the following.

```
# Create a group of the cross-linker and linear segment atoms involved in new
bonds group newbonds type 76 77 78 80 9 30
# Create a compute to calculate the lengths of bonds between atoms in the group
compute 1 newbonds bond/local dist
# Output the bond lengths every 10000 steps to a file called bond-length.dump
dump 1 newbonds local 10000 bond-length.dump c_1
```

After constructing a cross-linked system in which the distance between all pairs of reacting atoms is less than 4 Å, convert the final restart file to a LAMMPS data file (`data.BondsRelaxed`) using `restart2data`. The product of this stage is this LAMMPS data file, `data.BondsRelaxed`, with the cross-linking bonds close to their equilibrium lengths.

2.5 Stage 5. Finalize Force Field of Cross-Linked Network

The cross-linked system currently contains atoms that must be removed as a result of the cross-linking reaction, and the force field parameters of the cross-linked network must be updated.

2.5.1 Step 1: Delete Unnecessary Atoms

Delete atoms that should be eliminated as a result of bond formation (e.g., excess hydrogen atoms bonded to atoms involved in cross-linking) and the bonds, angles, and dihedrals of those atoms with other atoms using LAMMPS functionality. Atoms will be deleted by identifying LAMMPS atom types to remove. Visualizing the network structure in VMD will aid in identifying these atom types. Choose these atom types based on knowledge of the chemical structure of the network after cross-linking.

> Note: If a choice between chemically identical atoms must be made (e.g., 2 out of 3 hydrogen atoms in a methyl group, as in DCPD), it is unimportant which of the chemically identical atoms are selected for deletion.

For example, delete atom Type 18 with the following LAMMPS commands.

```
group g18 type 18
delete_bonds all atom 18 remove special
delete_atoms group g18
```

Run brief minimization and dynamics to relax the system after deleting atoms. After deleting these atoms and their bonded interactions, create a LAMMPS data file named

```
data.CorrectedTopology.
```

2.5.2 Step 2: Determine Final Atomic Charges and Force Field Parameters

Determine new charges and force field parameters for the network using Antechamber. The charge of the atoms involved in new cross-linking bonds typically changes because their bonded structure has changed. Typically, changes in atomic charge are mostly localized to the atoms that have formed new bonds and to the atoms bonded to those atoms.

Using a molecule builder (as in Stage 1), create an atomistic model of a cross-linker-centered region of the polymer network, which consists of a cross-linker bonded to as many linear segments as possible. An example of the cross-linker-centered region for DCPD is shown in Fig. 3.

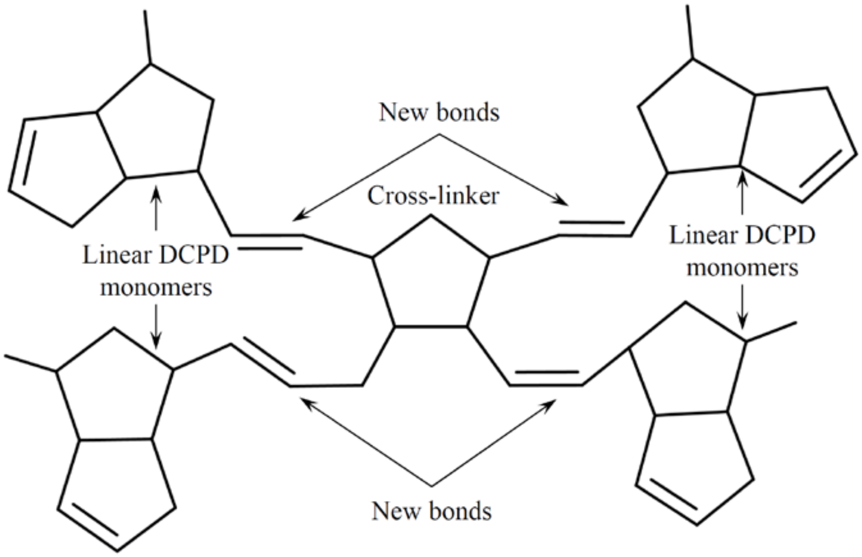

Fig. 3 Cross-linker-centered region for DCPD network. The cross-linker is shown in the center of the diagram. The cross-linker is double-bonded to 4 DCPD monomers with only the norbornene-like ring opened.

The design of the cross-linker-centered region will depend on the chemistry of the network of interest. For DCPD networks, recall that the DCPD linear segment is an oligomer of DCPDmonomers with only the norbornene-like ring opened (Fig. 2). In Fig. 3, however, the DCPD cross-linker is bonded to 4 single DCPD units, rather than to 4 DCPD oligomers. This structure is used to determine the charges of atoms near the location of new bonds. The additional DCPD groups in oligomers would have very little effect on the charge of those atoms. Therefore, the DCPD oligomers are truncated to monomers to reduce the computational cost of the charge calculation. For other polymer networks, different schemes may be required to determine the charges, including the use of multiple structures (e.g., cross-linker-centered and linear-segment-centered).

