RMD_Digging Manual

RMD_Digging: A Toolkit for Information Mining and Analysis from ReaxFF Simulations

Qiang Liu

forliubinqiang@163.com

Institute of Nuclear Physics and Chemistry, Mianyang, Sichuan, China

✧ To cite:

Qiang Liu, RMD_Digging ToolKit, version, https://github.com/dadaoqiuzhi/RMD_Digging, access date.

✧ Other literatures:

(1) Liu, Q.; Liu, S.; Lv, Y.; Hu, P.; Huang, Y.; Kong, M.; Li, G. Atomic-scale insight into the pyrolysis of polycarbonate by ReaxFF-based reactive molecular dynamics simulation. Fuel 2021, 287, 119484, DOI: https://doi.org/10.1016/j.fuel.2020.119484.

(2) Liu, Q.; Huang, W.; Liu, B.; Wang, P.-C.; Chen, H.-B. Gamma Radiation Chemistry of Polydimethylsiloxane Foam in Radiation-Thermal Environments: Experiments and Simulations. ACS Appl. Mat. Interfaces 2021, 13 (34), 41287-41302, DOI: https://doi.org/10.1021/acsami.1c10765.

(3) C. Li, Q. Liu, W. Gong, Z. Zhou, Z. Yao, X. Meng, Study on the atomic scale of thermal and thermo-oxidative degradation of polylactic acid via reactive molecular dynamics simulation, Thermochim. Acta 709 (2022) 179144.

(4) Liu, Q.; Huang, W.; Chen, H. Paving the Way to Simulate and Understand the Radiochemical Damage of Porous Polymer Foam. ACS Materials Letters 2023, 2174-2188.

1 Development history of RMD_Digging

Prof. Adri C.T. van Duin of Pennsylvania State University and Prof. William A. Goddard, III develop the reactive force field simulation (ReaxFF) to model the reactive events and dynamic behavior of various systems like polymer materials, fuel, energetic materials, catalytic systems, metal materials, etc. ReaxFF simulation is mainly supported by the open-source Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS), whose primary developers are from Sandia National Labs and Temple University. However, the pre- and post-processing as well as corresponding analysis are very hard for many beginners due to the lack of specialized knowledge. When I studied at Sichuan University for my doctorate during 2017-2020, I ever investigated the thermolysis of polymer materials by ReaxFF simulations. I also take advantage of this method to study the radiation effects of polymers now in the Institute of Nuclear Physics and Chemistry. I wrote some codes to process the raw data generated during simulations ever since 2019 based on MATLAB language. With the development of the codes, I gradually realize that I should devote myself to developing a toolkit, which is called RMD_Digging. RMD is the abbreviation of Reactive Molecular Dynamics. Digging denotes the main functions of the toolkit, which is to dig out information from a wealth of data for further analysis. The first version of RMD_Digging (V1.0) was finished in 2020. RMD_Digging can be accessed on the code hoster of GitHub. Since then, I have also kept improving the functions of the toolkit by adding new functions, optimizing codes and debugging with the feedback from users, and this toolkit is updated to the second version (V2.0) in 2022 and V3.0 in 2023. Now the toolkit is renamed RMDigging when I decided to apply for its copyright in China. This toolkit (RMD_Digging on the GitHub) is also an open-source utility code for researchers, and it can be downloaded and distributed free of charge and rewritten to meet a special need. I will also stick to its development in the future. I hope this toolkit can prosper in this field and contribute to one’s study.

2 Main functions of RMD_Digging

The main functions of RMD_Digging can be divided into six modules (Figure 1). We will give a brief introduction and some examples.

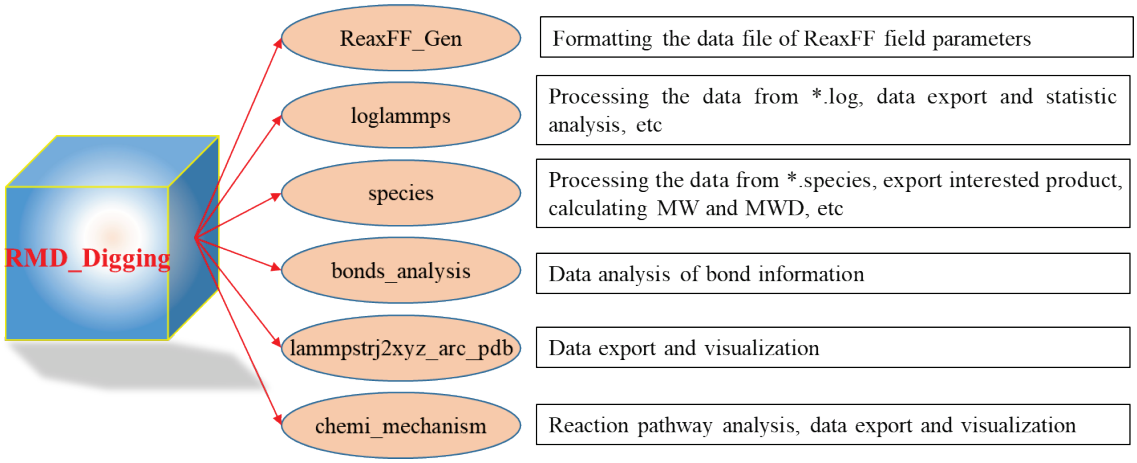


Figure 1 Six modules of the RMD_Digging toolkit.

2.1 ReaxFF_Gen module

This module is aimed to format the data file of ReaxFF parameters from the literature report. For picture files or PDF without editable permission, an online OCR tool is recommended to extract information first. Then the ReaxFF_Gen module can be taken to process the files. The detailed instruction is as follows (Figure 2):

1. Make a copy of the data file containing the ReaxFF parameters, delete the title lines of each section in the duplicate and leave the numeric data with comments alone, each section should be separated by a blank line（eg. ReaxFF_Gen/*input-example.txt*）.
2. Copy the processed content in the duplicate into the *input.txt* file in the ReaxFF_Gen folder.
3. Start MATLAB and change the working directory to the ReaxFF_Gen folder, and input ReaxFF_Gen to invoke the corresponding program.
4. Input the chemical elements involved in the *input.txt* in the cell array. Note: the sort order of these chemical elements should be in line with the original

file. All will be done within several seconds at most.

5. After the normal termination of the program, the formatted parameters will be written in the *outpute.txt* file. For the case if the names of some chemical elements are two characters, one can manually refine the alignment of these lines.
6. Open the *outpute.txt* file and copy the removed title lines in Step 1 into the corresponding positions. Now one can copy the final file and rename it as they wish.

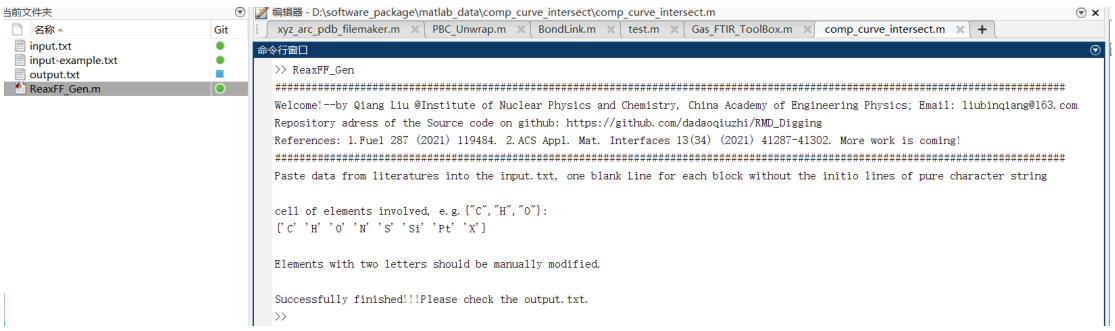


Figure 2 Illustration of running ReaxFF_Gen module.

