35

MATLAB Programming of Polymerization Processes using Monte Carlo Techniques

Mamdouh A. Al-Harthi
*Chemical Engineering Department,
King Fahd University of Petroleum & Minerals, Dhahran 31261,
Saudi Arabia*

1. Introduction

The expression "Monte Carlo method" is actually very general. Monte Carlo (MC) methods are stochastic techniques - meaning they are based on the use of random numbers and probability statistics to investigate problems. MC methods are used from economics to nuclear physics to regulating the flow of traffic. But the way MC methods are used, varies from one application to another. A plethora of algorithms are available for modeling a wide range of problems. But, to call something a "Monte Carlo" experiment, all you need to do is use random numbers to examine the problem. The use of MC methods to model physical problems allows us to examine more complex systems. For example, solving equations which describe the interactions between two atoms is fairly simple; solving the same equations for hundreds or thousands of atoms is impossible. With MC methods, a large system can be sampled in a number of random configurations, and that data can be used to describe the system as a whole (*Notes from University of Nebraska-Lincoln Physical Chemistry Lab*).

Free radical polymerization is one of the most widely used polymerization technique in polymer industry till date. Even a basic course on polymer science / engineering would without doubt start from free radical polymerization. Since it a pivot around which both academics and industrialist work around, it becomes imperative to study this class of polymerization. In this article, we would be illustrating how to model free radical copolymerization using the MC method based on Gillespie's algorithm. Even though a wide range of scholarly articles have been published on MC simulation in the field of polymerization from the 90's till the recent past (He et al., 1997,2000, Tobita 2001, 2003, Lu et a., 1993, Al-harthi et al., 2006,2007), but none of them give out the coding aspects of this process. Al -harthi et al. is one of the pioneers in modeling atom transfer radical polymerization, which is a subset/ branch of the free radical polymerization. Recently Al -harthi et al., have published two well accepted works on atom transfer radical copolymerization (Al -harthi et al., 2009 a & b). Combining these two aspects that the actual programming aspects have not been brought out till now and the authorial's immense experience in this field. We strongly feel the importance of educating chemical engineering students and industrials in this aspect. Even though we could have easily illustrated the homopolymerization and left the copolymerization to the readers, we on the contrary would like to educate them on a problem which is definitely difficult and hence by understanding these concepts, they can go on solve higher complex problems. We have dealt with atom transfer radical copolymerization directly which in fact is one step ahead of free radical polymerization. Hence the reader can model both free radical and atom transfer radical copolymerization.

People might wonder why MC methods for modeling polymerization? Researchers have predominantly used population balances and method of moments for modeling polymerization. But its inability to predict the molecular weight distribution is a big drawback. MC method can easily predict the complete molecular weight distribution as shown in many referred journals (Al -harthi 2007). The ease of using MC method is another bigger advantage when compared to other numerical techniques. Hence it becomes necessary for students and researchers to understand this technique in a broader sense.

For the university student, this article provides a number of benefits:

- Understanding the Gillespie's Algorithm
- A easy to understand procedure explained in detail
- The latest concept of atom transfer radical copolymerization which is not a subject matter in most general polymer science textbooks.

For the Industrialist, the article also provides a number of benefits:

- Our modeling technique produces a highly accurate result and hence of industrial importance
- Results like Molecular weight distribution, sequence length distribution, Polydispersity Index (PDI), conversion and other results can be easily modeled and relied upon on.
- Most of all, the ease with which the programming can be done.

2. General Monte Carlo procedure

The dynamic Monte Carlo approach used in this article is based on the method proposed by Gillespie (Gillespie 1977). Suppose we consider the following basic free radical kinetic scheme:

Initiation: $$PR + M \xrightarrow{k_i} R_1 \bullet \tag{1}$$

Propagation: $$R_r \bullet + M \xrightarrow{k_p} R_{r+1} \bullet \tag{2}$$

Termination by Combination: $$R_r \bullet + R_m \bullet \xrightarrow{k_{tc}} P_{r+m} \tag{3}$$

Where M is the monomer, PR is the propagating radical during initiation, $Rr \bullet$ is polymer radical. P_r and P_{m} are the dead polymer chains. k_p is the propagation rate constant, k_{tc} is the rate constant of termination by combination, k_i is the initiation rate constant, and the subscripts r and m indicate the number of monomer molecules in the chain.

First the deterministic, or experimental, rate constant (k^{exp}) should be changed to stochastic, or Monte Carlo, rate constants (k^{MC}) according to the following equations

$$k^{MC} = k^{\exp} \text{ for first order reactions} \tag{4}$$

$$k^{MC} = \frac{k^{\exp}}{VN} \text{ for bimolecular reactions between different species} \tag{5}$$

$$k^{MC} = \frac{2k^{\exp}}{VN} \text{ for bimolecular reactions between similar species} \tag{6}$$

Secondly all concentrations should be transformed to number of molecules in the control (simulation) volume *V;* in our example we have only monomer concentration, consequently:

$$X_m = [M]\ NV \tag{7}$$

Where *N* is the avagadro's number
Then calculate the reaction rate for every reaction according to the equations:
Rate of initiation:

$$Ri = k_i^{MC}\, X_{pr}\ \ X_m \tag{8}$$

Rate of propagation:

$$R_p = k_p{}^{MC}\, X_r\, X_m \tag{9}$$

Rate of termination by combination:

$$R_{tc} = \frac{k_{tc}{}^{MC} X_r (X_r - 1)}{4} \tag{10}$$

Where X_r and X_m are the number of polymer radicals and monomer molecules respectively.
The total reaction rate (R_{sum}) is then calculated as the summation of the individual reaction rates.
Then the probability of any reaction (P_v) taking place at a given time is calculated by the following equation

$$P_v \ = \ \frac{R_v}{R_{sum}} \tag{11}$$

Then the following relation is used to determine which reaction type will take place at a given polymerization time

$$\sum_{v=1}^{\mu-1} P_v < r_1 < \sum_{v=1}^{\mu} P_v \tag{12}$$

where μ is the number of the selected reaction type and r_1 is a random number uniformly distributed between [0, 1]. Another random number is generated to determine the time interval (τ) between two consecutive reactions. The time step is related to the inverse of total stochastic rates and the natural logarithmic of r_2 according to the equation:

$$\tau = \frac{1}{\sum_{v=1}^{N} R_v} \ln\left(\frac{1}{r_2}\right) \tag{13}$$

The algorithm for a general Monte Carlo simulation is of the following steps:

1. Input the deterministic reaction rate constant, $k_1^{\exp}$, $k_2^{\exp}$, $k_3^{\exp}$,…$k_\mu^{\exp}$, simulation volume,*V*, avagadro's number, *N*, reactant concentration (mol/Volume)
2. Set time to zero and conversion to zero , *t=0 and x=0*
3. Calculate the stochastic rate constants, k_1^{MC}, k_2^{MC}, ……. k_μ^{MC} using equation 4, 5 or 6
4. Calculate and store the rates of reaction R_1, R_2 ….. R_μ for the selected reaction mechanism

5. Calculate and store the sum of the rates of reaction, R_{sum} according to the following equation: $R_{sum} = \sum_{\mu=1}^{M} R_{\mu}$ where R_{μ} is the rate of the μ^{th} reaction and M is the number of reactions.
6. Get two random numbers, r_1 and r_2, uniformly generated between 0 and 1, and calculate μ and τ according equations 12 and 13.
7. Select which reaction will occur according to equation 12
8. Update the number of molecules of each type in the reactor
9. Update the simulation time, t , by t = t + τ , and calculate the conversion, x
10. Return to step 4 until we obtain the desired polymerization time $t = t_{final}$ or final conversion $x = x_{final}$

The sequence of the steps could be altered or changed depending on convenience, flexibility and programming skills of the programmer.

3. Detailed Monte Carlo procedure and MATLAB programming:

Before moving to the modeling aspect, a brief introduction of atom transfer radical polymerization would give a better understanding and knowledge on this much researched topic. It was in the year 1995, discoveries on this process was made by Matyjaszeweski's group and Sawamoto's group. Till date several reviews, books, and book chapters have summarize hundreds of papers that appeared in the literature on ATRP of a large variety of monomers (Matyjaszeweski 2006). ATRP can synthesize various polymers with controlled molecular weight and narrow MWD. It can be carried out in a wide range of polymerization temperatures and is not very sensitive to the presence of oxygen and other inhibitors.

Figure 1 shows the general mechanism of ATRP. In addition to the monomer, the ATRP system consists of an initiator that has an easily transferable halide atom (RX) and a catalyst.

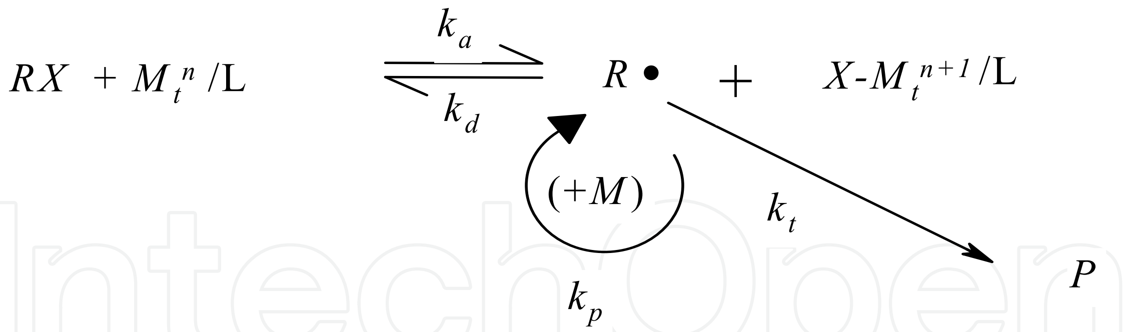


Fig. 1. ATRP Mechanism. RX: dormant species (alkyl halide); M tn /L : activator (metal complex); R• : propagating radical;X-Mtn+1/L : deactivator; M: monomer; P: dead chain.

The catalyst (or activator) is a lower oxidation state metal halide (MtnX) with a suitable ligand (L). Polymerization starts when the halide atom transfers from the initiator to the catalyst to form a free radical and a higher oxidation state metal halide Mtn+1-X (deactivator). This step is called activation or forward reaction. The deactivation step or backward reaction pushes the reaction to form dormant species (RX) rather than the radical's. The reaction of monomer molecules (M) in the propagation step is similar to conventional free radical polymerization.

For the modeling, we consider the following mechanism for atom transfer radical copolymerization. By removing equation 14 and 15 and by adding the dissociation of the initiator to give the free radicals would transform the process to free radical polymerization. Hence without any doubt we can apply our procedure to both free radical copolymerization and atom transfer radical copolymerization. Even though a general mechanism of Monte Carlo approach has been given before, a detailed description of how Monte Carlo simulations were carried for a specific polymerization process like atom transfer radical copolymerization would be interesting and will help the reader understand more about this process.

Initiation steps:

$$I + C \xrightarrow{k_a} PR + CX \tag{14}$$

$$PR + CX \xrightarrow{k_d} I + C \tag{15}$$

$$PR + M_A \xrightarrow{k_{iAA}} R_{1,A} \tag{16}$$

$$PR + M_B \xrightarrow{k_{iBB}} R_{1,B} \tag{17}$$

Equilibrium and propagation steps:

$$D_{r,A} + C \xrightarrow{k_{a,A}} R_{r,A} \bullet + CX \tag{18}$$

$$D_{r,B} + C \xrightarrow{k_{a,B}} R_{r,B} \bullet + CX \tag{19}$$

$$R_{r,A} \bullet + M_A \xrightarrow{k_{p,AA}} R_{r+1,A} \bullet \tag{20}$$

$$R_{r,B} \bullet + M_B \xrightarrow{k_{p,BB}} R_{r+1,B} \bullet \tag{21}$$

$$R_{r,A} \bullet + M_B \xrightarrow{k_{p,AB}} R_{r+1,B} \bullet \tag{22}$$

$$R_{r,B} \bullet + M_A \xrightarrow{k_{p,BA}} R_{r+1,A} \bullet \tag{23}$$

$$R_{r,A} \bullet + CX \xrightarrow{k_{d,A}} D_{r,A} + C \tag{24}$$

$$R_{r,B} \bullet + CX \xrightarrow{k_{d,B}} D_{r,B} + C \tag{25}$$

Transfer to monomer steps:

$$R_{r,A} \bullet + M_A \xrightarrow{k_{tr,AA}} P_r + R \bullet \tag{26}$$

$$R_{r,A} \bullet + M_B \xrightarrow{k_{tr,AB}} P_r + R \bullet \tag{27}$$

$$R_{r,B}\bullet + M_B \xrightarrow{k_{tr,BB}} P_r + R\bullet \tag{28}$$

$$R_{r,B}\bullet + M_A \xrightarrow{k_{tr,BA}} P_r + R\bullet \tag{29}$$

Termination by combination:

$$R_{r,A}\bullet + R_{m,A}\bullet \xrightarrow{k_{tc,AA}} P_{r+m} \tag{30}$$

$$R_{r,A}\bullet + R_{m,B}\bullet \xrightarrow{k_{tc,AB}} P_{r+m} \tag{31}$$

$$R_{r,B}\bullet + R_{m,B}\bullet \xrightarrow{k_{tc,BB}} P_{r+m} \tag{32}$$

Termination by disproportionation:

$$R_{r,A}\bullet + R_{m,A}\bullet \xrightarrow{k_{td,AA}} P_r + P_m \tag{33}$$

$$R_{r,A}\bullet + R_{m,B}\bullet \xrightarrow{k_{td,AB}} P_r + P_m \tag{34}$$

$$R_{r,B}\bullet + R_{m,B}\bullet \xrightarrow{k_{td,BB}} P_r + P_m \tag{35}$$

In Equations (14) to (35), I is the initiator, C and CX are the catalyst in its low and high valence states, M_A and M_B are the comonomers, $R_{r,A}\bullet$ and $R_{r,B}\bullet$ are polymer radicals terminated in monomer A and B, P_r is a dead polymer chain, D_r is a dormant polymer chain, k_i is the initiation rate constant, k_a is the activation rate constant, k_d is the deactivation rate constant, k_p is the propagation rate constant, k_{tc} is the rate constant of termination by combination, k_{td} is the rate constant of termination by disproportionation, k_{tr} is the transfer rate constant, and the subscripts r and m indicate the number of monomer molecules in the chain. The subscript A denotes that the chain ends with monomer A and the subscript B has an equivalent meaning.

The following assumptions and hypothesis are made in this mechanism:

1. All reactions are irreversible
2. All reactions are elementary
3. Rate Constants are chain length independent
4. Initiator and catalyst efficiencies are constant
5. Thermal initiation does not occur

Step 1:

As stated before we input the following constant:

- Temperature
- Concentration of Monomer A (*CMA*) and Monomer B (*CMB*)
- Avogadro's Number (*N*)
- Experimental Rate Constants (k_1^{exp}, k_2^{exp}, k_3^{exp} ,…k_μ^{exp})
- Reactivity ratios r_1 and r_2

Step 2:

During the process of polymerization in ATRcP, five different species are formed within the reactor namely:

- Dormant chain with end group corresponding to monomer A ($D_{r,A}$)
- Dormant chain with end group corresponding to monomer B ($D_{r,B}$)
- Growing Polymer Radicals with end group corresponding to monomer A ($R_{r,A}$)
- Growing Polymer Radicals with end group corresponding to monomer B ($R_{r,B}$)
- Dead Polymer chains (P)

MATLAB heavily relies on the creation of matrices in order to transform physical processes and to simulate such processes. A normal matrix represented by [ ] needs to have equal number of rows when constructing a column matrix.

Example: $A = \begin{bmatrix} 1\,0\,0\,1 \\ 1\;1\;1\,1 \\ 1\,0\,1\,1 \end{bmatrix}$ is the correct representation for a normal matrix

But in the case of polymerization we have thousands of unequal polymer chains and hence to represent such a scenario, a special type of matrix called cell matrix represented by { } have been utilized in our case.

Example: $B = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right\}$ is a special type of a collection of independent matrices.

Such a matrix allows us to know the following vital information:

1. Total number of chains, corresponding to 4 in example B
2. Chain length of independent chains - corresponding to (5,3,8,2) from example B
3. Sequence Length Distribution

   Consider chain 3(the longest chain), the program can access each element within the chain. If we assign 1 for monomer A and 0 for monomer B, the program scans for the sequence and gives us the following diads.

   [10]-AB, [00]-BB, [01]-BA, [01]-BA. Thus by assigning 0 and 1 for monomer A and B, we can easily know the diads and triads since we are able to access every element in every chain.

   The sequence can be obtained using the following syntax in MATLAB:

$$\texttt{Diads(0,0) =length(findstr(}\underbrace{\texttt{CELLMATRIXCONVERTEDTONORMALMATRIX}}_{\text{A ``cell2mat'' command and assigned to a variable ``vec''}}\texttt{,[0,0]))}$$

Note that you have to convert the cell matrix to a normal matrix and assign it another variable. We would see how to perform this operation and what is the term “vec” in detail in the coming sections. Figure 1 shows the diad prediction from our simulations. We have chosen the Styrene Methyl methacrylate(St-MMA) copolymer due to its applications as commodity plastics and also as engineering polymer. We have discussed the entire simulation results and tabulated the rate constants used, in our technical paper (Al-harthi et al 2009 a, b).