Next, use AmberTools to calculate charges for the cross-linker-centered region and to find force field parameters for the new bonds, angles, and dihedrals. First, use `Antech` to calculate AM1-BCC charges and assign GAFF atom types for the cross-linker-centered region. As in Stage 1, this step produces a `.mol2` file: `cross-linked.mol2`.

After calculating the charges with antechamber, manually make the following modifications to the charges. First, average charges of atoms that are chemically equivalent (e.g., the carbon atoms in the new double bonds in DCPD, the hydrogen atoms bonded to those carbon atoms, hydrogen atoms on methyl groups). Second, update the charges of the atoms directly involved in new bonds and the 1 to 2 and 1 to 3 neighbors of these atoms (i.e., all atoms within 2 bonds of an atom involved in new bonds) in the cross-linker-centric structure to the charges for the unreacted system. Finally, ensure that the cross-linked network has a zero net charge (or an integer, if the network is supposed to be charged) by rounding the charges on individual atoms or distributing a small charge over all atoms, as in Stage 1.

Use `parmchk` to output all of the `gaff` parameters for the cross-linked network in the file `cross-linked.frcmod`. Example command usage is provided below.

```
$AMBERHOME/bin/parmchk -i cross-linked.mol2 -o cross-linked.frcmod -f mol2 -a Y
```

The file `cross-linked.frcmod` contains force field parameters, such as the following.

```
MASS
c3        12.010  0.878 # atom type, mass, polarizability (not used)
BOND
c2-c2     589.70  1.324 # atom types, spring constant, equil. length
ANGLE
c3-c2-c2  64.330  123.42 # atom types, spring constant, equil. angle
DIHE
c3-c3-c3-c3  1  0.180  0.0    -3.0 # dihedrals
c3-c3-c3-c3  1  0.250  180.0  -2.0
c3-c3-c3-c3  1  0.200  180.0   1.0
```

The parameters for the cross-linking bond (c2-c2) were used in the previous step. These parameters for angles and dihedrals will be used in the next step.

> Note: In the current workflow, any improper dihedral types that are printed in `cross-linked.frcmod` are ignored. Improper dihedrals are typically intended to enforce planarity at sp2 atoms or to prevent racemization at chiral centers. Impropers have not been required for epoxy or DCPD networks because planarity is already enforced with proper dihedrals. It is possible that improper dihedrals will be required for other types of polymer networks. (Note, however, that Findtop [discussed below] is currently incapable of finding impropers.)

The product of this step is a `.mol2` file containing the GAFF atom types and AM1-BCC charges, with the manual changes described above, of the cross-linker-centered region: `cross-linked.mol2`.

2.5.3 Step 3: Add New Angle and Dihedral Interactions to Network Topology

Use the custom code, Findtop, to find and add new angles and dihedrals associated with the new cross-linking bonds. Initially, Findtop is used interactively to find the new bonded interactions. After this first use, Findtop can be used noninteractively as part of an automated workflow (e.g., for building multiple replicas of a network using the same linear segment and cross-linker components).

Run the interactive Findtop program (i.e., `/path/to/findtop`). Findtop asks for name of data file. Type this name (e.g., `data.CorrectedTopology`) and hit `Enter`. Findtop outputs some informational messages while reading the LAMMPS data file, ending with “File has been read successfully”. Findtop asks for the bond type ID of the new bonds. Type this integer number (i.e., the last Bond ID in the `BondCoeffs` section of `data.CorrectedTopology`, and the same bond type ID that was relaxed in Stage 4) and hit `Enter`. Say `y` to the next prompt, `Do you want to enter the new angle and dihedral parameters based on atom type? You will need to force field information`, and hit `Enter`. The program outputs some informational messages about the number of new bonds, angles, and dihedrals and the number of new angle and dihedral types. The program writes 3 files containing the new bonds, angles, and dihedrals that have been detected: `NewBondsDump`, `NewAnglesDump`, and `NewDihedralsDump`. These files contain lines similar to the following (from `NewDihedralsDump` as an example) line

```
129151 30807974 5754 5759 5760 9620
```

where the numbers are 1) dihedral ID, 2) a unique identifier for this dihedral type, and 3) then (3–6) the 4atom IDs in the dihedral. The unique identifier is a combination of the atom types (in this example, Types 30, 80, 79, and 74). These 3 files are simply for informational purposes. The program writes 2 files, `NewAngleTypesDump` and `NewDihedralTypesDump`, which contain lines of only the unique identifier of each new angle or dihedral type found. These lines look like the following (from `NewAngleTypesDump` as an example).

```
010976
010977
010978
```

As in `NewAnglesDump` and `NewDihedralsDump`, the unique identifier is a combination of the atom types involved in the angle or dihedral (i.e., the first angle type in the above example involves Types 1, 9, and 76). The purpose of `NewAngleTypesDump` and `NewDihedralTypesDump` is to indicate to Findtop which angle and dihedral type IDs these new angles and dihedrals will have in the finalized LAMMPS data file.

