SANDIA REPORT

SAND2002-0604
Unlimited Release
Printed March 2002

Molecular Simulation of Reacting Systems

Aidan P. Thompson

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/ordering.htm

SAND2002-0604
Unlimited Release
Printed March 2002

Molecular Simulation of Reacting Systems

Aidan P. Thompson
Computational Materials & Molecular Biology
Sandia National Laboratories
P.O. Box 5342
Albuquerque, NM 87185-0316

Abstract

The final report for a Laboratory Directed Research and Development project entitled, *Molecular Simulation of Reacting Systems* is presented. It describes efforts to incorporate chemical reaction events into the LAMMPS massively parallel molecular dynamics code. This was accomplished using a scheme in which several classes of reactions are allowed to occur in a probabilistic fashion at specified times during the MD simulation. Three classes of reaction were implemented: addition, chain transfer and scission. A fully parallel implementation was achieved using a checkerboarding scheme, which avoids conflicts due to reactions occurring on neighboring processors. The observed chemical evolution is independent of the number of processors used. The code was applied to two test applications: irreversible linear polymerization and thermal degradation chemistry.

Contents

1. Introduction

2. Reactive Molecular Dynamics: Algorithm and Code Development

3. Reactive Molecular Dynamics: Example Applications

    3.1 Linear Polymer Growth

    3.2 Thermal and Oxidative Degradation

4. Summary and Discussion

5. References

Figures

Figure 3.1 Schematic of the irreversible polymerization process. At each MD timestep, the reactive sites (gray) can bond with monomers (white).

Figure 3.2 Polymer chain length versus time for several different reaction rates. The time axis is scaled by the reaction time constant, so that in the absence of diffusion limitation, all the curves would coincide with the top curve.

Figure 3.3 Snapshots taken from 2D (left) and 3D (right) simulations of linear irreversible polymerization. In the left image, the reactive chain end is clearly trapped within the coiled polymer. In the right image, no such trapping is observed.

Figure 3.4 Average trapping time versus degree of polymerization in 2D (left) and 3D (right). In the 2D case, results are shown for three different reaction rates. In the 3D case, results are shown for two different reaction rates.

Figure 3.5 Photographs of a virgin and an aged o-ring cross-section.

Figure 3.6 Outline of the reaction chemistry for thermal degradation.

Figure 3.7 Complete list of input commands required to specify the thermal degradation chemistry. To save space, the bond type commands are listed in the second column.

Figure 3.8 Evolution of a system undergoing thermal degradation chemistry, starting from a dense fluid of 2000 methylene radicals. The normal species are shown in black and the radical species in red.

Figure 3.9 Blow-up of Figure 3.7, showing only the two most dilute species. In each cases the four curves shown are the results obtained running on 1, 2, 4 and 8 nodes of CPlant. The thick vertical lines indicate the size of the error bars (one standard deviation confidence interval).

1. Introduction

Conventional modeling approaches to chemically reactive systems fall into two broad categories: a) *ab initio* quantum calculations of reaction paths involving the coordinates of only a handful of isolated atoms b) macroscopic kinetic models in which molecular interactions are not explicitly represented. Neither of these approaches can address the strong coupling between chemical reaction and molecular diffusion which frequently occurs in liquids and amorphous materials such as rubber polymers. At the outset of this project, the aim was to a) develop molecular simulation methods which incorporate chemical reaction events into the conventional framework of molecular dynamics simulation of liquids and amorphous materials b) develop efficient parallel implementations of these methods for use on massively parallel machines such as ASCI Red and Cplant c) apply the methods to the following problems, chosen for their programmatic relevance, broader technological importance, and also for the expected pay-off, in terms of scientific impact:

1. Diffusion-limited curing of glassy polymers, which occurs during encapsulation of weapon components
2. Atomic-scale simulation of non-linear chaos in liquid-state reaction-diffusion systems.
3. Reaction-diffusion coupling in oxidative degradation of organic polymers, which is of fundamental importance to aging of polymer materials in the nuclear stockpile.
4. Thermal degradation of polymer networks. Elevated temperatures cause strong local concentrations of kinetic energy, resulting in spontaneous breaking of covalent bonds and subsequent loss of physical properties.

This ambitious program was not accomplished in its entirety. However, considerable progress towards these goals was made. Specifically, a simple approach to representing a general class of reactions in the LAMMPS molecular dynamics code was developed. This was done in a fully parallel manner, so that the results were independent of the number of processors used. This enabled us to simulate systems containing one million atoms on 64 processors of ASCI Red. The code has also been ported to CPlant. The method was applied to two test problems: linear polymer growth and thermal degradation. This second application is a prototype for chemical aging of organic materials, of great importance to the nuclear weapon stockpile. The reactive dynamics code is now being used to study chemical aging as part of the ASCI M&PM project "Degradation of Organic Materials".

2. Reactive Molecular Dynamics: Algorithm and Code Development

In this section, we describe a code that has been developed to represent chemical reaction events in a molecular dynamics simulations. The key features of this approach are:

1. The code is based on the LAMMPS parallel MD code (Ref. 1)
2. The set of available chemical reactions is quite general, and entirely specified in the input script
3. The code is fully parallel. Like LAMMPS, it can be run on an arbitrary number of processors and the results are independent of the number of processors used.
4. The code was developed and tested on CPlant.