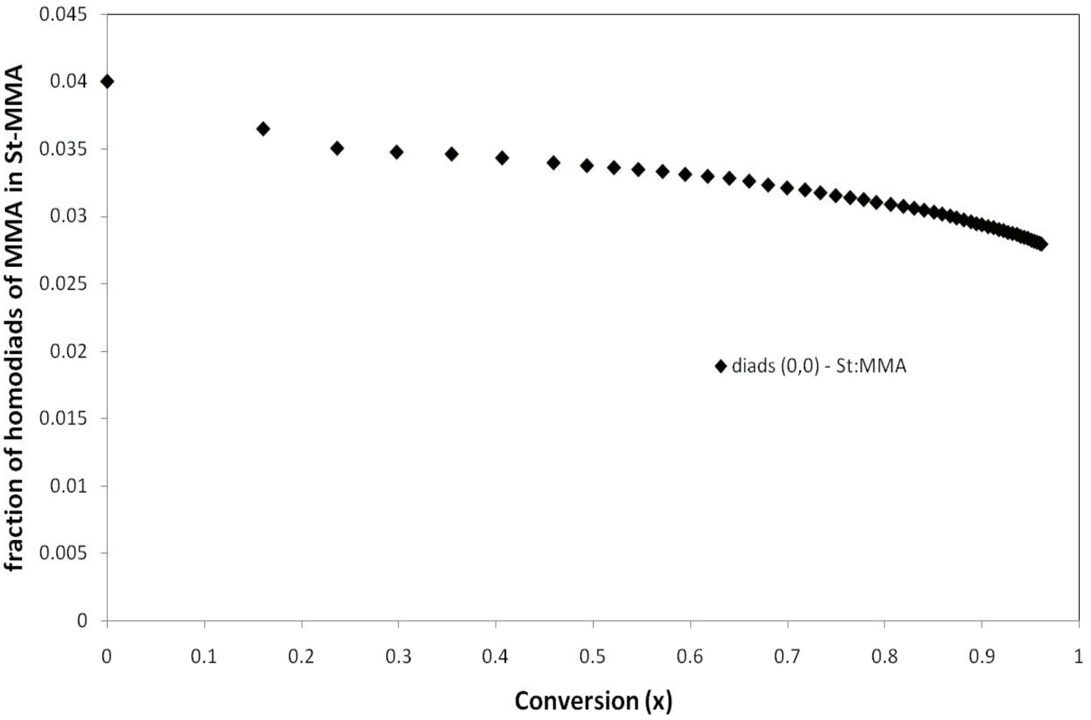


Fig. 1. Monte Carlo simulation results for the copolymerization of styrene and methyl methacrylate: fraction of homodiad of MMA as a function of conversion. The initial comonomer molar fractions in the reactor were $f_{0,St}$ = 0.75, $f_{0,MMA}$ = 0.25. The rate constants and other parameters are tabulated in our journal paper (Al-harthi 2009a)

4. Molecular Weight Distribution(MWD)
   Chain lengths are obtained from individual chains. From the chain length, the molecular weight could be obtained by knowing the physical properties of the monomers. Since individual chain lengths and their corresponding molecular weights are obtained, the MWD could be generated using Monte Carlo simulation.

Another example is Figure 2 which shows the strength of Monte Carlo simulations to predict the molecular weight distribution.

5. The type of end group in the chain (monomer A ended or monomer B)

Thus we allocate four cell matrices to $D_{r,A}$, $D_{r,B}$, $R_{r,A}$ and $R_{r,B}$ to the following cell matrices Da, Db, Ra and Rb respectively. The dead polymer chains are denoted by a normal matrix P. The dead polymers chains are classified as a normal matrix since only addition of terminated chains are involved without any further operations on them.

Step 3:

The number of monomer molecules, catalyst and initiator are calculated using the following relationship.

$$X_A = [M_A]NV$$

$$X_B = [M_B]NV$$

$$X_i = [I]NV$$

$$X_C = [C]NV$$

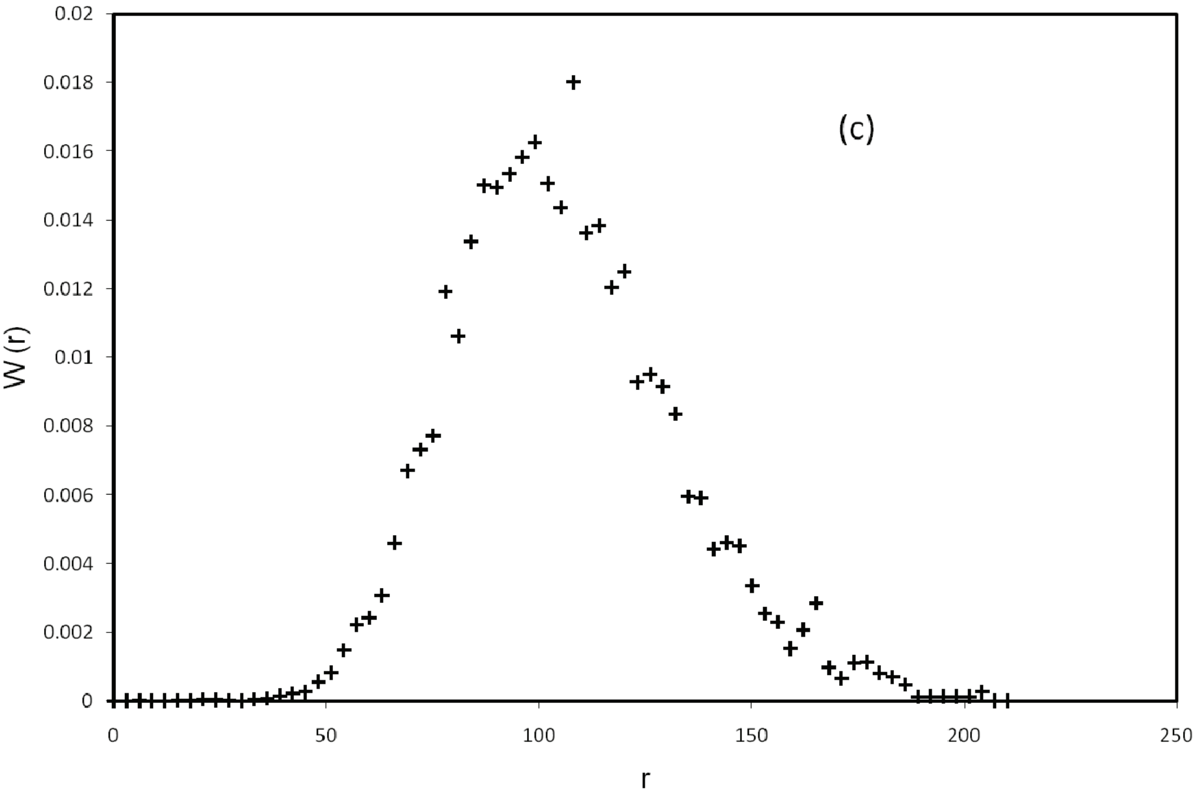


Fig. 2. Molecular weight distribution of styrene methyl methacrylate copolymers at a conversion of 0.99. The initial comonomer molar fractions in the reactor were $f_{0,St}$ = 0.5, $f_{0,MMA}$ = 0.5. The rate constants and other parameters are tabulated in our journal paper (Al-harthi 2009a)

The objective is to eliminate the volume in the concentration and find the number of molecules of monomer, catalyst and initiator.
The experimental rates are also converted into stochastic rates by equations 4, 5 and 6.