2.2 loglammps module

This functional module can process the data in the **.log* file outputted to the terminal by the thermos_style command, which can screen out the interesting data and carry out statistical analysis. The running example is shown below (Figure 3):

1. Start MATLAB and change the working directory to the loglammps folder. Make a copy of the *log.** file, it is not necessary to remove all the non-numeric text for V3.0. It is also not necessary to copy the number block with the same size (same column) into the *log.lammps* file, and the user can control this by limit the column number and export data several times.
2. Input loglammps to invoke the corresponding program.
3. According to the prompt, you should input the filename (*log.lammps*) to be processed.

4. Input the total column number of the data expected to be exported, which limit the data object. Then input the column number of the interesting data, which should start from 1 and multi numbers should be separated by a space. “all” can be used to export all data.
5. Input the timestep of the simulation (default unit: fs) and the Step/Span number used to average the data (Note: the raw data are usually outputted every hundred steps).
6. The processed data is saved in the *dataoutput* matrix while the raw data is saved in the datainput matrix in the workspace. One can export these data in *dataoutput* matrix to the *output_mydata.xlsx* file by uncommenting several lines.
7. One can further perform statistical analysis using statisave code and just input option one by one according to the prompt. The final processed data are assigned to the datastatis matrix。
8. Go to step 1 to process other data blocks if necessary.

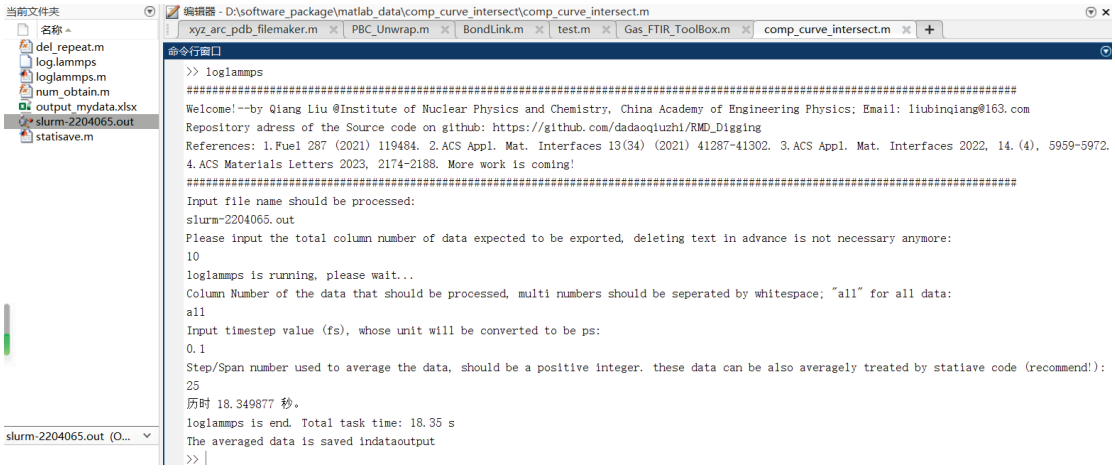


Figure 3 Illustration of running loglammps module.

2.3 species module

This module is aimed to process and generate data related to the *species.** file,

For example, to obtain the evolution of specified products or reactants, to select and export special species and to calculate the molecular weight or molecular weight distribution of the limited species.

2.3.1 species_analysis

The species_analysis code will sort all the species according to the timestep. Note: this code cannot distinguish the isomers. Some instances are displayed here (Figure 4).

1. Make a copy of the *species.** file (here is species-PDMS.out) to the species folder.
2. Start MATLAB and change the working directory to the species folder. Input species_analysis to invoke this code.
3. Input the name of the *species.** file according to the requirement.
4. The code will take some time to finish this task. The time cost depends on the size of the *species.** file. A pop-up window will appear when the task is all normally finished.
5. The final output of the data can be found in the cell array named outputdata。

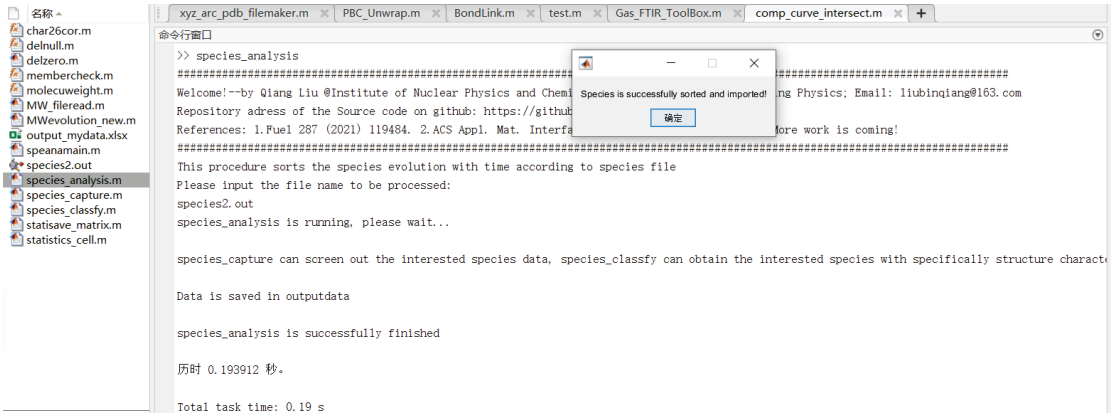


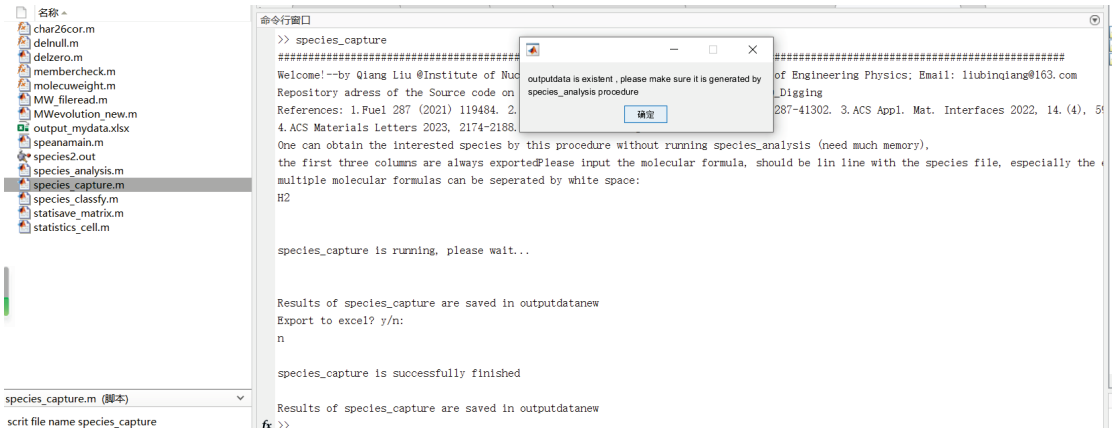
Figure 4 Diagram of the running of species_analysis code in the species module.

2.3.2 species_capture

The subroutine called species_capture can find out the interesting species from

the generated cell array called outputdata according to the molecular formula given by the user (Figure 5).

1. Make sure the working directory is located in the species folder. Input species_capture to call this code.
2. Input the molecular formula expected to be exported for further analysis. For multiple molecular formulas, one can separate them by white space. Note: the molecular formula should be the same as that in the *species.** file, and the order of these elements can also be confirmed by the *in.** file.
3. Generally, several seconds is enough to complete this job and the processed data is saved in the outputdatanew matrix。
4. The code will ask one if to export these data to *output_mydata.xlsx*. I suggest not to do so, one can just copy them from the outputdatanew matrix.
5. From V3.0 on, running species_analysis program in advance is not necessary, which will import all products (need much time and memory for large file). Instead, run species_capture directly to import the interested products frame by frame.