In this next section we describe the reactive dynamics methodology, and how the code is used. In a subsequent section we present results for two specific applications.

A variety of approaches are possible for introducing chemical reaction into a molecular dynamics simulation. Quantum electronic structure calculations can be used to study the dynamics of a small number of atoms during a single reaction event. Tight-binding provides an approximate way of doing the same thing with far less effort. In recent years, reactive force fields have been developed that eliminate the need to treat the electronic structure. Because they are parameterized to reproduce known features of the full potential energy surface, such as bond energies and activation energies. We have pursued an even more simplified approach, which is to treat the chemical reactions as discrete events that are allowed to occur in a probabilistic fashion. The major deficiency of this approach compared to the previously mentioned methods is that the reaction events cause discontinuous changes in the energies and forces of the system. Hence, we can not use it to study the detailed dynamics of a system immediately before and after a reaction. However, it is a viable approach for looking at the evolution of reactive systems on longer timescales. This is particularly true for polymeric systems, where relaxation times exist that are much longer than the timescale for relaxation in the immediate vicinity of a reaction event.

The discrete event approach (with or without the probabilistic component) has been used by several groups in the past, as it is conceptually straightforward. We based our implementation on a scheme proposed by Toxvaerd *et al.* (Ref. 1) for bond formation. They used the scheme to model linear polymer growth, which we will describe in detail in section 3. The key is to associate a particular atom (the reactive site) with each potential reaction event. In the case of a bond formation reaction, either one of the atoms forming the bond can be chosen. After a prescribed period of time, an attempt is made to form a bond at each reactive site. The bond formation probability is proportional to the number of partners within a prescribed radius. By making reaction radius sufficiently small, bond formation will only occur between atoms that have collided strongly with each other, which qualitatively matches the true physics of chemical dynamics. Also, by considering the total number of reactive partners, we preserve first order

kinetics, which is important if more than one partner is available for reaction. The same approach can be used to represent other types of reactions. In general, the probability P of an addition reaction i occurring at a reactive site of type j_{i1} on timestep t_n is given by:

$$\begin{aligned} P &= P_i N(j_{2i}, r \le r_i), \quad \mathrm{mod}(t_n, t_i) = 0 \\ &= 0, \qquad\qquad\quad \mathrm{mod}(t_n, t_i) \neq 0 \end{aligned} \tag{2.1}$$

where $N(j_{i1}, r \le r_i)$ is the number of atoms of type j_{i2} within a distance r_i of the reactive atom of type j_{i2}. The parameters j_{i1}, j_{i1}, P_i, r_i, and t_i are specified for each reaction in the input file. In this project, we have treated three classes of reaction: addition, chain transfer and scission. Addition was implemented exactly as described above. Chain transfer is identical to addition except that no bond is formed. Scission is the reverse process of addition. In this case, if the bond length is *greater* than r_i, the probability of bond scission is P_i.

$$\begin{aligned} P &= P_i N(j_{2i}, r \ge r_i), \quad \mathrm{mod}(t_n, t_i) = 0 \\ &= 0, \qquad\qquad\quad \mathrm{mod}(t_n, t_i) \neq 0 \end{aligned} \tag{2.2}$$

With this conceptual picture in mind, we can now discuss how the method was implemented into the LAMMPS code. In doing this, we had two goals: generality and parallelism. It was important to be able to specify a large set of different chemistries without having to further modify the code. Rather, it should be possible to specify the details of the chemistry entirely from the LAMMPS input script. Secondly it was important that the code exploit the same spatial decomposition of atoms used in the main body of the LAMMPS code.

To achieve generality, we created some new datastructures in LAMMPS. For each of the three classes of reactions described above the user must provide a style flag and parameters for each specific reaction type in that class. Currently, each class of reaction has just one active style, other than `none`. All the current reaction style commands are listed below:

```
addit style     none
trans style     none
sciss style     none
addit style     proximity/addition 256
trans style     proximity/transfer 256
sciss style     extension/scission 256
```

The integer argument is the seed used to initialize the sequence of pseudorandom numbers used for that reaction class. A value of zero indicates that the sequence should continue from the current value. This is useful when performing a series of independent simulations from a single input script, as it does not require specifying a different seed for each simulation.

These reaction styles all have the same syntax for specifying reaction types:

```
      A          B   C D    E F    G     H     I
addit coeff      1   6 6    2 2   1.12  0.05  0.01
addit coeff      2   6 7    2 3   1.12  0.05  0.01
:
```

```
trans coeff  1  6 1   1 6  1.12 0.05  0.01
trans coeff  2  6 2   1 7  1.12 0.05  0.01
:
sciss coeff  1  2 2   6 6  1.12 0.05  0.01
sciss coeff  2  2 3   6 7  1.12 0.05  0.01
:
```

In each case, the numerical arguments are:

B: Reaction index: an integer between 1 and the maximum allowable number of reactions in the class, specified by the parameters `maxadditiontype`, `maxtransfertype` and `maxscissiontype`, respectively The reaction index of the first reaction type of each class specified in the input script must be one, the second must be two, and so on.

C: Atom type of first atom before reaction. Atom types are used by LAMMPS to identify atoms of different elements, or more generally to identify atoms which have different force field parameters. For example, a carbon atom in a methane molecule and a carbon atom in an ethane molecule may have different force field parameters. In addition, atom types may be used to distinguish pseudo-atoms that have identical force field parameters, but undergo different reactions, which is often true in coarse-grained systems.