Step 4:

The rate of the reaction is calculated and is explained by the following examples.
Example 1: Initiation
Consider the following initiation reaction (equation 14)

$$I + C \xrightarrow{k_a} PR + CX \text{ ,}$$

The rate of the reaction would be calculated as

$$\text{R (1)} = k_{a1}{}^{MC} X_C X_i$$

Where $k_{a1}{}^{MC}$ is the stochastic rate constant of activation for monomer A.
Example 2: Equilibrium
Consider the following equilibrium reaction (equation 18)

$$D_{r,A} + C \xrightarrow{k_{a,A}} R_{r,A} \bullet + CX$$

The rate of the reaction would be calculated as

$$\text{R (5)} = k_{a1}^{\ MC} X_{da} X_C$$

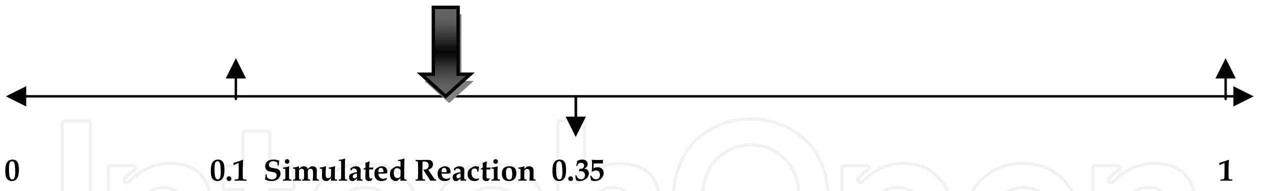
Thus all the rate of reaction are calculated as a product of their respected stochastic rate and number of molecules of either dormant chains, catalyst, monomers or growing polymer radical chains

Step 5:

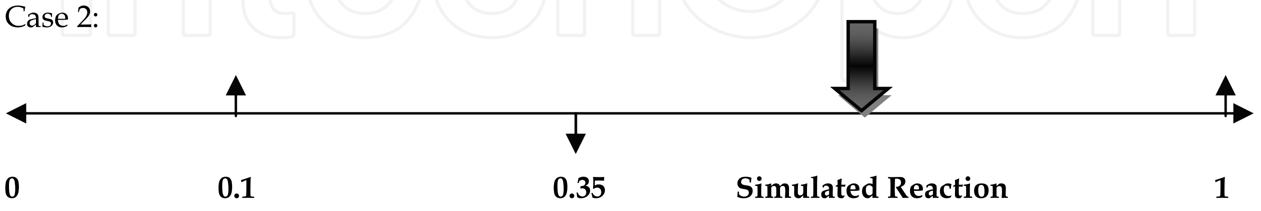
The next step is to calculate the total rate and probability of each reaction. Summation of all the rates gives us the total rate of reaction. The probability of a particular reaction to take place is calculated from equation 2.13
Example 3:

$$probability\ of\ activation\ reaction = \frac{Rate\ of\ activation\ reaction}{Total\ rate\ of\ reaction\ (R_{sum})}$$

While the probability of the specific reaction type is obtained by generating a random number uniformly distributed between 0 and 1. Using the random number and the probability of individual reactions, a suitable reaction is chosen. This is illustrated with an example below
Example 4:
A random number is generated in MATLAB by the command rand (r_1 = rand) results in an output of 0.25 (Case 1). Consider that individual probabilities of three reactions are P1 =0.1, P2=0.25, P3=0.65. The program would choose the reaction having the probability of 0.25 (P2 in this case which refers R (2) for the reaction type). Consider another case where we again generate a random number and we get an output of 0.5 (Case 2). In this case the program will pick a value which is above P2 which is P3. Thus the program chooses which reaction type to take place at every loop.
Case 1:



Case 2:



Step 6:

Calculation of time
Another random number (r_2) is chosen between 0 and 1 and used to calculate the time step by the following equation

$$\tau = \frac{1}{R_{sum}} \ln \left(\frac{1}{r_2} \right)$$

The time (t) is updated at every loop with the following relation:

$$t = t + \tau$$

Step 7:

Simulation of selected reaction

Once a particular reaction is chosen randomly as is in the case of a polymerization reactor, the next step is in carrying out the selected reaction by using the language of MATLAB. Many examples are given in order to clearly illustrate the procedure adopted while handling different type of reactions like initiation, equilibrium, propagation, cross propagation, transfer, cross transfer and bimolecular termination.

Example 5: Initiation

Consider the following reaction

$$I + C \xrightarrow{k_a} PR + CX$$

In this reaction, the initiator and catalyst lose molecules while an increase in the propagating radical and the deactivator/catalyst in the lower valence take is observed. Since Monte Carlo simulation works at the molecular level, we increase and decrease the number of molecules according to the reaction. Thus in this type of reaction we perform the following operation

- Decrease the number of Initiator molecules by 1: $X_i = X_i - 1$
- Decrease the number of Catalyst molecules by 1: $X_c = X_c - 1$
- Increase the number of the PR molecules by 1: $X_{pr} = X_{pr} + 1$
- Increase the number of CX molecules by 1: $X_{cx} = X_{cx} + 1$

Example 6: Equilibrium

$$D_{r,A} + C \xrightarrow{k_{a,A}} R_{r,A} \bullet + CX$$

As seen from the reation, there is a decrease in the catalyst and dormant chains while an increase in polymer radical and the lower valence catalyst. We should note that the dormant and the growing polymer radicals are in form of chains and we have represented them as cell matrices earlier. Hence a more sophisticated approach has to be adopted than the one adopted during initiation.

The following operations are performed:

- Decrease the number of catalyst molecules by 1: $X_c = X_c - 1$
- We need to add one molecule to X_{ra} (number of growing radical chain) and reduce one molecule from X_{da}(number of dormant chains) .
  $X_{ra} = X_{ra} +1;$
  $X_{da} = X_{da} - 1;$
- We need to randomly select one dormant chain from the cell matrix Da and transfer it to the cell Ra. To perform this physical phenomenon in MATLAB, we use the following procedure.
  - Find the total number of chains of the Dormant Radicals :
    ```
    >> LEN = length (Da)
    ```

- Randomly choose one chain among the available chains:
  >> rvec = randsample(LEN,1)
- The randomly chosen chain (rvec) would be placed in the growing polymer radical as follows:
  >> Ra{Xra} = Da{rvec}
- This ensures that the chain is placed in the new location created before and hence avoiding overlaying of chains.