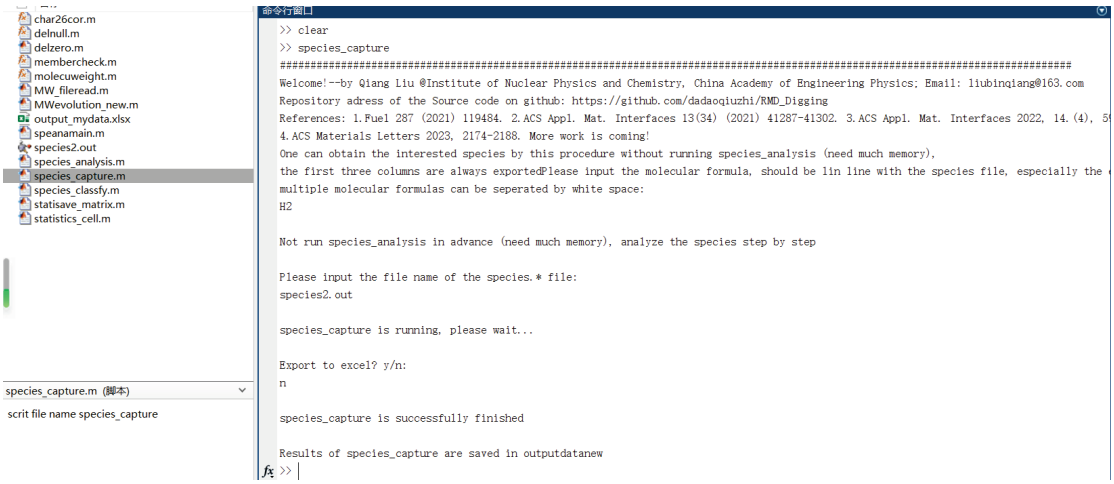


Figure 5 Example of running species species_capture code in the species module.

2.3.3 species_classfy

The species_classfy code is used to retrieve species data from the outputdata cell array according to species limitations determined by some criteria (like restrictive conditions on molecular weight, elements and their content). One can repeatedly take this strategy to specify their target species (Figure 6):

1. Make sure the working directory is located in the species folder. Input species_classfy to call this code.
2. Four options will be shown immediately in the interactive interface, and some examples are provided to explain their meaning。
3. Choose “a”, “b”, “c” or “d” according to your requirements. Meanwhile, the code needs to know if to sum up the filtered data. Then wait for the end of the subroutine.
4. The legal data will be extracted to the dataexport cell in the working space and the summing data is saved in sumdata.

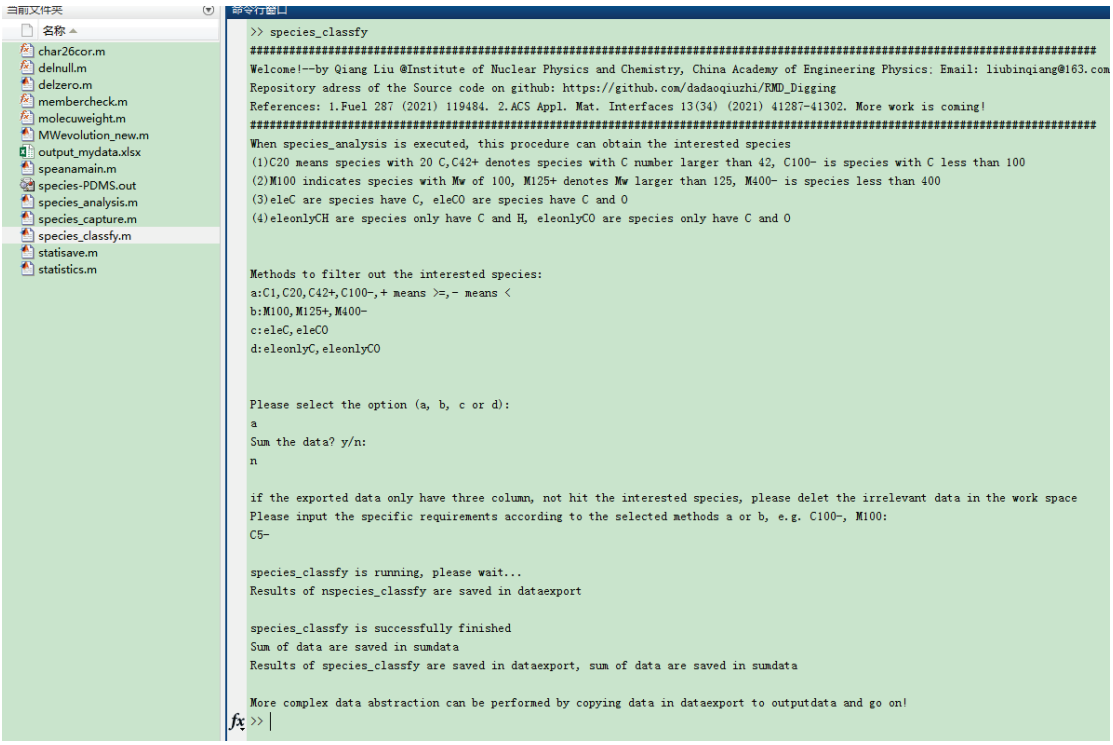


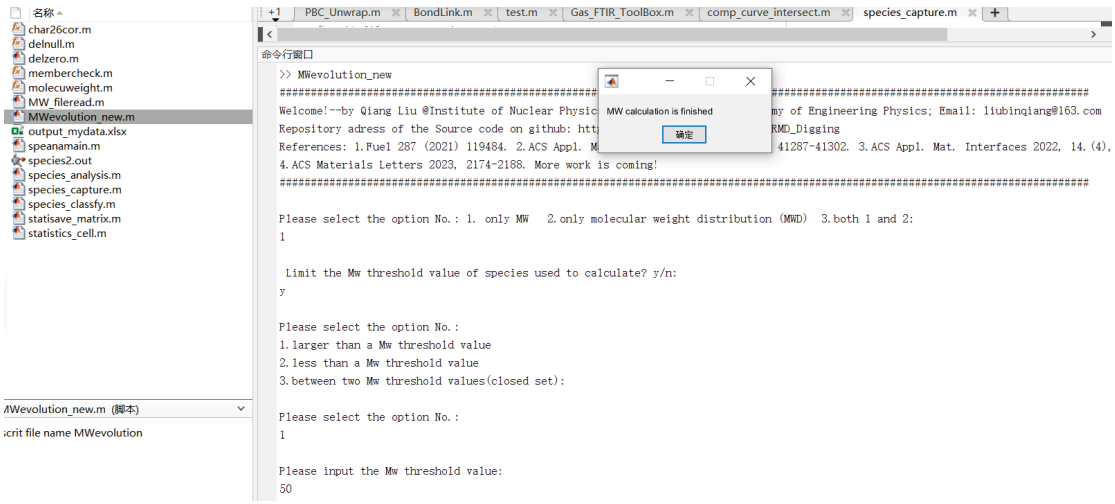
Figure 6 Instance of running species species_classfy code in the species module.

2.3.4 MWevolution_new

The MWevolution_new code is used to calculate the number-average and weight-average molecular weight as well as molecular weight distribution of the specified species (Figure 7). From V3.0 on, running species_analysis program in advance is not necessary, which will import all products (need much time and memory for large file). Instead, run MWevolution_new directly to import and process the interested products frame by frame：

1. Make sure the working directory is located in the species folder. Input MWevolution_new to call this code.
2. Make option on calculation according to the requirements and hints. Option 1 means only calculating number-average and weight-average molecular weight. Option 2 indicates only calculating the molecular weight distribution of the specified frame. Option 3 is forced to calculate the molecular weight and molecular weight distribution of the specified frame.