D: Atom type of second atom before reaction

E: Atom type of first atom after reaction

F: Atom type of second atom after reaction

G: The cut-off distance for reactions (r_i, Eq. 2.1). The units for r_i are either in Angstroms or reduced units, depending on the value of the LAMMPS parameter `units`.

H: The reaction probability coefficient (P_i, Eq. 2.1)

I: The period between reaction attempts in time units (t_i, Eq. 2.1)

Note that no reference is made to the type of bond that is formed in addition reactions or that is destroyed in scission reactions. Rather it is assumed that all bond types are determined by the identity of the atoms. The bond type is an index used to locate the force field parameters for the bond. The bond type corresponding to each pair of atoms types that will occur in the simulation is specified as follows:

```
bond type 2 2 1
bond type 2 3 2
:
```

where the three integer arguments are the first atom type, the second atom type and the bond type. If a bond occurs for which the bond type has not been specified in this manner, the simulation terminates with an error condition.

This information is sufficient to fully define how a particular chemistry is represented within a LAMMPS simulation. The implementation within LAMMPS is concentrated in two locations. Firstly, each time the neighbor lists for non-bonded and bonded interactions are constructed,

additional lists of all the reactive sites are constructed. In the case of scission reactions, this requires merely constructing a list of all the bonds whose atom types match those of a given scission reaction. In the case of addition and transfer reactions, for each atom matching the atom type of the first reactant, we construct a list of atoms of the second atom type that are within r_c+r_{skin}. This is done in a manner similar to that used for the neighbor lists themselves, it assumes that the reneighboring is performed using the binning method, and so it avoids the need to perform an $O(N^2)$ search over all atoms. Secondly, at each timestep, after the position update we check to see if any addition, chain transfer or scission reactions need to be performed by checking the values of t_a, t_t, and t_s against the current timestep. On those timesteps for which a given reaction type is scheduled, the lists constructed during the neighbor list update are consulted to identify reaction candidates. A successful reaction results in immediate update of the atom types and addition or removal of the bond from the bond list. In addition, the atoms must be deleted from the lists of reactions in which they occur. The atoms are also added to the lists of reactions corresponding to their new atom type. Atoms which belong to a different processor are loaded into a buffer. After all the reactions are complete, this buffer is communicated to neighboring processors which then update the atoms that they own. Then the list of reaction partners are constructed for all the new atoms in each reaction list. A call to the LAMMPS routine `communicate` is used to update neighboring processors of changes in atom types. Finally the bond types are update to match the new atom types, and the bond scission reaction lists are update accordingly. In order to avoid conflicts between processors, this sequence of events must be repeated eight times, once for each of the octants of the cubic domain belonging to the processor. Only atoms residing in the current octant are allowed to react. For processor domains bigger than the maximum separation of reactants, this avoids conflicts.

3. Reactive Molecular Dynamics: Example Applications

3.1 Linear Polymer Growth

Using an earlier version of the code described above, we simulated irreversible growth of linear polymers in a monomer liquid. Simulations of this type were previously published by Akkermans & Toxvaerd (Ref. 2). The system initially consists of a dense fluid of monomer species, with one or more reactive species. Reactive sites were allowed to bond *irreversibly* with neighboring monomers according to a Monte Carlo probability dependent on the probability coefficient P_i and the number of monomers within a collision diameter r_i, as defined in Eq. 2.1. This scheme does not require scission or chain transfer reactions, only addition.

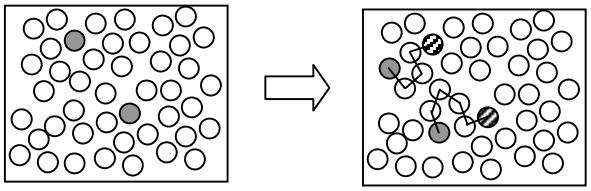

Figure 3.1 Schematic of the irreversible polymerization process. At each MD timestep, the reactive sites (gray) can bond with monomers (white).

Depending on the relative importance of diffusion and reaction limitation, the overall rate of polymerization and the conformational properties of the growing polymer vary. Toxvaerd *et al.* found that as the reaction time t was reduced, the actual rate of polymer growth did not increase proportionately. Rather the growth rate at a given polymer length approached a maximum diffusion limited value. However, they were unable to explore how this behavior scaled with polymer size, because they were limited to relatively small system sizes (10^4 particles). With this system size, they we unable to accurately study polymers longer than 50 monomers, due to the onset of interactions between the polymer and its periodic images. This causes the polymer to behave more like a semi-dilute system. The rate of polymer growth decreases and the polymer dimensions become more compressed. By developing a parallel simulation capability, we are able to simulate much larger systems with ease (e.g. 10^6 particles on only 64 processors of ASCI Red), allowing us grow polymers containing 1000 monomers without producing the finite-size artifacts referred to above.