> Note: If all atoms are defined as having a unique type ID in the LAMMPS data file, there will be many angle and dihedral types identified by Findtop as “unique”. But many of these interactions involve atoms of the same GAFF atom types and will actually use the same force field parameters. In the following steps, the angles and dihedrals that are truly unique will be indicated by the user by editing `NewAngleTypesDump` and `NewDihedralTypesDump`.

The program pauses with a prompt similar to, “Please use the NewAngleTypesDump and `NewDihedralTypesDump` files to enter the types. Use blank space to add the multiplier term followed by the new type/types. Press y to continue or n to close . . .” Before proceeding, the indicated files must be edited to include information about the new angle and dihedral types. Edit `NewAngleTypesDump` to indicate the angle type ID of each of the angle types in the file. The following instructions will describe how to edit the file. The file will look similar to the following after editing.

```
010976 1 20
010977 1 20
010978 1 21
```

The first number is the unique angle identifier written by Findtop; do not change this number. The second number is a multiplier term that indicates how many interaction types will follow (i.e., how many numbers remain on the line). The second number should always be 1 for angles, but may be greater than 1 for dihedrals. The third number indicates the angle type ID. The number of the new angle type IDs in `NewAngleTypesDump` should start after the number of angle types already defined in the LAMMPS data file. In this example, the LAMMPS data file already has 19-angle types, so the first new angle type ID is 20. The first 2-angle types (`010976` and `010977`) are actually identical so they have the same angle type ID (20). But the third angle type is unique, so it has a unique angle type ID (21).

To determine the angle type IDs to assign to each of the angle types identified by Findtop, use the LAMMPS data file (`data.CorrectedTopology`) and the `mol2` file created with `antechamber` for the cross-linker-centered region (`cross-linked.mol2`, from Step 2 of this stage). For each angle identifier in `NewAngleTypesDump` (e.g., `010976`), cross-reference the numeric LAMMPS atom types that comprise the angle identifier (e.g., 1, 9, and 76) with `data.CorrectedTopology` and `cross-linked.mol2` to determine the alphanumeric GAFF atom types of those numeric LAMMPS atom types (e.g., c2, c3, hc). Assign all angle identifiers in `NewAngleTypesDump` that correspond to a particular set of GAFF atom types (e.g., c2-c2-c3, c2-c2-ha) the same angle type ID. In the example above, the angle identifiers, 010976 and 010977, could both correspond to the gaff angle type c2-c2-ha, while angle identifier 010978 could correspond to the GAFF angle type c2-c2-c3.

> Note: simultaneously visualizing `data.CorrectedTopology` and `cross-linked.mol2` with VMD can aid in identifying which LAMMPS atom types correspond to which GAFF atom types.

Similarly, edit `NewDihedralTypesDump` to indicate the dihedral type ID of each of the dihedral types in the file. The file will look similar to the following after editing.

```
30807974 1 10
30807975 2 11 12
30807976 3 13 14 15
```

The first number is the unique dihedral identifier written by Findtop. The second number is the multiplier term that indicates how many dihedral types will follow (i.e., how many numbers remain on the line). The remaining numbers on each line are the dihedral type ID(s). The number of new dihedral type IDs in `NewDihedralTypesDump` should start after the number of dihedral types already defined in the LAMMPS data file. In this example, the LAMMPS data file already has 9 dihedral types, so the first new dihedral type ID is 10. The first dihedral type (`30807974`) only has a single torsion potential, but the other dihedral types (`30807975` and `30807976`) have 2 and 3 torsion potentials each, respectively.

> Note: Applying multiple torsion potentials to a single quartet of atom types allows force field developers to create a more realistic potential energy function with multiple minima and barriers of varying depths and heights.

Determine the dihedral type IDs in a similar manner as the angle type IDs. To determine the multiplicity of each dihedral type (i.e., the number of torsion potentials for this type), examine the file `cross-linked.frcmod`. As an example, these are the force field parameters for a dihedral with a multiplicity of 3.

```
c3-c3-c3-c3   1  0.18    0.0  -3.0
c3-c3-c3-c3   1  0.25  180.0  -2.0
c3-c3-c3-c3   1  0.20  180.0   1.0
```

The first column contains the atom types in this dihedral. The second column contains a divisor for the barrier height; it is always 1 in the `frcmod` file format. The third column contains the barrier height. The fourth column is the phase shift. The fifth column contains the absolute value of the periodicity. A negative value in the fifth column indicates that the next line of dihedral parameters also apply to this quartet of atom types. A positive value in the fifth column indicates that the next line applies to a different quartet of atom types. Because there are 2 lines ending in a negative value followed by a final line ending in a positive value, the multiplicity of the c3-c3-c3-c3 dihedral is 3.

After editing NewAngleTypesDump and NewDihedralTypesDump to include the above information, copy the files to another location to prevent them from being overwritten. Then, returning to the Findtop program, press they y ke and then press enter to continue the execution of Findtop, which will next ask for the name of the LAMMPS data file that will contain the new angles and dihedrals: data.CorrectedTopologyInternals. This data file contains all the new angles and new dihedrals but does not yet contain the necessary force field parameters (coefficients) for the new angle types and new dihedral types.