3. If option 1 or 3 was chosen in the last step, the subroutine will ask the user whether to restrain the molecular weight of the species which will be considered to be calculated later. Three items are provided to limit the molecular weight of the interesting species to be larger than and less than one critical number as well as between a certain range. For item 3, a matrix should be given to limit the minimum and maximum molecular weight, eg. [200 50000].
4. If item 2 or 3 is chosen in the last step, the frame/timestep number is required for the molecular weight distribution. There are two ways to specify the frame/timestep number. Option 1 is to specify the frame/timestep number manually. Option 2 is to specify the frame/timestep number via arithmetic sequence. For option 2, the following information is required: the minimum timestep, maximum timestep, output frequency (LAMMPS), and common difference.
5. Then wait for the end of the running code until the corresponding pop-up window is shown. The molecular weight data are saved in the MD matrix while molecular weight distribution data is stored in the MWDdata matrix. The massive zero data can be removed by delzero code.



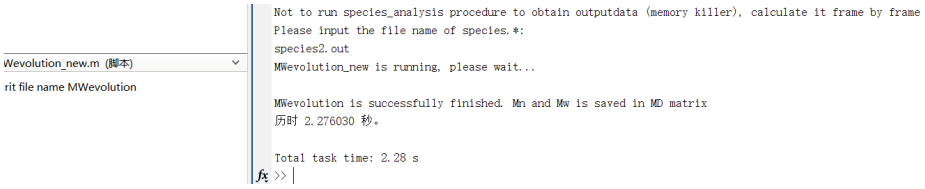


Figure 7 illustration of running MWevolution_new code in the species module.

2.4 bonds_analysis module

This module can deal with the bond information, which is the key to identify products and perform topology analysis.

2.4.1 bonds_analysis_speedup

Both bonds_analysis_speedup and bonds_analysis are the main routines to handle the bond information. We recommend bonds_analysis_speedup for practical use due to its faster operation (Figure 8).

1. Start MATLAB and change the working directory to the bonds_analysis folder. Input bonds_analysis_speedup to invoke the corresponding program.
2. Input the name of the file to be processed.
3. Input the interesting frame/ timestep number.
4. Input the total atom number of the systems.
5. Wait for the code to run and finish successfully. The data is saved in the bondoutdata matrix.

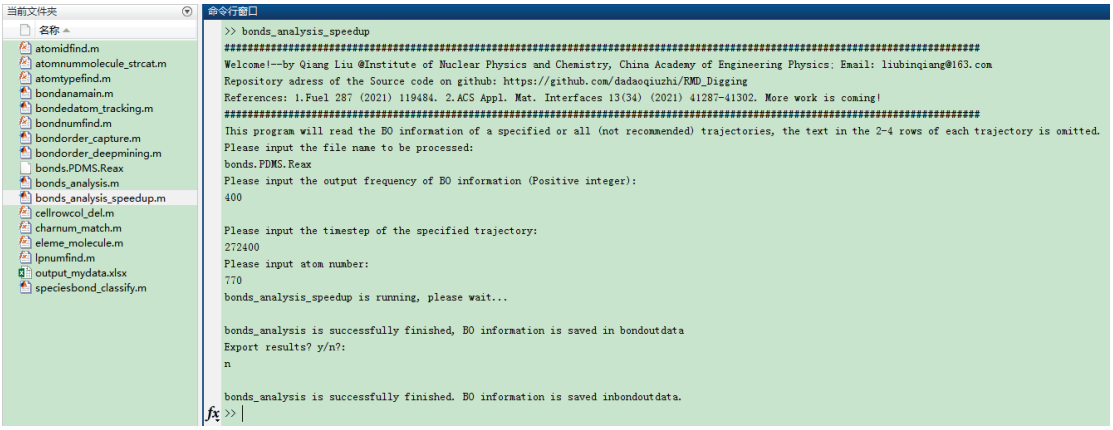


Figure 8 Examples of running bonds_analysis_speedup code of the bonds_analysis module.

2.4.2 bondorder_deepmining

The bondorder_deepmining code can further treat the bond information in view of molecular formula or species compositions. The as-processed data block is separated by the pound sign (Figure 9).

1. Start MATLAB and change the working directory to the bonds_analysis folder. Input bonds_ deepmining to invoke the corresponding program. The bonds_analysis_speedup code should be executed first.
2. Step by step, input the interesting frame/timestep number, the involved elements whose order should be in line with the original file as mentioned earlier.
3. The console output message that you should verify if you performed element mapping (eg, O is mapped to S). For general simulations, it is no (n).
4. When the subroutine exits, the handled bond data can be found in the tarBOinform cell while the relevant molecular formula or species compositions are stored in the tarelenummatch cell. Data in tarelenummatch are generally identical with the *species.** file in almost all cases. The exception can be triggered if an abnormality occurs. The possible reason will be shown by the program and some solutions are given, where the most possible cause is the difference in the output frequency of bond data and

species data. One can ignore this warning to continue.

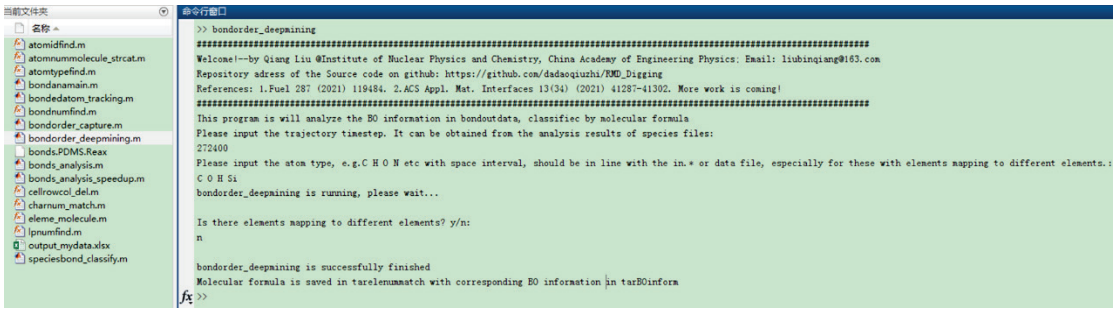


Figure 9 Examples of running bondorder_deepmining code of the bonds_analysis module.

2.5 lammpstrj2xyz_arc_pdb module

The purpose of this module is to export formatted files based on trajectory file (*.*lammpstrj*) and bond data file (*bonds.**). The formatted files indicate standard xyz, arc or pdb files. The user can select one format according to their software resource and personal habit. Usually, Materials Studio or VMD are the most intended software. They can give pretty pictures for further analysis and paper publications or academic reports.

2.5.1 lammpstrj_analysis

The function of the lammpstrj_analysis code is to read and process topological information in the trajectory file (Figure 10).

1. Start MATLAB and change the working directory to the lammpstrj2xyz_arc_pdb folder. Make a copy of the *.*lammpstrj* file (here is *PLA.lammpstrj*) to the lammpstrj2xyz_arc_pdb folder.
2. Input lammpstrj_analysis to invoke this code.
3. Input the filename of the trajectory file, the output frequency of the trajectory file, the interesting frame/timestep number of the trajectory, and the total atom number according to the requirements.

4. After the normal exit of the code, one should choose whether to export the results to Excel (*output_mydata.xlsx*). No is always recommended.
5. The processed data are saved in the trjdata matrix.