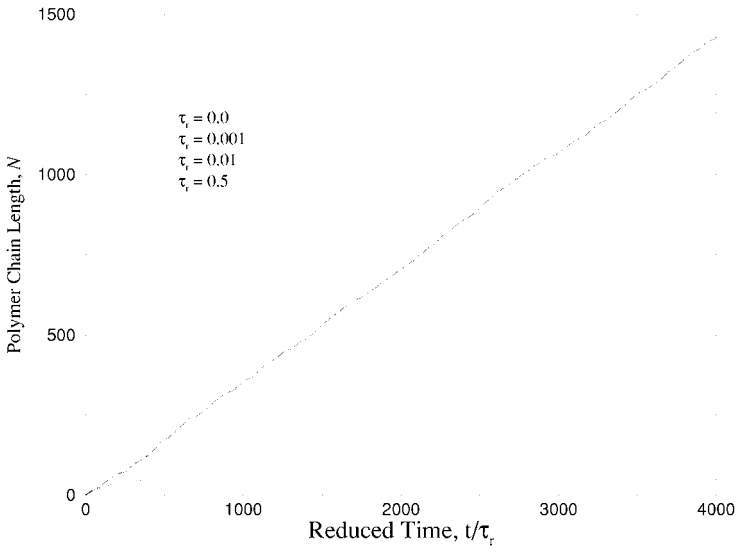

Figure 3.2 Polymer chain length versus time for several different reaction rates. The time axis is scaled by the reaction time constant, so that in the absence of diffusion limitation, all the curves would coincide with the top curve.

Figure 3.2 shows the average polymer chain length as a function of time for several different reaction rates (reported as a reaction time constant). As the rate of reaction is increased, the overall polymerization rate increases to a maximum limiting rate that is a function of diffusion or mass transfer resistance. Toxvaerd *et al* concluded that this mass transfer resistance was due to the reactive end of the polymer being obstructed by the rest of the polymer chain. In our simulations, direct observation of polymer configurations showed that while this form of trapping clearly occurs in 2-D, it is not seen in 3-D (see Fig. 3.3). This is not surprising, given the well-known added freedom of random walks in 3-D as compared to 2-D. One manifestation of this added freedom is given by the well-known theorem due to Polya, which states that in 2-D a random walk will always eventually return to it starting point, whereas in 3-D, it often never does so.

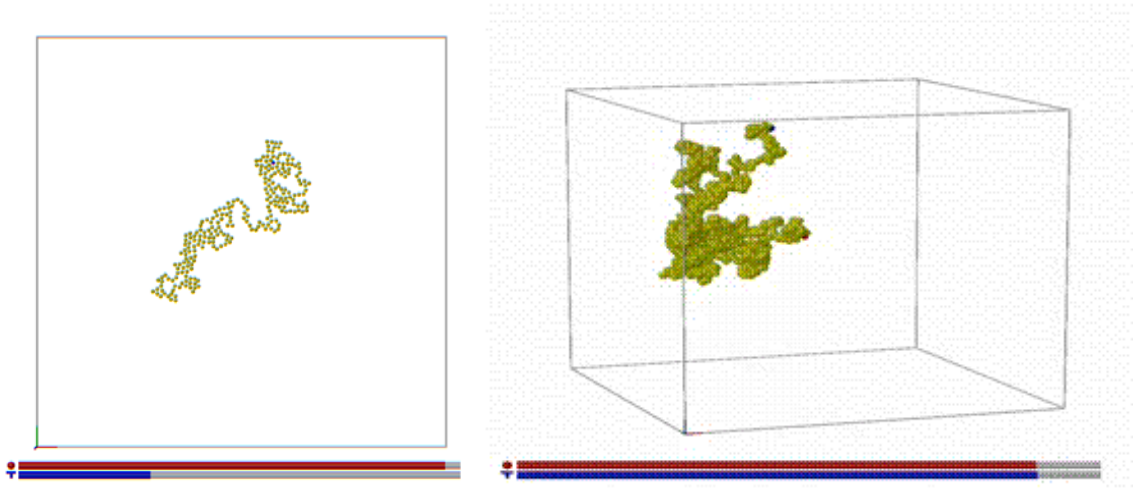

Figure 3.3 Snapshots taken from 2D (left) and 3D (right) simulations of linear irreversible polymerization. In the left image, the reactive chain end is clearly trapped within the coiled polymer. In the right image, no such trapping is observed.

In order to estimate the effect of trapping on the overall growth rate, two issues to be considered: the probability of a trap occurring and the time required to escape a trap. It is important to note that the polymer trapping of the type described here is not exactly described by Polya's theorem, for two reasons. Firstly, the polymer can form only a subset of all random walk conformations, since it can not overlap with itself. In the limit of infinitely slow reaction, the growing polymer is fully relaxed at all times. Hence the polymer conformational distribution is identical to that of the equilibrium polymer. Under the conditions we have studied, excluded volume interactions are dominant, and so the equilibrium conformational distribution of the relaxed polymer is in the same universality class as the self-avoiding walk (SAW). At the opposite extreme of infinitely fast reaction, the polymer grows on a set of particles that form a connected cluster. Growth terminates when the reactive site has no neighboring monomers within the reaction radius. Assuming that the connected cluster is infinite and non-fractal i.e. we are well above the percolation threshold, then the polymer grows in a manner similar to the Rosenbluth walk, also called the growing self-avoiding walk (GSAW). Lyklema and Kremer (Ref. 3) have used Monte Carlo simulation to study this class of random walk on 2-D and 3-D lattices. The GSAW samples exactly the same conformations as the SAW, but the growth process biases towards more compact conformations, whereas for the SAW, all non-overlapping conformations occur with equal probability. Despite this distinction, Lyklema has shown that in two dimensions the GSAW exhibits the same asymptotic scaling exponents as the SAW. In three dimensions, it was not possible to grow the chains long enough to reach the asymptotic scaling regime. Nonetheless, the trends in the effective exponents of finite length GSAW chains were consistent with the SAW universality class.