To find force field parameters for the new angle and dihedral types, open the file cross-linked.frcmod that was created in a previous step. Locate the lines containing the parameters for each unique GAFF angle type identified in the previous step (e.g., c3-c2-c2). The order of the atom types may be reversed (e.g., c3-c2-c2 is identical to c2-c2-c3).

> Note: For a typical AmberTools installation, the entire set of GAFF parameters is located in $AMBERHOME/dat/leap/parm/gaff.dat. The parameters in cross-linked.frcmod are a subset of those in gaff.dat.

Add the force field parameters for the new angle and dihedral types to data.CorrectedTopologyInternals. Adding angle coefficients to the LAMMPS data file is straight-forward. For example, the gaff coefficients for the c3-c2-c2 angle are as follows.

```
c3-c2-c2  64.330  123.420
```

These coefficients should appear as follows in LAMMPS data file format

```
17 64.33 123.42
```

where 17 is the angle type ID.

Adding dihedral coefficients to a LAMMPS data file is also simple but less straight-forward. Using the example of the c3-c3-c3-c3 dihedrals, the GAFF format is as follows.

```
c3-c3-c3-c3  1      0.18   0.0    -3.0
c3-c3-c3-c3  1      0.25   180.0  -2.0
c3-c3-c3-c3  1      0.20   180.0  1.0
```

In LAMMPS data file format, these dihedral potentials are as follows.

```
1      0.18    1     3
2      0.25    1     2
3      0.2    -1     1
```

In the LAMMPS data file format, the first column is the dihedral type ID. The second column is the barrier height, which is in the same units as in the AMBER format. The third column is the phase shift: a phase shift of 0°in AMBER format translates into a phase shift of 1 in LAMMPS format, and a phase shift of 180° in AMBER format translates into a phase shift of -1 in LAMMPS format. The fourth column is the periodicity; the periodicity must be a *positive integer* in LAMMPS format

Edit the Lennard-Jones parameters (Pair Coeffs) in `data.CorrectedTopologyInternals` to update the parameters for any atoms for which the `GAFF` atom type has changed as a result of cross-linking (i.e., atoms that have gained or lost bonds and atoms that have bonded to those atoms). These non-bonded parameters are listed in `cross-linked.frcmod` in the NONBON section. The following example shows the non-bonded parameters.

```
NONBON
c3  1.9080  0.1094 # atom type, van der Waals radius, L-J well depth
c2  1.9080  0.0860
hc  1.4870  0.0157
ha  1.4590  0.0150
```

Both the radius and well depth may differ between atom types of the same element. Identify atoms that have new atom types by comparing `cross-linked.mol2` with the `.mol2` files generated in Stage 1. Determine the atom type IDs of these atoms by visualizing `data.CorrectedTopologyInternals`. In the `Pair Coeffs` section of `data.CorrectedTopologyInternals`, edit the coefficients of these atom types. A conversion from the AMBER format to the LAMMPS format must be performed. The `Pair Coeffs` section for the above 4 atom types (c3, c2, hc, ha) would appear as follows:

```
1 0.1094 3.39967
2 0.0860 3.39967
3 0.0157 2.64953
4 0.0150 2.59964
```

The first column is the LAMMPS atom type ID. The second column is the Lennard-Jones well depth, which is in the same units as in the AMBER format. The third column is the Lennard-Jones diameter (double the L-J sigma), which differs from the van der Waals radius (i.e., the distance at the potential energy minimum, or 2^[1/6] times sigma) given in the AMBER format. Multiply the AMBER format van der Waals radius by a factor of 2/(2^[1/6]) to convert to the LAMMPS format Lennard-Jones diameter.

After this initial process of determining the angle and dihedral types and coefficients, Findtop can be used non-interactively to assist in creating automated workflows. The non-interactive version of Findtop uses a configuration file in the current directory, `FindtopConfig`, to gather the required information. `FindtopConfig` has the following form.

```
InputDatafile              data.CorrectedTopology
OutputDatafile             data.CorrectedTopologyInternals
NewBondID                  8
NewAngleTypesDumpName      /path/to/NewAngleTypesDump.final
NewDihedralTypesDumpName   /path/to/NewDihedralTypesDump.final
NewAngleCoeffsDumpName     /path/to/NewAngleCoeffsData.final
NewDihedralCoeffsDumpName  /path/to/NewDihedralCoeffsData.final
```

`InputDatafile`, `NewBondID`, and `OutputDatafile` are the same inputs required by the interactive version of Findtop: the data file to read, the bond type ID of the new cross-linking bonds, and the name of the data file to write with new angles and dihedrals. The files `NewAngleTypesDump.final` and `NewDihedralTypesDump.final` have the same format as the edited versions of `NewAngleTypesDump` and `NewDihedralTypesDump` created above. The noninteractive version of Findtop reads these files instead of requiring user input. The additional files listed in `FindtopConfig`, `NewAngleCoeffsData.final` and `NewDihedralCoeffsData.final`, list the new coefficients angle and dihedral coefficients, which Findtop automatically writes into the output data file (`data.CorrectedTopologyInternals`). The format of `NewAngleCoeffsData.final` and `NewDihedralCoeffsData.final` is similar to the following.

```
2
12 64.33 123.42
13 50.04 120.94
```

The first line, a single integer, indicates how many angle or dihedral coefficients will follow (i.e., the number of lines remaining in the file). The remaining lines are the angle or dihedral type IDs and coefficients in the same format as in the LAMMPS data file.