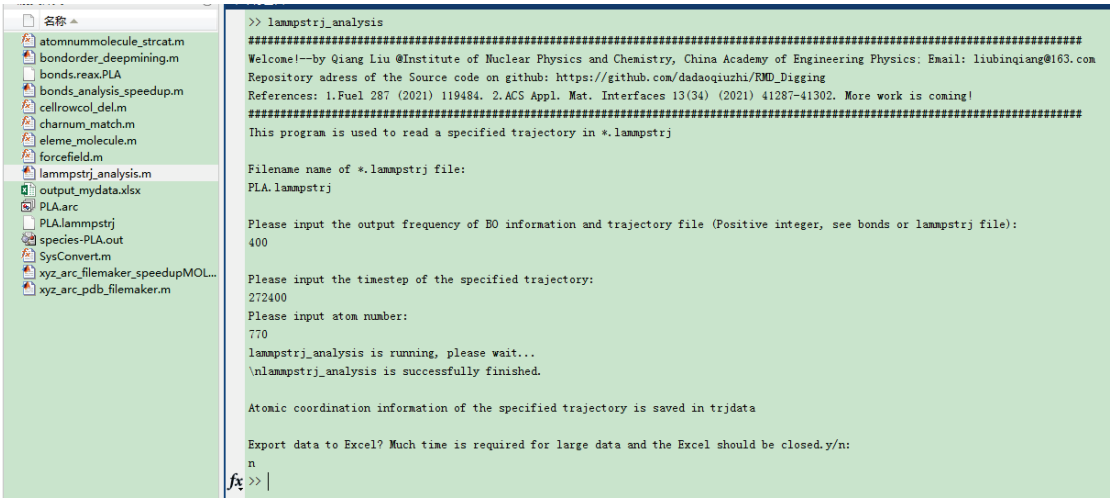


Figure 10 Examples of running the lammpstrj_analysis code of the lammpstrj2xyz_arc_pdb module.

2.5.2 xyz_arc_pdb_filemaker

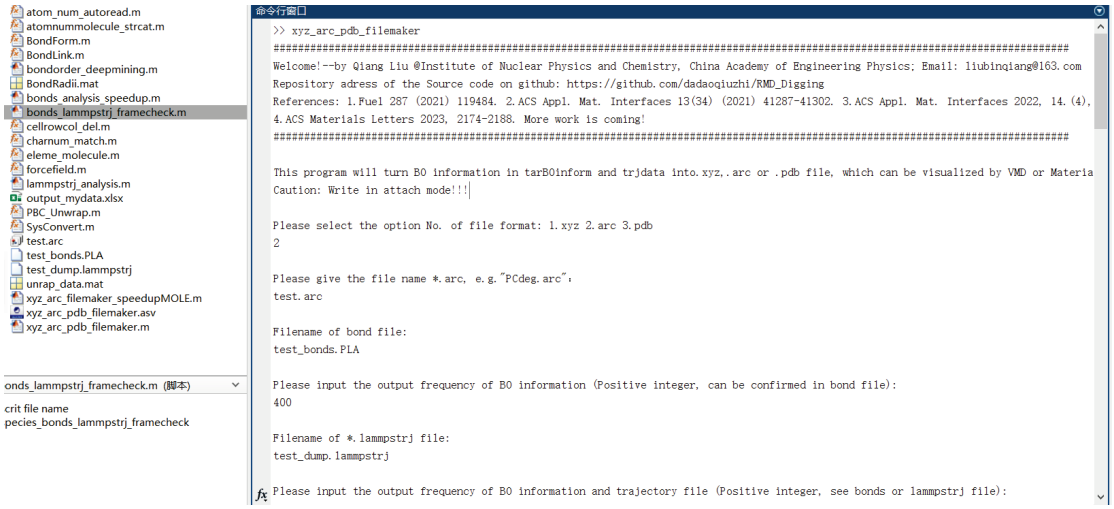
The xyz_arc_pdb_filemaker subroutine is aimed at exporting visual files by combining topological trajectory file and bond data file. Both single static diagrams and dynamic graphs can be generated via this code. The format of the exported file can be xyz, arc or pdb (Figure 11).

1. Make sure the working directory is situated in the lammpstrj2xyz_arc_pdb folder. Make a copy of the **.lammpstrj* file (here is *PLA.lammpstrj*) and *bonds.** (*bonds.reax.PLA*) file to the lammpstrj2xyz_arc_pdbfolder.
2. Input xyz_arc_pdb_filemaker to call this code.
3. Select one option for the desired output format, namely xyz, arc or pdb.
4. Name the output file according to your last choice and the file suffix should be explicitly given.

5. Input the file name of *bonds.** and its output frequency.
6. Input the file name of **.lammpstrj* and its output frequency.
7. Input the total atom number. From V3.0 on, the program will automatically read the total atom number.
8. The program asks the user to input the number for frame No. check of bonds.* and *.lammpstrj files, so as to avoid mismatch induced error. One can choose no in the next step if you are self-confident about the non-existent of the possible problem.
9. User is asked to give answer about if to run check process for the consistency of frame No. in the bonds.* and lammpstrj.* (need extra time and memory for large files). If the residual check file in the workspace is found, rerun question will be triggered.
10. Input the involved elements whose order must be in line with the original file as mentioned earlier (check *in.** or *species.** files). The console output message that you should verify if you performed element mapping (eg, O is mapped to S). For general simulations, it is no (n).
11. If the mismatch number between bonds.* and lammpstrj.* files can not be ignored, the program will output the data to screen and let the user to choose corresponding methods to solve this problem. The program will automatically take care of the target frame No. input by user later (give correction).
12. Input the number system (binary, decimal and decimal, etc) for the task, which will be considered to encode the atom id. The selection of the number system is based on the total atom number and the limitation on the length of the name/serial number of any atom (id). A recommended value will be shown and the user should input a value not less than the recommended one.
13. Input the frame/timestep number of the interesting trajectory. There are two ways to specify the frame/timestep number. Option 1 is to specify the frame/timestep number manually. If there are several frames, these numbers should be given simultaneously and separated by white space. Option 2 is to

specify the frame/timestep number via arithmetic sequence. For option 2, the following information is required: the minimum timestep, maximum timestep, output frequency (LAMMPS), and common difference. Option 3 includes methods described in option 1 and 2. And the input sequence should abide by the rules: option 1 first and then option 2.

14. Input answers for a periodic structure. ON for periodic structure, OFF for the contrary.
15. Input answers for scaled coordinate. User should check the *in.** file to know if the coordinate in the **.lammpstrj* file is scaled. The scaled coordinate will be converted into a real space coordinate. I recommend not to scale the coordinate.
16. The program asks whether to perform unwrap for atoms with coordinate affected by ghost position, causing discontinuity of structure. This can be managed by Materials studio via changing PBC display mode. However, this function here cannot realize this goal now.
17. For periodic structure, the cell parameters (angle, four decimal place) and space group (P1 for general case) should be provided according to the prompt message.
18. Finally, the code will run and export specified files. The detailed information of code invoking is output to show the task progress.