A second reason is that Polya's theorem addresses the probability of the walk visiting an occupied site. Trapping requires that the walk visit a site that is surrounded by occupied sites, a much more complex event. Pietronero (Ref. 4) has shown that while a GSAW is far less likely to get trapped in three dimensions than in two dimensions, it is still possible. In fact, preferential trapping of compact configurations is the reason why GSAWs are expected to exhibit the same asymptotic scaling behavior as SAWs. However, trapping events are so rare in three dimensions that the SAW scaling regime is expected to be reached only for walks longer than 10^6, and so has never been observed. For chains smaller than 1000, the growth process favors more compact conformations, and the scaling is similar to that of a theta polymer.

The time required to escape a trap has not been treated in the literature. We have attempted to directly measure this in our simulations. We measured the average duration of apparent trapping events i.e. average duration of periods when no monomers are within the reaction radius r_c. Figure 3.4 shows the average trapping time as a function of polymer length in two and three dimensions. Time is measured in the reduced Lennard-Jones units, which corresponds roughly to the average time required for a particle to traverse one particle diameter, ignoring collisions with other particles. In each case, we used simulation data obtained at several different reaction rates. Since trapping is a diffusion-limited phenomenon, we do not expect to see any dependence on reaction rate. In 2-D, the trapping time was quite large, and increased logarithmically with chain length, suggesting that trapping release is limited by relaxation of the polymer conformation. In 3-D the average trapping time was much shorter, about equal to the average time between collisions. Moreover, 3-D the trapping time was independent of chain length. This shows clearly that in 3-D, that trapping release is not controlled by polymer relaxation. Rather it is limited only by the much faster phenomenon of particle-particle collisions.

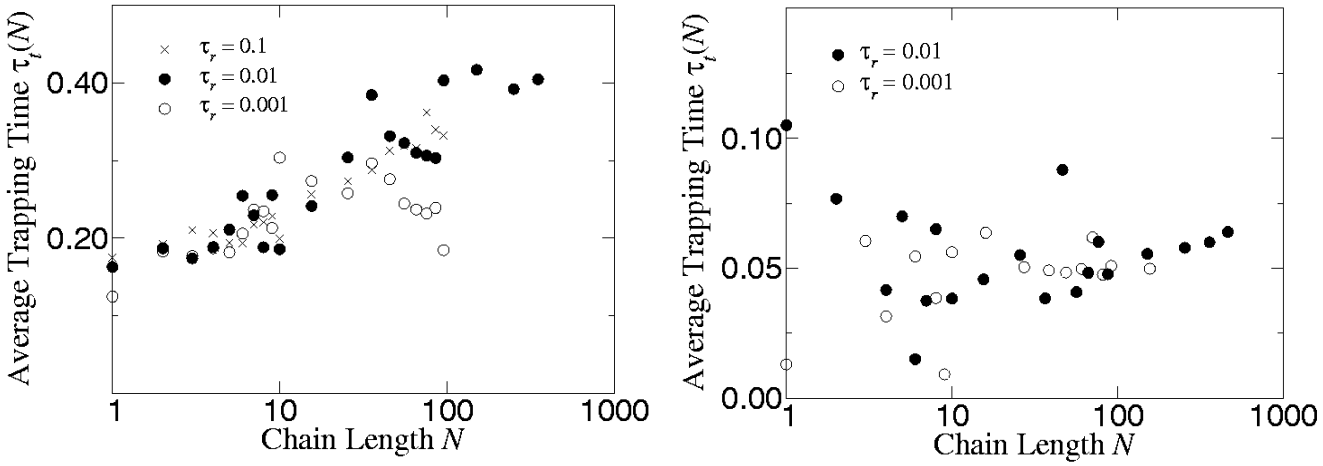

Figure 3.4 Average trapping time versus degree of polymerization in 2D (left) and 3D (right). In the 2D case, results are shown for three different reaction rates. In the 3D case, results are shown for two different reaction rates.

3.2 Thermal Degradation and Oxidation of Polymer Networks

Having developed the code to handle bond formation for the polymerization application described above, we then addressed the more challenging problem of oxidation of polymer networks. This effort was driven by the needs of the ASCI M&PM project "Degradation of Organic Materials". Oxygen-initiated breakdown of the polymer networks in o-ring materials, sometimes referred to as chemical aging, is a serious concern for the nuclear weapon stockpile. Butyl rubber O-rings prevent penetration of moisture into weapon compartments containing water-sensitive components. Hence the performance and reliability of o-ring is of critical importance to stockpile stewardship.

The process of chemical aging is understood in general terms. When an o-ring is deployed in the field, it is compressed between two surfaces. The compressive strain is resisted by the polymer network, which exerts a compressive stress on the surfaces. The presence of oxygen causes chemical reactions that result in scission of polymer chains in the network, which relieves a certain amount of stress. The broken chains often form new cross-links that do not carry any stress. Over time this process causes the o-ring to forget its original shape and adopt the shape of the confining volume, as illustrated in Figure 3.5. The right image is of an o-ring that was aged for fifteen years. The aged o-ring has clearly undergone chemical aging that caused its equilibrium shape to evolve towards its deformed shape in use.