2.5.4 Step 4: Apply Post-Reaction Atomic Charges to Network

Apply the new charges and use brief minimization and dynamics to relax the new angles and dihedrals. Use the data file from the previous step, `data.CorrectedTopologyInternals`, as the starting point for this step.

Update the charges of the cross-linked network using LAMMPS commands similar to the following.

```
group g01 type 9 30
set group g01 charge -0.161
```

These commands set the charges of atom Types 9 and 30 to –0.161. Use similar commands to update the charges of all atom types for which the charge has changed as a result of cross-linking.

Conduct brief minimization and dynamics at elevated temperature with the following LAMMPS commands.

```
# Minimize
min_style     sd
minimize      0 0 10000 10000
min_style     cg
minimize      0 0 10000 10000
# Run dynamics at 500 K for 100 ps
Fix           1 all npt temp 500.0 500.0 100.0 iso 1.0 1.0 1000.0
Run           100000 #fs
Unfix         1
# Write final configuration
write_restart restart.CorrectedTopologyInternalsCharges
```

Create a data file (`data.CorrectedTopologyInternalsCharges`) containing the final configuration of the cross-linked network.

In the final product of this stage, `data.CorrectedTopologyInternalsCharges`, the network topology and force field are finalized, and any extremely unfavorable structures (e.g., highly bent angles or twisted torsions) have been eliminated with relatively short minimization and dynamics. The network still must be relaxed toward its equilibrium state using long molecular dynamics simulations.

2.6 Stage 6. Relax Network Structure with Molecular Dynamics Simulated Annealing

Use long molecular dynamics simulations to relax the network to its equilibrium structure. The topology and force field of the cross-linked network are now finalized, but the structure of the network is most likely not similar to the equilibrium structure.

Run molecular dynamics at high temperature (T_g + 150 K or higher, e.g., approximately 735 K for epoxy) for 5 to 10 ns to equilibrate the density and potential energy of the network in the isothermal-isobaric ensemble (NPT).

Monitor the density (or specific volume) of the network with the following LAMMPS commands, which record a rolling average over 1,000 steps of the temperature, potential energy, kinetic energy, enthalpy, pressure, and specific volume in a file named `sv.TEMPERATURE`.

```
variable      temp equal temp
variable      massgrams equal mass(all)/6.0225e23
variable      vcm3 equal vol*1.0e-24
variable      sv equal v_vcm3/v_massgrams
variable      pe equal pe
variable      ke equal ke
variable      h equal enthalpy
variable      press equal press
fix           2 all ave/time 1 1000 1000 v_temp v_pe v_ke v_h v_press v_sv &
              file sv.TEMPERATURE
```

Monitor the bond lengths of the new cross-linking bonds with LAMMPS commands similar to the commands used in Stage 4. Monitoring the bond length distribution allows us to observe whether a particular network has overly stretched bonds, which may indicate that that network topology is unphysical and should be discarded.

Examples of LAMMPS commands to run dynamics for 5 ns at 735 K follow.

```
# Set temperature   variable
variable T index    735
# Create temperature-specific log file
log                 log.$T
# NPT integration at T
fix 1 all npt temp $T $T 100 x 1.0 1.0 1000 y 1.0 1.0 1000 z 1.0 1.0 1000 &
       couple none
# Run dynamicsrun   5000000  # fs
# Write restart at end of this equilibration
write_restart rst.$T
```

Perform simulated annealing starting at high temperature (e.g., from 735 K as in the previous step) and anneal to lower temperature (e.g., 200 K) in a step-wise fashion (e.g., 15 K/2 ns). This step is a different form of simulated annealing than the method used earlier to obtain the network connectivity. With this step, classical molecular dynamics is used rather than Monte Carlo. The slowest cooling rate with a reasonable computational cost should be used. The range of temperatures covered should start far above T_g in the rubbery range and should end far below T_g in the glassy range.

Example LAMMPS commands for simulated annealing with molecular dynamics are given below.

```
# List of temperatures to simulate (with some omitted at [...] for brevity in
this document)
variable T index  720 705 690 675 [...] 270 255 240 225
# Begin looping over T
label loop
log   log.$T
fix   1 all npt temp $T $T 100 x 1.0 1.0 1000 y 1.0 1.0 1000 z 1.0 1.0 1000 &
       couple none
# Set the length of time at this temperature based on cooling rate
run 2000000 #fs
# write restart and kill the current fixes
write_restart rst.$T
unfix 1
unfix 2

next T # move to next temp
jump SELF loop
# end of loop
```

3. Application

After constructing and annealing cross-linked networks, various physical properties of the polymer can be calculated. Consider the following example of DCPD networks with different cross-link densities: DCPD-3 where the linear segments consist of 3 DCPD monomers (see Fig. 2b), and DCPD-6 where the linear segments contain 6 DCPD monomers. The cross-link density of DCPD-3 is approximately double that of DCPD-6. Experimentally, the cross-link density in DCPD polymers can be controlled by varying the reaction conditions (e.g., catalyst, temperature, concentration).