- Finally we need to remove the randomly chosen chain (rvec) from Da since it has been placed in Ra. The following command removes the chosen chain from Da.
  >> Da(rvec) = [ ]

Example 7: Propagation

$$R_{r,A}\bullet + M_A \xrightarrow{k_{p,AA}} R_{r+1,A}\bullet$$

The following example shows how propagation reactions are handled in the program. We have a decrease in monomer A and an increase in the chain length of Ra. Hence the following operations are performed.

- Decrease in the number of monomer A by 1 : $X_A = X_A - 1$
- The physical phenomenon is the addition of monomer of the same kind (Monomer A to Radical A) to a particular chain in Ra which increases the chain length by adding one Monomer A unit. To incorporate this physical phenomena in the program the following operation was done:
  - Find the total number of Radical chains :
    >> LEN = length (Ra)
  - Randomly choose one chain :
    >> rvec = randsample (LEN,1)
    Convert the obtained chain which is a cell matrix type to a normal matrix in order to perform further operation on it since a cell matrix has access to limited basic operation. Hence the need arises to convert it to a normal matrix at certain junctures.
    >> vec = cell2mat (Ra (rvec))
    is the command for such an operation
  - Adding the monomer to the particular chain:
    >> lvec = length (vec) +1
  - Specifying the type of monomer added (Whether monomer A or B). Monomer A has a indexing of "1" and monomer B has an indexing of "0" in the program. Thus the following assigns that the monomer added is of type A.
    >> vec (lvec) =1
    performs the indexing operation.
  - Finally we need to have the chain in the cell matrix form due to aforementioned advantages of using a cell matrix. Hence the final part of the program is to convert the normal matrix to a cell matrix and place it in Ra
    >> Ra(rvec) = vec

Example 8: Cross propagation

$$R_{r,A}\bullet + M_B \xrightarrow{k_{p,AB}} R_{r+1,B}\bullet$$

The programming of cross propagation is slightly different from the previous section since two dissimilar radicals are involved. As emphasized before here a decrease in monomer B and Ra is seen while an increase is observed in Rb.

- Decrease in the number of monomer B by 1 : $X_B = X_B - 1$
- Increase in the number of growing polymer radical (Rb) : $X_{rb} = X_{rb} + 1$
- The physical phenomenon is the addition of monomer of the different kind (Monomer B to Radical A) while creating an increase in the chain length of Rb by adding one Monomer B unit. To incorporate this physical phenomena in the program the following operation was done:
  - Find the total number of Radical chains :
    >> LEN = length (Ra)
  - Randomly choose one chain :
    >> rvec = randsample (LEN,1)
    Convert the obtained chain which is a cell matrix type to a normal matrix in order to perform further operation on it since a cell matrix has access to limited basic operation. Hence the need arises to convert it to a normal matrix at certain junctures.
    >> vec = cell2mat (Ra (rvec))
    is the command for such an operation
  - Adding the monomer to the particular chain:
    >> lvec = length (vec) +1
  - Specifying the type of monomer added (Whether monomer A or B). Monomer A has a indexing of "1" and monomer B has an indexing of "0" in the program. Thus the following assigns that the monomer added is of type A.
    >> vec (lvec) = 0 performs the indexing operation.
  - Finally we need to have the chain in the cell matrix form due to aforementioned advantages of using a cell matrix. Hence the final part of the program is to convert the normal matrix to a cell matrix and place it in Ra
    >> Rb (xrb) = vec
- Finally the chain which was taken from Ra has to be removed since it has been transferred to Rb.
  Xra = Xra – 1
  >> Ra (rvec) = [ ] removes the chain which had been transferred to Rb.

The technique for handling transfer and cross transfer reactions are similar to the propagation reactions and hence will not be discussed.

A few of the results at this stage are given which could be used as a starting point and validation check for the results obtained using our methodology. Conversion is very well related to the propagation reaction and hence we would like to show a sample result for the linear relation obtained between conversion and the average chain length. Figure 3 shows the linear relationship between the average chain length and conversion.

Example 9: Bimolecular termination

$$R_{r,A}\bullet + R_{m,A}\bullet \xrightarrow{k_{tc,AA}} P_{r+m}$$

Termination is the process by which two growing polymer radical chains combine and form a dead polymer. The programming is done in the following manner