atom_num_autoread.m
atomnummolecule_strcat.m
BondForm.m
BondLink.m
bondorder_deepmining.m
BondRadii.mat
bonds_analysis_speedup.m
bonds_lammpstrj_framecheck.m
cellrowcol_del.m
charnum_match.m
eleme_molecule.m
forcefield.m
lammpstrj_analysis.m
output_mydata.xlsx
PBC_Unwrap.m
SysConvert.m
test.arc
test_bonds.PLA
test_dump.lammpstrj
unrap_data.mat
xyz_arc_filemaker_speedupMOLE.m
xyz_arc_pdb_filemaker.asv
xyz_arc_pdb_filemaker.m
onds_lammpstrj_framecheck.m (脚本)
crit file name
pecies_bonds_lammpstrj_framecheck
命令行窗口
400
Automatically read atom number from th
The atom number line is found, ready t
The atom number is found, which is 770
lease treat the frame No. of lammpstrj.* file by the following method
确定
Please input the number for frame No. check of bonds.* and *.lammpstrj files compared with species file,
avoiding mismatch induced error, >=5 is suggested:
5
If to run check process for the consistency of frame No. in the bonds.* and lammpstrj.* (need extra time and memory)y/n:
y
Check the consistency of frame No. between bonds.* and lammpstrj.* files.
Sometimes energy minimization causes the inconsistent record problem
species_bonds_lammpstrj_framecheck is running, please wait...
Firstly, check frame No. in bonds.* file, please wait...
The appointed frame No. in bonds.* file (5 in total) has been read
Then, check frame No. in lammpstrj.* file, please wait...
The appointed frame No. in lammpstrj.* file (5 in total) has been read
bonds_lammpstrj_framecheck is end
5 frame No. in the bonds.* and lammpstrj.* are as follows:
0 100000
400 100400
800 100800
1200 101200
1600 101600
As above, frame No. in the bonds.* and lammpstrj.* files are inconsistent,
please treat the frame No. of lammpstrj.* file by the following method:
Please treat the frame No. of lammpstrj.* file (outputnew) by the following method:
1.Substract an integer 2.Add an integer 3.No action
3
Please input the unsigned number, eg.8, 29, or 377, etc.:
0
Please input atom type like C,H.O,N, seperated by white space, corresponding to 1,2,3,4...n (see *.data or in.*,
especially for element mapping:
C O H
Does there exist element mapping?y/n:
n
Different coding system is adopted according to the atom number (ASCII)
10 base coding system is recommended for atom id
Please select coding system for atom id, positive integer and not less than the recommended value:
10
Please select the export method No.:
1.Manually specify multi-trajectories
2.Monotonically increasing frame(s) in arithmetic sequence(closed interval)
3.Both method 1 and 2 (1 first, 2 last)
1
Trajectories in ascending order, seperated by white space,
100000 300000
Please input periodic boundary condition, ON/OFF:
ON
Does the coordinate is scaled in the *.lammpstrj file, y/n:
n
If to perform unwrap for atoms with coordinate affected by ghost position, causing discontinuity of structure,y/n? (Unusable Now!):
n
Periodic boundary condition, alpha, four decimal digits:
90.0000
Periodic boundary condition, beta, four decimal digits:
90.0000
Periodic boundary condition, gamma, four decimal digits:
90.0000
Point group name, eg."(P1)" for *.arc and "P 1" for *.pdb:
(P1)
xyz_arc_pdb_filemaker is running, p;ease wait...
xyz_arc_pdb_filemaker is searching for Group 1 trajectory: 100000,2 trajectories in total
Step1:Group 1 trajectory 100000 is successfully processed by bonds_analysis_speedup and bondnumdata is generated,
continue running bondorder_deepmining program, please wait...
Step2:Group 1 trajectory 100000 is successfully processed by bondorder_deepmining, and tarBOinform is generated,
continue running lammpstrj_analysis, please wait...

c

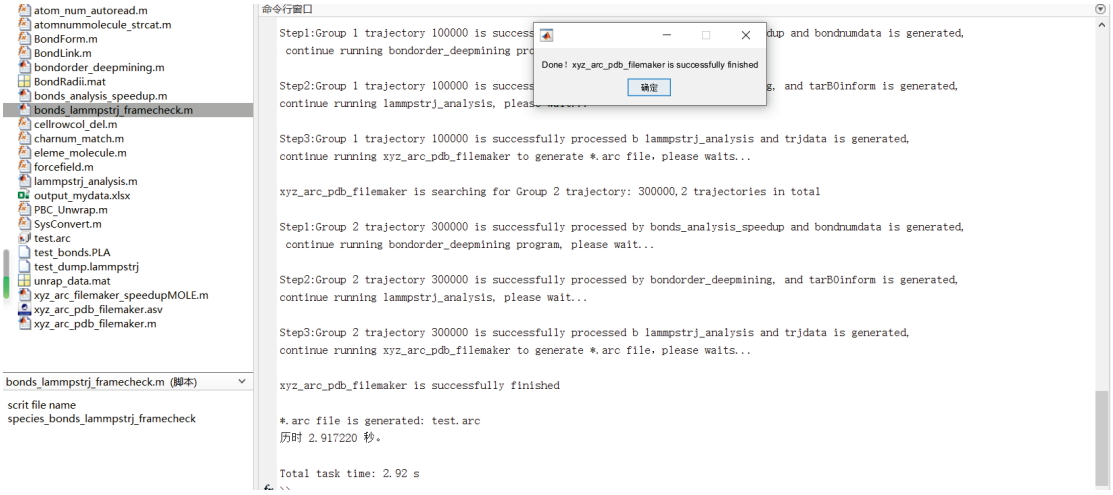


Figure 11 Examples of running xyz_arc_pdb_filemaker code of the lammpstrj2xyz_arc_pdb module.

2.5.3 xyz_arc_filemaker_speedupMOLE

The function of the xyz_arc_filemaker_speedupMOLE code is the same as the xyz_arc_pdb_filemaker, but it can only generate xyz or arc files. Users can try this subroutine on their own.

2.6 chemi_mechanism

This module is used to study the reaction pathways and generate visualization files. Further analysis can be performed with the help of Materials Studio or VMD. The code will invoke most core subroutines in other modules.

2.6.1 chemi_mechanism

The code named chemi_mechanism is the core integrated code to analyze the reaction pathways and corresponding mechanisms, which relies on the most main code in other modules. The in-depth analysis also depends on the visualization analysis with the help of bond order data (Figure 12).

1. Start MATLAB and change the working directory to the chemi_mechanism

folder. Make a copy of the *species.** (*species-PLA.out*) file,**.lammpstrj* file (here is *PLA.lammpstrj*) and *bonds.** (*bonds.reax.PLA*) file to the chemi_mechanism folder.

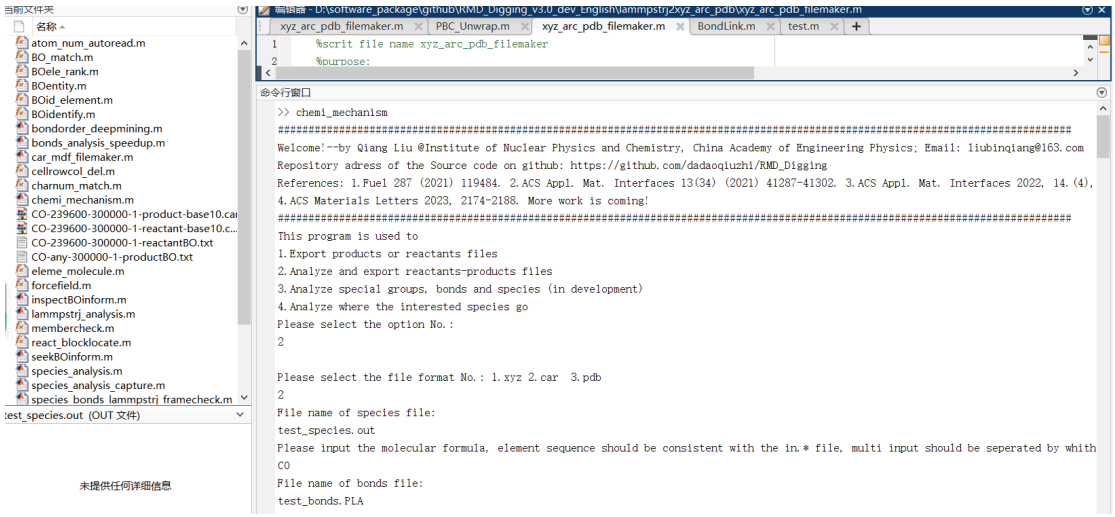
2. Input chemi_mechanism to invoke this code.
3. The subroutine shows the four functions first and waits for feedback. These functions include: (1) Export products or reactants files; (2) Analyze and export reactants-products files; (3) Analyze special groups, bonds and species (in development); (4) Analyze where the interested species go. Users should select one item to continue.
4. Here we continue our tutor by selecting option 2. The main process of other options has little difference. One can finish other analysis task according to this tutor and corresponding explanatory text.
5. Input the file name of the *species.** (*species-PLA.out*).
6. Input the molecular formula of the interesting product (CO2). Note: The molecular formula should be the same as that in the *species.** file, and the order of these elements can also be confirmed by the *in.** file.
7. Input the file name of *bonds.** and **.lammpstrj*.
8. Input the output frequency of bond information and trajectory information or their least common multiple. Now the output frequency is deemed to be the same. In the future, I will deal with the case that the output frequency separately with a view to the actual situation. Besides, I consider to obtain these data automatically. If the user uses two different output frequencies for bond information and trajectory information, one should input the lowest common multiple to gain reasonable results.
9. Input the total atom number of the simulation. From V3.0 on, the program will automatically read the total atom number.
10. The program asks the user to input the number for frame No. check of bonds.* and *.lammpstrj files compared with species.* file, so as to avoid mismatch induced error. One can choose no in the next step if you are self-confident about the non-existent of the possible problem. User is asked