The DCPD-3 networks contain 1,400 cross-linkers, 2,800 linear segments (each containing 3 DCPD monomers), and a total of 215,600 atoms. The DCPD-6 networks contain 729 cross-linkers, 1,458 linear segments (each containing 6 DCPD monomers), and a total of 208,494 atoms. To assess the statistical uncertainty of the data, 5 replicas of both cross-linked polymers were constructed using different initial velocities in the Stage 2 equilibration step, which results in different network connectivity in Stage 3. The replicas have the same composition but different network topology, and they represent nanoscopic samples spatially distributed throughout a macroscopic material. Two properties of these networks, the specific volume and the coefficient of volumetric thermal expansion (CVTE), are shown in Fig. 4. Specific volume is calculated as the total volume of the polymer network divided by the total mass of the network. CVTE is calculated using the equation $CVTE = \frac{1}{\hat{v}}\left(\frac{\delta \hat{v}}{\delta T}\right)_P$, where $\hat{v}$ is the specific volume at each temperature (T), and the subscript, $_{P,}$ indicates that the partial derivative is taken at constant pressure. The error bars in Fig. 4 are the standard deviation of the means of the 5 replicas.

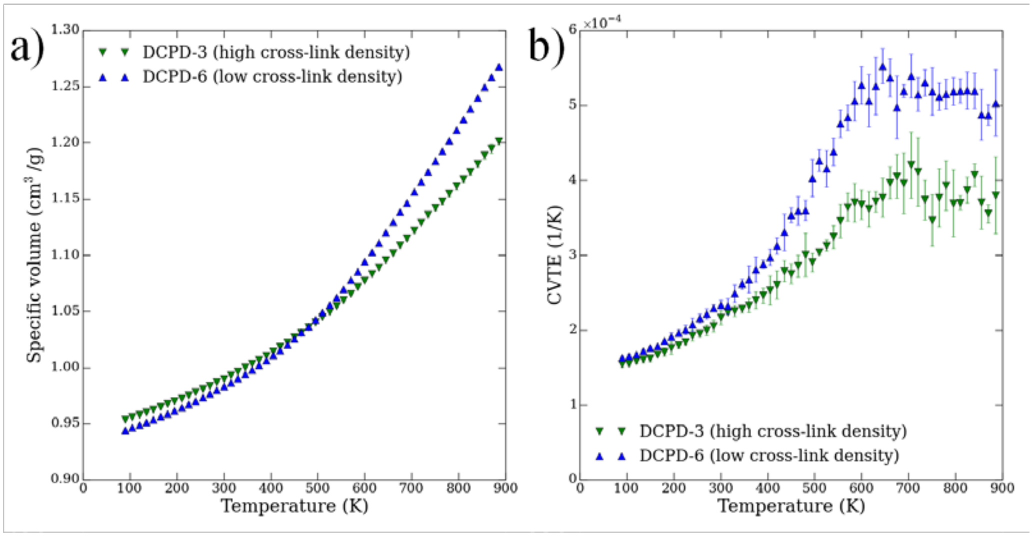

Fig. 4 a) Specific volume and b) CVTE of 2 poly(dicyclopentadiene) networks with differing cross-link density

The simulated specific volume of both DCPD-3 and DCPD-6 at 300 K is approximately 0.97 cm^3/g, which is in good agreement with the experimental value of 0.96 cm^3/g (corresponding to a density of 1.04 g/cm^3).[26, 27] The simulated CVTE values are somewhat lower than expected from experimental measurements of the coefficient of linear thermal expansion,[27] which can be approximated as one-third the CVTE. The experimental values of CVTE in the glassy and rubbery states are 2.6×10^{-4} K^{-1} and 6.2×10^{-4} K^{-1}, [(27)] respectively, whereas the simulated CVTE values are approximately 25% lower (approximately 2×10^{-4} K^{-1} and approximately 4.5×10^{-4} K^{-1}). As suggested in previous studies,[12,16] the lower values of CVTE in simulations can be attributed to complete conversion and 100% cross-linking in the simulated network. Whereas, in experimental samples, complete conversion and cross-linking are likely hindered by topological and steric constraints.