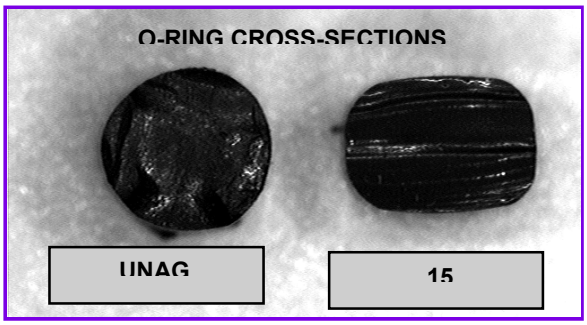

Figure 3.5 Photographs of a virgin and an aged o-ring cross-section.

The details of this process are not well understood. This is a problem, because evaluation of o-ring batches is performed using accelerated aging experiments designed to mimic the decades-long process of chemical aging in a short laboratory experiment. A better understanding of the relationship between aging experiments and actual behavior in the field would aid substantially in preventing deployment of poor-performing o-ring batches such as the one shown above.

The current development effort was motivated in part because direct investigation of chemical aging in a MD simulation could greatly extend the understanding of o-ring aging under laboratory and field conditions. This work is ongoing and is now being funded through the ASCI M&PM project cited above. The detailed mechanism of polymer oxidation is believed to be reasonably well represented by what is known as the Basic Auto-oxidation Scheme (BAS) (Ref. 5). BAS is essentially a free radical chain reaction process, which is accelerated by the presence of oxygen.

Thermal Degradation Reaction Scheme

- **Initiation (Chain Scission)**

$$-CH_2-CH_2- \rightarrow -CH_2\bullet + \bullet CH_2-$$

- **Chain Transfer**

$$-CH_2\bullet + -CH_2-CH_2- \rightarrow -CH_3 + -\bullet CH-CH_2-$$

- **Termination (Cross-Linking)**

$$\rightarrow -CH_2\bullet + -\bullet CH-CH_2- \rightarrow -CH_2-CH-CH_2-$$

Figure 3.6 Outline of the reaction chemistry for thermal degradation.

We have used a simpler phenomenon, thermal degradation, as a prototype for the BAS scheme, as it is simpler, but contains all the essential ingredients. The free radical chain reaction chemistry for thermal degradation involves addition, chain transfer and scission reactions (Figure 3.6). Each backbone carbon and its associated hydrogens are treated as a single pseudo-atom. Hence we have five normal pseudo-atom types corresponding to the methane molecule, methyl group, secondary, tertiary and quaternary carbon groups. In addition we need to have four free radical pseudo-atom types corresponding to the methylene radical and the primary, secondary, and tertiary free radical carbon groups. The quaternary carbon free-radical does not exist. The pseudo-atoms are all taken to interact according to Lennard-Jones potentials. The seven pseudo-atom types which have bonds can form a total of 7x8/2=28 different bond types. By using the same bond type for bonds involving the normal pseudo-atom and the corresponding free radicals, we can reduce the number of bond types to 4x5/2=10. The complete set of reactions involving these various atom types consists of 10 addition reactions, 16 chain transfer reactions and 10 scission reactions. The full listing of these reactions and the bond type definitions is given in Figure 3.7.

```
addit coeff  1  6 6 2 2  1.12 0.05  0.01     | bond type 2 2 1
addit coeff  2  6 7 2 3  1.12 0.05  0.01     | bond type 2 3 2
addit coeff  3  6 8 2 4  1.12 0.05  0.01     | bond type 2 4 3
addit coeff  4  6 9 2 5  1.12 0.05  0.01     | bond type 2 5 4
addit coeff  5  7 7 3 3  1.12 0.05  0.01     | bond type 3 3 5
addit coeff  6  7 8 3 4  1.12 0.05  0.01     | bond type 3 4 6
addit coeff  7  7 9 3 5  1.12 0.05  0.01     | bond type 3 5 7
addit coeff  8  8 8 4 4  1.12 0.05  0.01     | bond type 4 4 8
addit coeff  9  8 9 4 5  1.12 0.05  0.01     | bond type 4 5 9
addit coeff  10 9 9 5 5  1.12 0.05  0.01     | bond type 5 5 10

trans coeff  1 6 1 1 6  1.12 0.05  0.01      | bond type 2 7 1
trans coeff  2 6 2 1 7  1.12 0.05  0.01      | bond type 2 8 2
trans coeff  3 6 3 1 8  1.12 0.05  0.01      | bond type 2 9 3
trans coeff  4 6 4 1 9  1.12 0.05  0.01
trans coeff  5 7 1 2 6  1.12 0.05  0.01      | bond type 3 7 2
trans coeff  6 7 2 2 7  1.12 0.05  0.01      | bond type 3 8 5
trans coeff  7 7 3 2 8  1.12 0.05  0.01      | bond type 3 9 6
trans coeff  8 7 4 2 9  1.12 0.05  0.01
trans coeff  9 8 1 3 6  1.12 0.05  0.01      | bond type 4 7 3
trans coeff  10 8 2 3 7  1.12 0.05  0.01     | bond type 4 8 6
trans coeff  11 8 3 3 8  1.12 0.05  0.01     | bond type 4 9 8
trans coeff  12 8 4 3 9  1.12 0.05  0.01
trans coeff  13 9 1 4 6  1.12 0.05  0.01     | bond type 5 7 4
trans coeff  14 9 2 4 7  1.12 0.05  0.01     | bond type 5 8 7
trans coeff  15 9 3 4 8  1.12 0.05  0.01     | bond type 5 9 9
trans coeff  16 9 4 4 9  1.12 0.05  0.01

                                             | bond type 7 7 1
sciss coeff  1  2 2 6 6  1.12 0.05  0.01     | bond type 7 8 2
sciss coeff  2  2 3 6 7  1.12 0.05  0.01     | bond type 7 9 3
sciss coeff  3  2 4 6 8  1.12 0.05  0.01     | bond type 8 8 5
sciss coeff  4  2 5 6 9  1.12 0.05  0.01     | bond type 8 9 6
sciss coeff  5  3 3 7 7  1.12 0.05  0.01     | bond type 9 9 8
sciss coeff  6  3 4 7 8  1.12 0.05  0.01
sciss coeff  7  3 5 7 9  1.12 0.05  0.01
sciss coeff  8  4 4 8 8  1.12 0.05  0.01
sciss coeff  9  4 5 8 9  1.12 0.05  0.01
sciss coeff  10 5 5 9 9  1.12 0.05  0.01
```