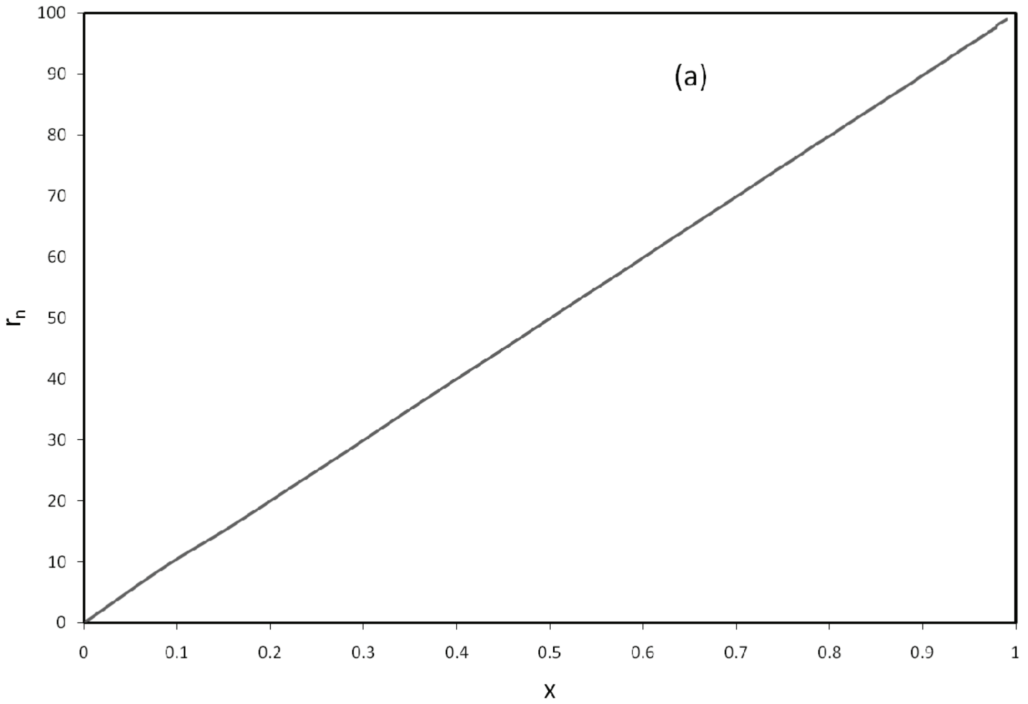


Fig. 3. Monte Carlo simulation results for the copolymerization of styrene and methyl methacrylate: average chain length as a function of conversion. The initial comonomer molar fractions in the reactor were $f_{0,St}$ = 0.5, $f_{0,MMA}$ = 0.5. The rate constants and other parameters are tabulated in our journal paper (Al-harthi 2009a)

- According to the equation, two chains from Ra will combine to form dead polymer and hence Ra should have a minimum of two chains. This check is done as follows
  >> LEN = length (Ra) and only if LEN > 2 , the reaction would proceed to completion.
- Two random chains are chosen from Ra as follows:
  >> rvec1 = randsample (LEN, 1)
  >> rvec2 = randsample (LEN, 1)
- The following chains are in the cell matrix form and hence need to be converted to a normal matrix in order to perform further operation on them, hence the following code is utilized:
  >> vec1=cell2mat(Ra(rvec1))
  >> vec2=cell2mat(Ra(rvec2))
- The number or the length of each chain is calculated as follows:
  >> lvec1=length(vec1)
  >> lvec2=length(vec2)
- We have the formation of a dead polymer chain and hence an increase in the number of dead polymer takes place as follows:
  $X_p = X_p + 1$
- Now the summation of the two growing radical chains takes place becoming a dead polymer as follows
  >> P(xp)= lvec1+lvec2
- Finally the growing chains which formed the dead polymer should be removed from Ra as follows;
  X_{ra}=X_{ra}-2;
  >> Ra{rvec1}=[ ];
  >> Ra{rvec2}=[ ];

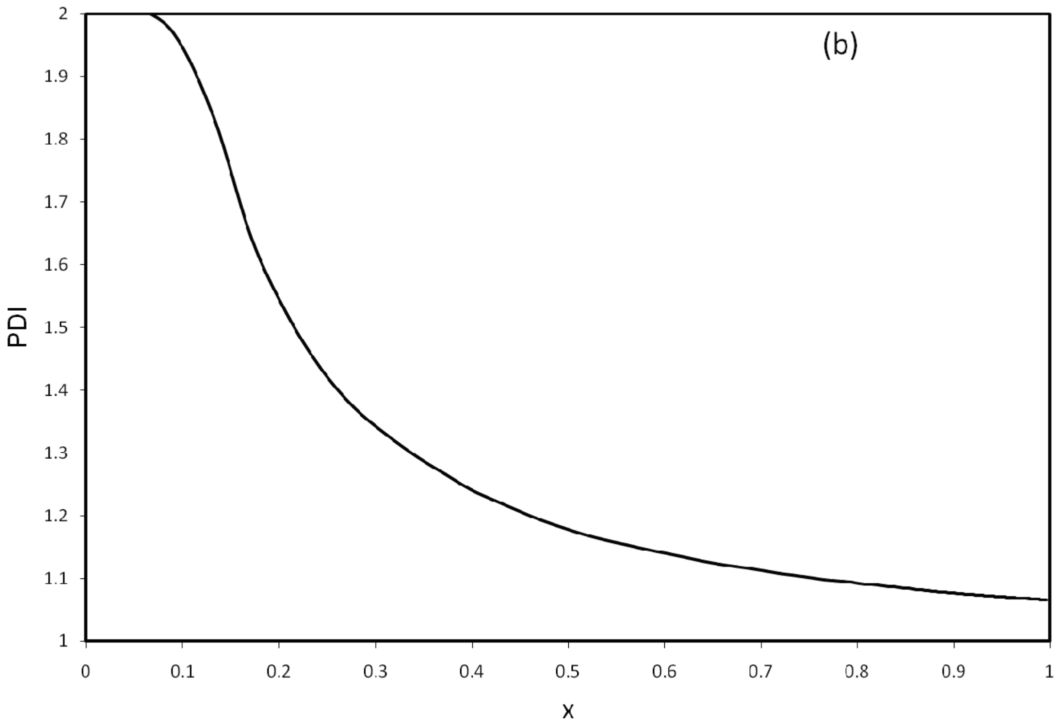


Fig. 4. Monte Carlo simulation results for the copolymerization of styrene and methyl methacrylate: PDI as a function of conversion. The initial comonomer molar fractions in the reactor were $f_{0,St}$ = 0.75, $f_{0,MMA}$ = 0.25. The rate constants and other parameters are tabulated in our journal paper (Al-harthi 2009a)