We observe the following effects of cross-link density on the specific volume and CVTE in Fig. 4. The specific volume (Fig. 4a) in the glassy state (i.e., below approximately 450 K) is approximately equal for DCPD-3 and DCPD-6, which is consistent with previous simulations of epoxy networks with varying cross-link density.[16, 28] This finding suggests that the density in the glassy state is relatively insensitive to small changes in cross-link density. Likewise, the cross-link density has little effect on CVTE in the glassy state (Fig. 4b). We note, however, that much larger differences in cross-link density than those considered here can lead to larger differences in density and CVTE in the glassy state, as observed in previous studies.[16] Above the glass transition temperature, the specific volume of DCPD-6 is significantly higher than that of DCPD-3. This change with increasing temperature occurs because the longer linear segments in DCPD-6 are entropically driven to extend by a greater amount than the shorter linear segments in DCPD-3. This difference in expansion with increasing temperature is also evident in Fig. 4b, which shows that the CVTE of DCPD-6 in the rubbery state is higher than the rubbery state of DCPD-3, as expected for a polymer network with lower cross-link density.[16]

The above application of the simulated annealing polymerization method to poly(dicyclopentadiene) demonstrates how simulations can be used to qualitatively, and often quantitatively, characterize cross-linked polymer networks and to provide insight that can guide the design of new materials.

4. Conclusions

The details of the simulated annealing polymerization method for constructing cross-linked polymer networks have been described. This one-step simulated annealing method is appealing because it is computationally efficient and, therefore, allows the construction of larger cross-linked networks than is feasible with other methods. These instructions can be used to construct a cross-linked polymer network with any type of chemistry. We have demonstrated the application of this method to cross-linked networks of poly(dicyclopentadiene) with varying cross-link density. The ability to quickly and easily build computational models of cross-linked polymer networks with varying chemistry will aid in the rational design of improved materials. Future directions for this work include adding the ability to tune the cure (i.e., less than 100% cross-linking) and the connectivity (e.g., to create regions of heterogeneous cross-link density). Another goal is to further automate the method so that less user intervention is required and new networks can be constructed more rapidly.

5. References

1. Knorr Jr DB, Yu JH, Richardson AD, Hindenlang MD, McAninch IM, La Scala JJ, Lenhart JL. Glass transition dependence of ultrahigh strain rate response in amine cured epoxy resins. Polymer 2012;53(25):5917–5923.

2. Opportunities in protection materials science and technology for future army applications. Washington (DC): The National Academies Press (US); 2011.

3. Bogoslovov RB, Roland CM, Gamache RM. Impact-induced glass transition in elastomeric coatings. Applied Physics Letters. American Institute of Physics; 2007;90(22).

4. McGrath LM, Parnas RS, King SH, Schroeder JL, Fischer DA, Lenhart JL. Investigation of the thermal, mechanical, and fracture properties of alumina–epoxy composites. Polymer 2008;49(4):999–1014.

5. Lee H, Neville K. Handbook of epoxy resins. New York: McGraw-Hill; 1967.

6. Aldridge M, Shankar C, Changgua Zhen, Lang Sui, Kieffer J, Caruso M, Moore J. Combined experimental and simulation study of the cure kinetics of DCPD. Journal of Composite Materials 2010;44(22):2605–2618.

7. Nowers JR, Broderick SR, Rajan K, Narasimhan B. Combinatorial methods and informatics provide insight into physical properties and structure relationships during IPN formation. Macromolecular Rapid Communications 2007;28(8):972–976.

8. Kalcioglu ZI, Mrozek RA, Mahmoodian R, VanLandingham MR, Lenhart JL, Van Vliet, KJ. Tunable mechanical behavior of synthetic organogels as biofidelic tissue stimulants. Journal of Biomechanics 2013;46(9):1583-1591.

9. Mrozek RA, Cole PJ, Otim KJ, Shull KR, Lenhart JL. Influence of solvent size on the mechanical properties and rheology of polydimethylsiloxane-based polymeric gels. Polymer 2011;52(15):3422–3430.

10. Bandyopadhyay A, Odegard GM. Molecular modeling of crosslink distribution in epoxy polymers. Modelling and Simulation in Materials Science and Engineering 2012;20(4):045018.

11. Fan HB and Yuen MMF. Material properties of the cross-linked epoxy resin compound predicted by molecular dynamics simulation. Polymer 2007;48(7):2174–2178.

12. Lin P-H, Khare R. Molecular simulation of cross-linked epoxy and epoxy−POSS Nanocomposite. Macromolecules 2009;42(12):4319–4327.

13. Sirk TW, Khare KS, Karim M, Lenhart JL, Andzelm JW, McKenna GB, Khare R. High strain rate mechanical properties of a cross-linked epoxy across the glass transition. Polymer 2013;54(26):7048–7057.

14. Khare KS, Khare R. Effect of carbon nanotube dispersion on glass transition in cross-linked epoxy–carbon nanotube nanocomposites: Role of Interfacial Interactions. J Phys Chem B 2013;117(24):7444–7454.

15. Sliozberg YR, Mrozek RA, Schieber JD, Kröger M, Lenhart JL, Andzelm JW. Effect of polymer solvent on the mechanical properties of entangled polymer gels: Coarse-grained molecular simulation. Polymer 2013;54(10):2555–2564.

16. Soni NJ, Lin P-H, Khare R. Effect of cross-linker length on the thermal and volumetric properties of cross-linked epoxy networks: A molecular simulation study. Polymer 2012;53(4):1015–1019.

17. Christensen S. Using molecular dynamics coupled with higher length scale simulations for the development of improved composite matrix materials. Proceedings of the 17th International Conference on Composite Materials; 2009 Jul 27–31; Edinburgh, UK.