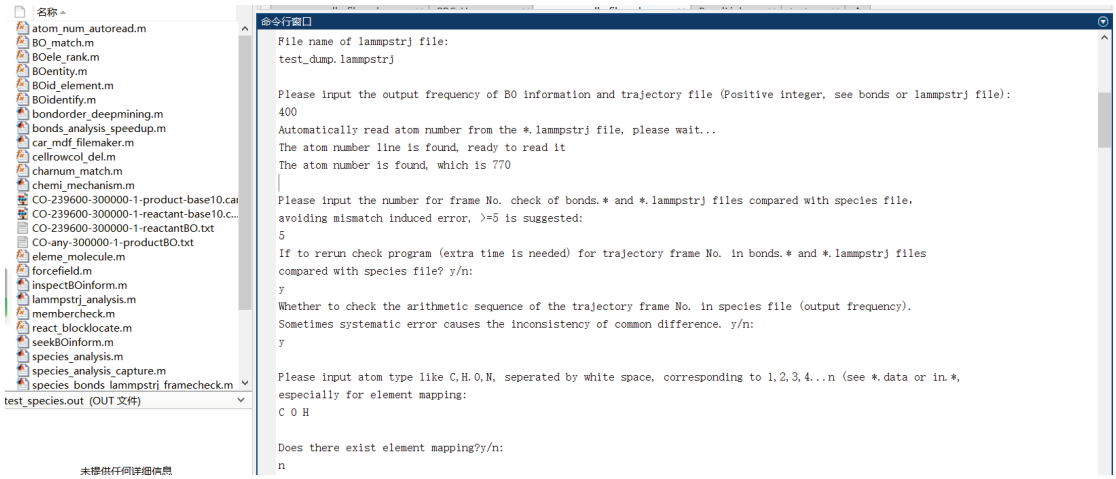
to give answer about if to run check process (need extra time and memory for large files). If the residual check file in the workspace is found, rerun question will be triggered.

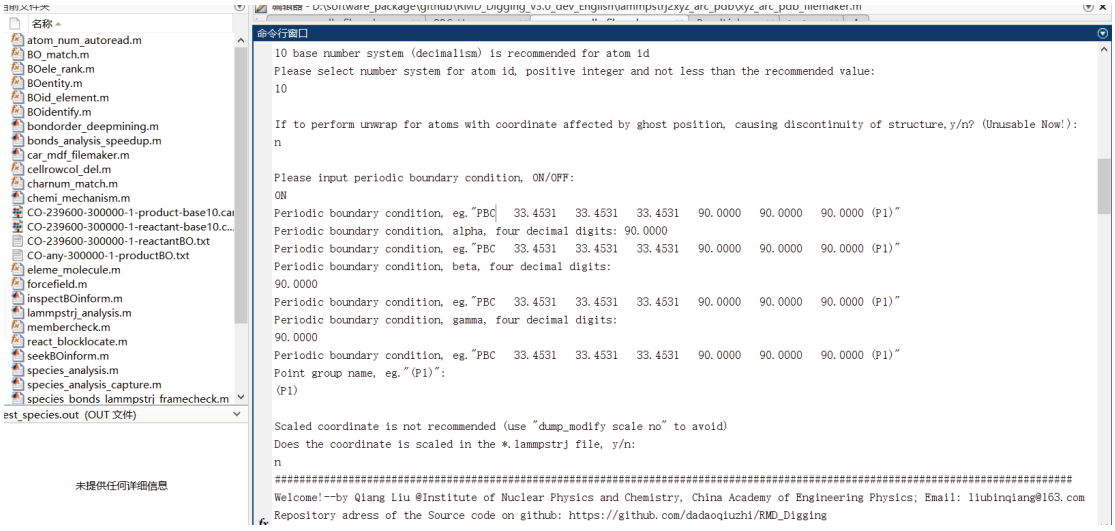
11. 10. The program asks the user to check the frame No. in the species.* file meets the arithmetic progression. Sometimes energy minimization will cause problem.
12. Input the atom type used in the simulation, which should be separated by white space for more than one element. And the order must be identical to the information in **.data* or *in.** files.
13. The console outputs message that you should verify if you performed element mapping (eg, O is mapped to S). For general simulations, it is no (n).
14. Input the number system (binary, decimal and decimal, etc) for the task, which will be considered to encode the atom id. The selection of the number system is based on the total atom number and the limitation on the length of the name/serial number of any atom (id). A recommended value will be shown and the user should input a value not less than the recommended one.
15. The program asks whether to perform unwrap for atoms with coordinate affected by ghost position, causing discontinuity of structure. This can be managed by Materials studio via changing PBC display mode. However, this function here cannot realize this goal now.
16. Input answers for a periodic structure. ON for periodic structure, OFF for the contrary.
17. For periodic structure, the cell parameters (angle, four decimal place) and space group (P1 for general case) should be provided according to the prompt message.
18. Input answers for scaled coordinate. User should check the *in.** file to know if the coordinate in the **.lammpstrj* file is scaled. The scaled coordinate will be converted into a real space coordinate. I recommend not to scale the coordinate.