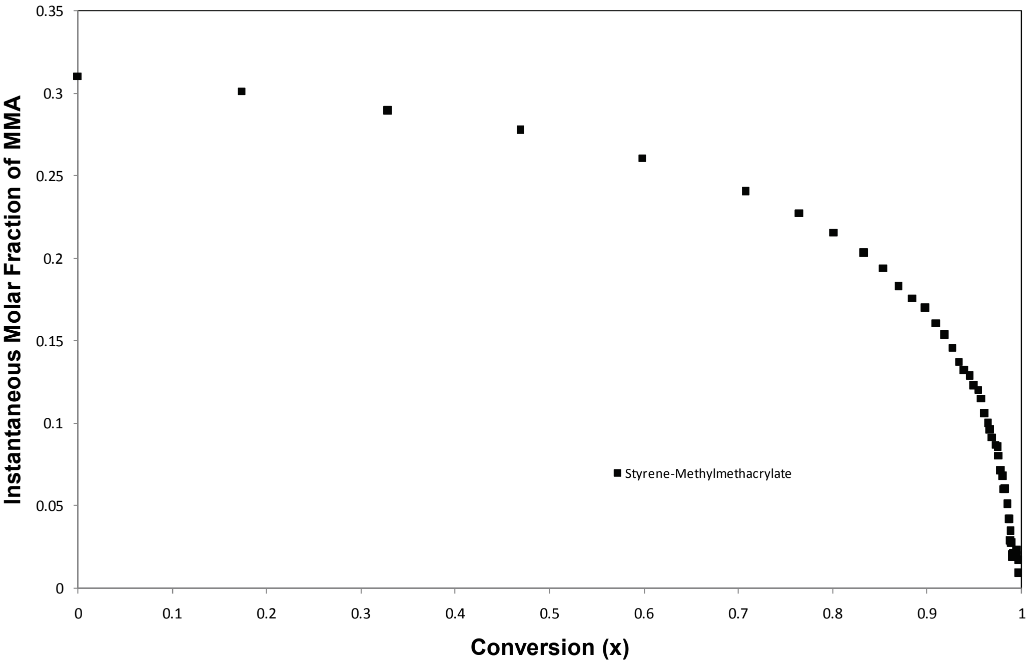


Fig. 5. Monte Carlo simulation results for the copolymerization of styrene- methyl methacrylate and acrylonitrile-methylmethacrylate copolymers : Instantaneous molar fraction as a function of conversion. The initial comonomer molar fractions in the reactor were $f_{0,St}$ = 0.75, $f_{0,MMA}$ = 0.25. The rate constants and other parameters are tabulated in our journal paper (Al-harthi 2009a)

Thus bimolecular termination reactions are programmed using the Monte Carlo approach. Other termination reactions can also programmed with a similar methodology/ slightly modified approach as appropriate to the situation. Thus were some examples which were used to apply the generalized Monte Carlo approach for programming our physical system (ATRcP).

We would like to show some important results like PDI and instantaneous molar fraction. PDI is defined as the ratio of weight average molecular weight to the number average molecular weight. It is an important factor which governs the uniformity in the molecular weight of the polymer. Once we have simulated the termination reactions, most of the results can be predicted. Thus the correct place to show the result after explaining how to simúlate the termination reactions. Figure 4 shows the simulated results for PDI obtained in ATRcP in a batch Reactor. We have shown that the PDI approaches a value of 1.1 in ATRcP which is coherent with the experimental data published in the field of ATRcP. PDI can be expressed mathematically in terms of means and standard deviation. We would like to encourage the students to search for the mathematical expression relating PDI in terms of means and standard deviations, hence we are not disclosing the simple mathematical expression. It also keeps the students interested and may also result in discovering other ways of incorporating the PDI. Figure 5 shows the instantaneous molar fraction of MMA in St-MMA produced in a batch reactor. The use of mayo-lewis equation can easily predict the instantaneous molar fraction.

4. Conclusion

A thorough understanding of Monte Carlo procedure based on Gillispie's algorithm has been detailed. Specific MATLAB commands have been given for ease in programming. This would be of benefit to the readers as this is one of the first attempts giving out the actual procedure and programming technique for modeling free radical copolymerization. This procedure has been used by graduate students, researchers at King Fahd University of petroleum and minerals with great success. The authorial Al-harthi had developed the program and we have always wanted to educate others on this technique which has been the prime purpose of the article.

5. References

J. He, H. Zhang, J. Chen, Y. Yang, Macromolecules 1997, *30*, 8010

J. He, L. Li, Y.Yang, Macromolecular Theory and Simulations 2000, *9*, 463

H. Tobita, *Macromol Theory Simul.* 2003, *12*, 32.

H. Tobita *Macromolecules*, 1995, *28*, 5119

J. Lu, H. Zhang, Y. Yang, *Makro Chemie Theory Simul*. 1993, *2,* 747.

M. Al-Harthi, J. Soares, L. Simon, *Macromolecular Material Science* 2006, *291*, 993

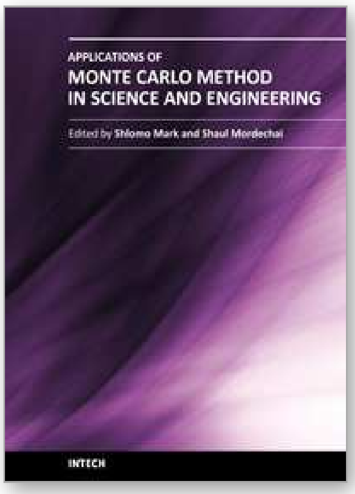
M. Al-Harthi, J. Soares, L. Simon, *Macromolecular Reaction Engineering* 2007, 1, 95

M. Al-Harthi, J. Masihullah, S.H. Abbasi, J. Soares *Macromolecular reaction engineering* 2009a in press

M. Al-Harthi, J. Masihullah, S.H. Abbasi, J. Soares *Macromolecular Theory and Simulation* 2009b in press

D.Gillespie, *J Phys Chem*. 1977, *81,* 2340

Editor Kryzsztof Matyjaszewski - Book Controlled/Living Radical Polymerization From Synthesis to Materials (ACS Symposium Series)


APPLICATIONS OF
MONTE CARLO METHOD
IN SCIENCE AND ENGINEERING
Edited by Shlomo Mark and Shaul Mordechai
INTECH

Applications of Monte Carlo Method in Science and Engineering

Edited by Prof. Shaul Mordechai

ISBN 978-953-307-691-1
Hard cover, 950 pages
Publisher InTech
Published online 28, February, 2011
Published in print edition February, 2011

In this book, Applications of Monte Carlo Method in Science and Engineering, we further expose the broad range of applications of Monte Carlo simulation in the fields of Quantum Physics, Statistical Physics, Reliability, Medical Physics, Polycrystalline Materials, Ising Model, Chemistry, Agriculture, Food Processing, X-ray Imaging, Electron Dynamics in Doped Semiconductors, Metallurgy, Remote Sensing and much more diverse topics. The book chapters included in this volume clearly reflect the current scientific importance of Monte Carlo techniques in various fields of research.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Mamdouh A. Al-Harthi (2011). MATLAB Programming of Polymerization Processes using Monte Carlo Techniques, Applications of Monte Carlo Method in Science and Engineering, Prof. Shaul Mordechai (Ed.), ISBN: 978-953-307-691-1, InTech, Available from: http://www.intechopen.com/books/applications-of-monte-carlo-method-in-science-and-engineering/matlab-programming-of-polymerization-processes-using-monte-carlo-techniques

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821