18. Li C and Strachan A. Molecular simulations of crosslinking process of thermosetting polymers. Polymer 2010;51(25):6058–6070.

19. Wu C and Xu W. Atomistic molecular modelling of crosslinked epoxy resin. Polymer 2006;47(16):6004–6009.

20. Heine DR, Grest GS, Lorenz CD, Tsige M, Stevens MJ. Atomistic simulations of end-linked poly(dimethylsiloxane) networks: structure and relaxation. Macromolecules 2004;37(10):3857–3864.

21. Varshney V, Patnaik SS, Roy AK, Farmer BL. A molecular dynamics study of epoxy-based networks: cross-linking procedure and prediction of molecular and material properties. Macromolecules 2008;41(18):6837–6842.

22. Yarovsky I, Evans E. Computer simulation of structure and properties of crosslinked polymers: application to epoxy resins. Polymer 2002;43(3):963–969.

23. Plimpton S. Fast parallel algorithms for short-range molecular dynamics. J Comp Phys 1995;117(1):1–19.

24. Jakalian A, Bush BL, Jack DB, Bayly CI. Fast, efficient generation of high-quality atomic charges. AM1-BCC model: I. Method. J Comp Chem 2000;21(2):132–146.

25. Wang J, Wolf RM, Caldwell JW, Kollman PA, Case DA. Development and testing of a general amber force field. J Comp Chem 2004;25(9):1157–1174.

26. Constable GS, Lesser AJ, and Coughlin EB. Morphological and mechanical evaluation of hybrid organic−inorganic thermoset copolymers of dicyclopentadiene and mono- or tris (norbornenyl)-substituted polyhedral oligomeric silsesquioxanes. Macromolecules 2004;37(4):1276–1282.

27. Knorr Jr DB, Hindenlang MD, Richardson AD, Yu JH, Lenhart JL. Private communication.

28. Sirk TW, Karim M, Khare KS, Lenhart JL, Andzelm JW, Khare R. Private communication.

Appendix A. List of Required Software

- AmberTools12—constructing initial topology and coordinate files using the general AMBER force field (GAFF); freely available at http://ambermd.org/.

- LAMMPS—running minimization and dynamics and converting several file formats; freely available at http://lammps.sandia.gov

- CambridgeSoft ChemOffice, VegaZZ, MarvinSketch—building molecular structures

- VMD—visualizing molecular structures; freely available at http://www.ks.uiuc.edu/Research/vmd

- In-house US Army Research Laboratory codes

    - ClassLAMMPSDataFile—reads, edits, and writes LAMMPS data files.
    - CombinedReactants4Epoxy—combines LAMMPS data files.
    - SimulatedAnnealing—determines network connectivity using Monte Carlo approach.
    - Findtop—identifies new angle/dihedral interactions created by forming cross-link bonds.

Appendix B. Generic Header for LAMMPS Input Files

This appendix appears in its original form without editorial change.

Use these commands prior to the commands listed in each “Example LAMMPS input” section. Note that the name of the data file below (data.NAME) must be changed.

```
#Simulation parameters
units               real
neighbor            10 bin
neigh_modify        delay 5 every 1 page 2000000 one 200000
atom_style          full
bond_style          harmonic
angle_style         harmonic
dihedral_style      harmonic
pair_style          lj/cut/coul/cut 9
pair_modify         mix arithmetic
boundary            p p p
special_bonds       amber

# Read data file
read_data           data.NAME
# Generate velocities
velocity            all create 735.0 12345 dist gaussian loop local

# Calculate thermodynamic quantities
variable            mass_g equal mass(all)/6.022e23
variable            vol_cm3 equal vol*1.0e-24
variable            mydensity equal v_mass_g/v_vol_cm3
thermo_style        custom step temp press vol etotal epair emol evdwl ecoul
elong ebond         eangle enthalpy pe ke v_mydensity lx ly lz cpu
thermo              10000
timestep            1.0 #fs
```

List of Symbols, Abbreviations, and Acronyms

Å	ångström
AMBER	Assisted Model Building with Energy Refinement
AM1	Austin Model 1
BCC	bond charge corrections
CVTE	coefficient of volumetric thermal expansion
db	equilibrium bond length
DCPD	dicyclopentadiene
GAFF	general AMBER force field
ID	identification
K	Kelvin
kb	bond spring constant
LAMMPS	Large-scale Atomic/Molecular Massively Parallel Simulator
NPT	isothermal-isobaric ensemble
T	temperature
T_g	glass transition temperature
$\hat{v}$	specific volume
VMD	Visual Molecular Dynamics
δ	partial derivative operator

1 DEFENSE TECHNICAL
(PDF) INFORMATION CTR
DTIC OCA

2 DIRECTOR
(PDF) US ARMY RESEARCH LAB
RDRL CIO LL
IMAL HRA MAIL & RECORDS MGMT

1 GOVT PRINTG OFC
(PDF) A MALHOTRA

1 DIR USARL
(PDF) RDRL WMM G
R ELDER