Figure 3.7 Complete list of input commands required to specify the thermal degradation chemistry. To save space, the bond type commands are listed in the second column.

We tested the code and the reaction scheme on a simple example: a system that starts out as a dense fluid of methylene free radical pseudoatoms. We performed ten independent simulations with a timestep of 0.005 in reduced Lennard-Jones units, so the reactions were allowed to occur every second timestep. Each simulation started from a different initial configuration of atoms and used a different random number seed for performing the chemical reactions.

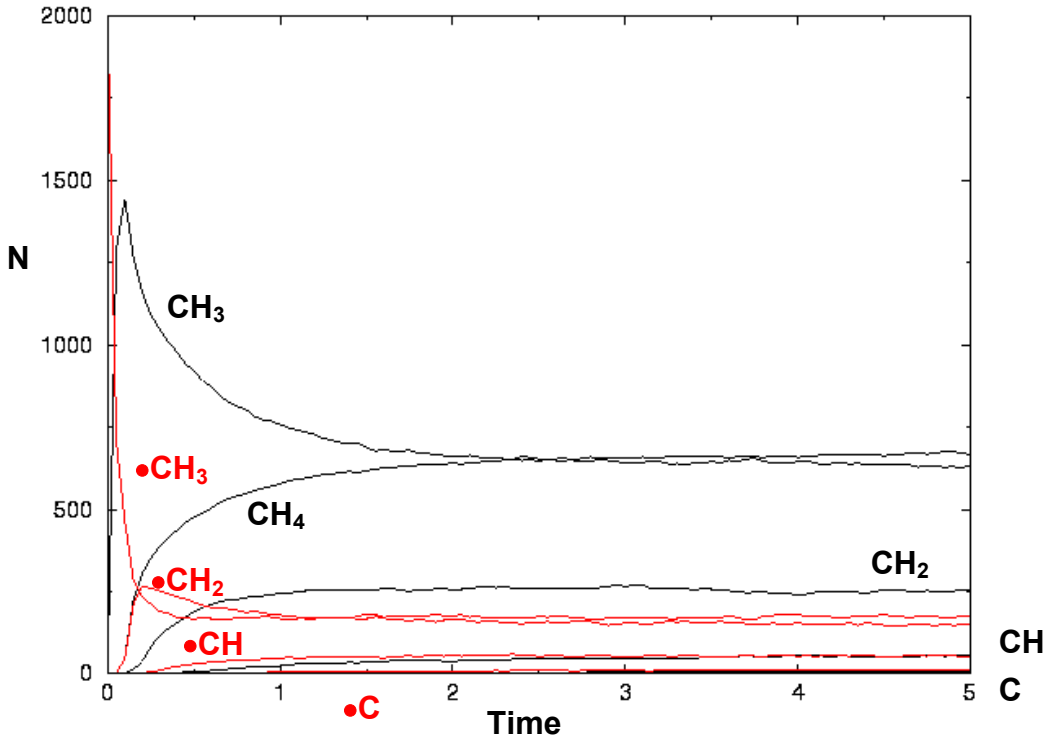

Figure 3.8 Evolution of a system undergoing thermal degradation chemistry, starting from a dense fluid of 2000 methylene radicals. The normal species are shown in black and the radical species in red.

Figure 3.8 shows a plot of the number of atoms of each species versus time, averaged over the ten independent simulations. This data was obtained on 8 nodes of CPlant. The normal pseudoatoms are colored in black and the free radical pseudoatoms are colored in red. At very short times, the concentration of methylene radicals decreases sharply, while the number of methyl groups increases at the same rate. This is due to the formation of dimers. Chain transfer and further bond formation causes the concentration of methyl groups to decrease again, and the concentration of higher coordination species increases. At a reduced time of 5, or 1,000 timesteps, all of the species concentrations have stopped changing. At this point, a dynamic equilibrium has been reached between the bond-making, chain transfer and scission reactions. Figure 3.9 is a blow up showing the concentration of the two most dilute species. These species are highly coordinated, and so are very sensitive to the details of the reactive dynamics scheme. In each case, the four graphs correspond to simulations on 1, 2, 4 and 8 nodes of Cplant. The results are in statistical agreement, as indicated by the error bars included in the plot. This is a good validation of the parallel algorithm, as even slight discrepancies in the treatment of reactions at the domain boundaries should show up as a the domain sizes are made smaller.

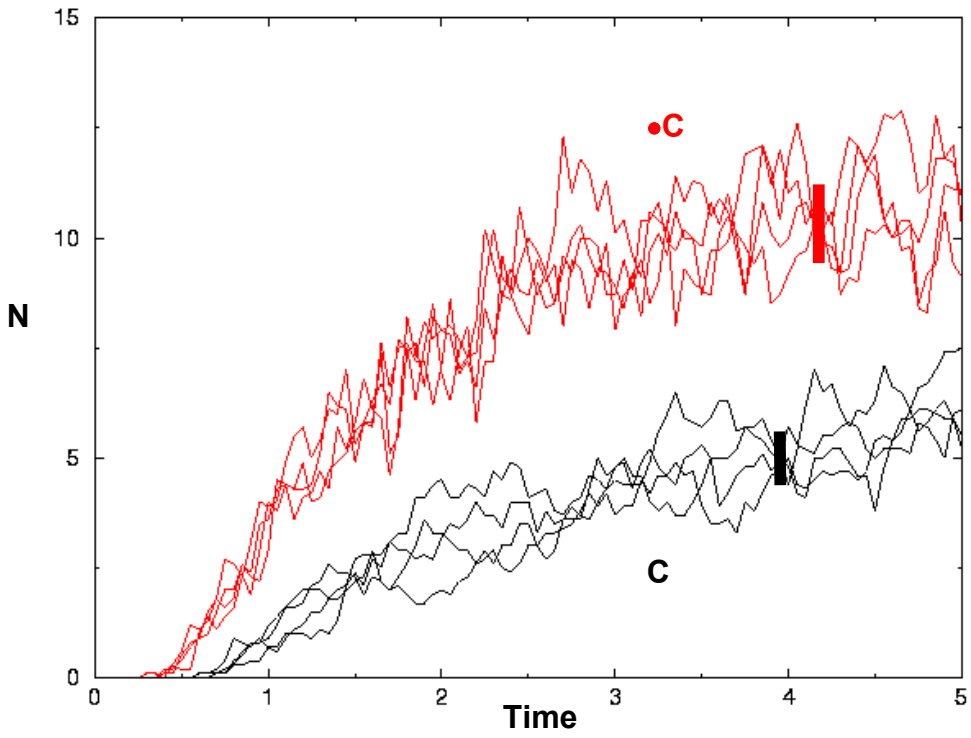

Figure 3.9 Blow-up of Figure 3.7, showing only the two most dilute species. In each cases the four curves shown are the results obtained running on 1, 2, 4 and 8 nodes of CPlant. The thick vertical lines indicate the size of the error bars (one standard deviation confidence interval).

4. Summary and Discussion

A general parallel code has been developed for implementing chemical reaction events in the LAMMPS molecular dynamics code. It has been successfully tested on two different reactive systems. The code can be run on an arbitrary number of processors, and the results are independent of the number of processors used. This has enabled simulation of very large systems using massively parallel computers such as Sandia's ASCI Red machine and the CPlant clusters. The code is currently being used to simulate chemical aging of organic polymers.

There current code has certain limitations, some of which will be addressed in the near future. It is based on an early version MPI version of LAMMPS (LAMMPS99). Hence, it does not have some of the more recent LAMMPS improvements, such as run-time memory allocation. The reaction update schemes do not extend to special atoms, bond angles, dihedral angles and multivalent bonds. The reaction scheme assumes that each bond is only associated with one atom. For parallel calculations this requires that the "Newton" option be used for bonded interactions. Finally, the parallel and serial efficiency of the code need to be improved, particularly in cases where the number of reactive sites is comparable to the number of atoms in the system.

5. References

1. S. J. Plimpton, R. Pollock, M. Stevens , "Particle-Mesh Ewald and rRESPA for Parallel Molecular Dynamics Simulations", *Proc of the Eighth SIAM Conference on Parallel Processing for Scientific Computing*, Minneapolis, MN, March 1997.

2. R. L. C. Akkermans, S. Toxvaerd, and W. J. Briels, “Molecular dynamics of polymer growth”, *J. Chem. Phys.*, **109** 2929 (1998)

3. J. W. Lyklema and K. Kremer "Monte Carlo series analysis of irreversible self-avoiding walks: The growing self-avoiding walk", *J. Phys. A: Math. Gen.* **19** 279 (1986)

4. L. Pietronero "Survival Probability for Kinetic Self-avoiding Walks", *Phys. Rev. Lett.* **55** 2025 (1985)

5. K. T. Gillen, J. Wise and R. l. Clough "General Solution for the Basic Auto-oxidation Scheme", *Polymer Degradation and Stability*, **47** 149 (1995)