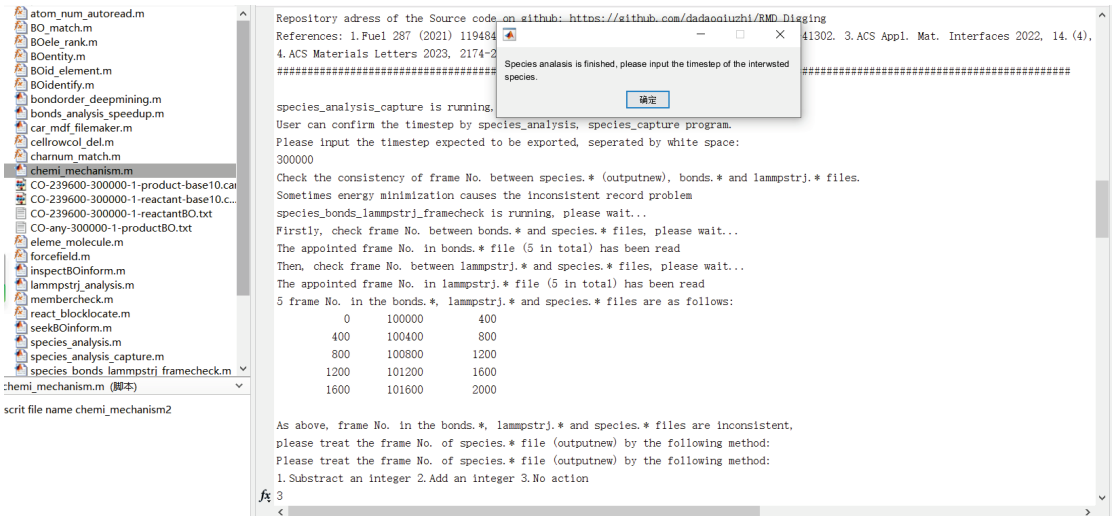
19. The chemi_mechanism code invokes species_analysis code to read and process species data. Note: if there has been a variable related to the processed species data, a warning is triggered to ask if re-execute the species_analysis code. This function is aimed at several analysis requirements, which is time-saving for large data. Users can also execute species_analysis code individually to obtain basic analysis before using chemi_mechanism code.
20. Then species_capture is called to abstract the interesting products.
21. When the species information is ready and the species module exists, a pop-up window is shown.
22. The chemi_mechanism code asks for further input before the next action.
23. Input the frame/timestep number of the interesting products.
24. The program checks the total number of interesting products specified in the frame, which is printed on the screen. Meanwhile, the program asks whether to restrict the number of exportation times of the reaction pathway pairs (products->reactants). Because there are usually many reaction pathways, so the restriction on exportation times is always necessary. This will reduce the search range. Users should act according to circumstances.
25. If you expect to impose restriction on exportation times, an explicit number should be given. Otherwise, all pairs of reaction pathways will be searched and exported.
26. The program further asks if to only search the trajectory frame with varying number of products. This can save much time if users authorize the program to only search the trajectory frame with a changed product number. For option 2, the subroutine only searches the foregoing trajectory frame with decreasing product number. For option 4, the subroutine only searches the subsequent trajectory frame with the increasing product number.
27. When all inputs are accepted, the chemi_mechanism code calls the bonds_analysis module and lammpstrj2xyz_arc_pdb module to perform the appointed task. The completed progress is briefly shown in real-time.

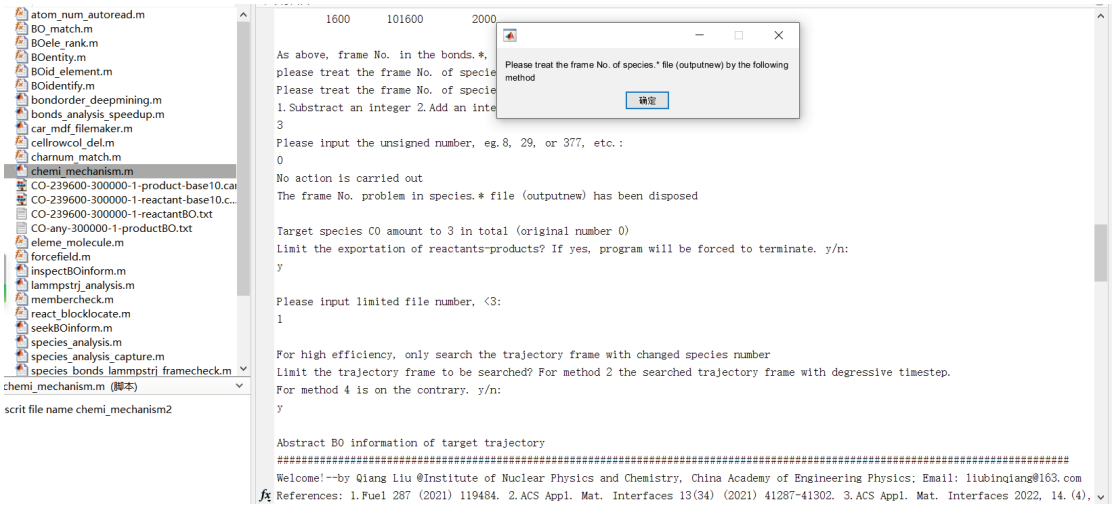
28. Finally, the target files with the specified format are generated in the working folder. One can further analyze them by other software like Materials Studio or VMD. Moreover, these formatted files are in favor of searching, checking and analyzing other relevant information.

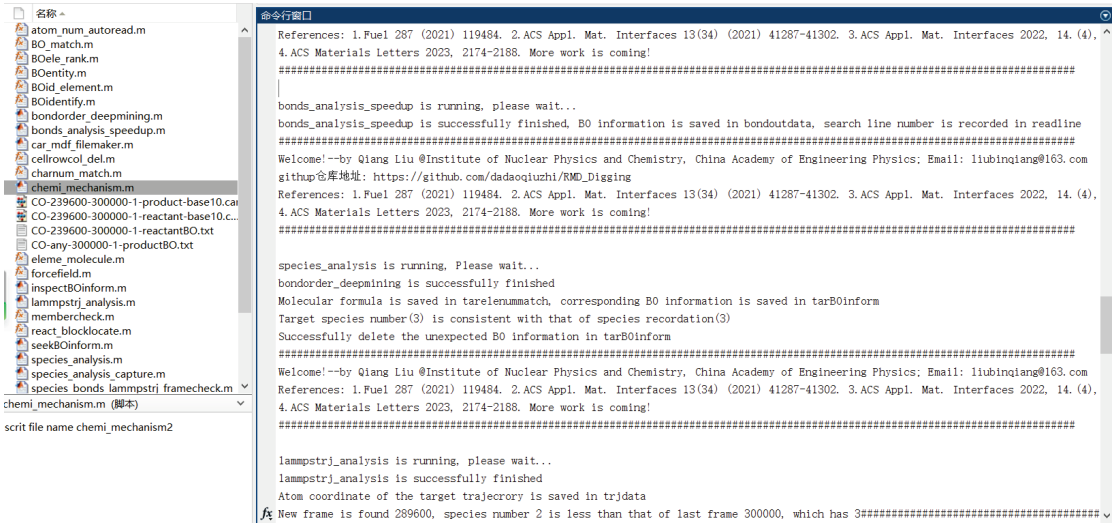


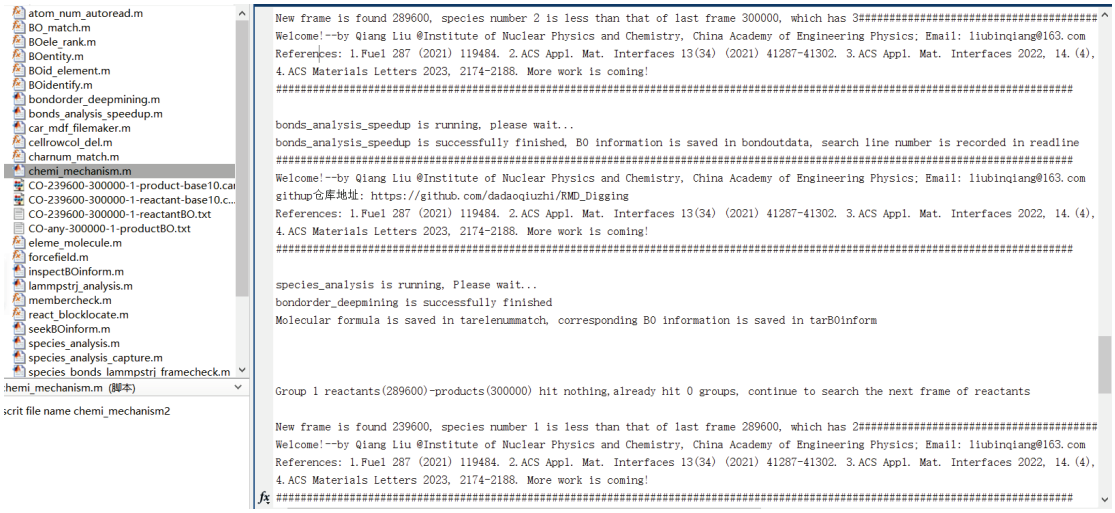


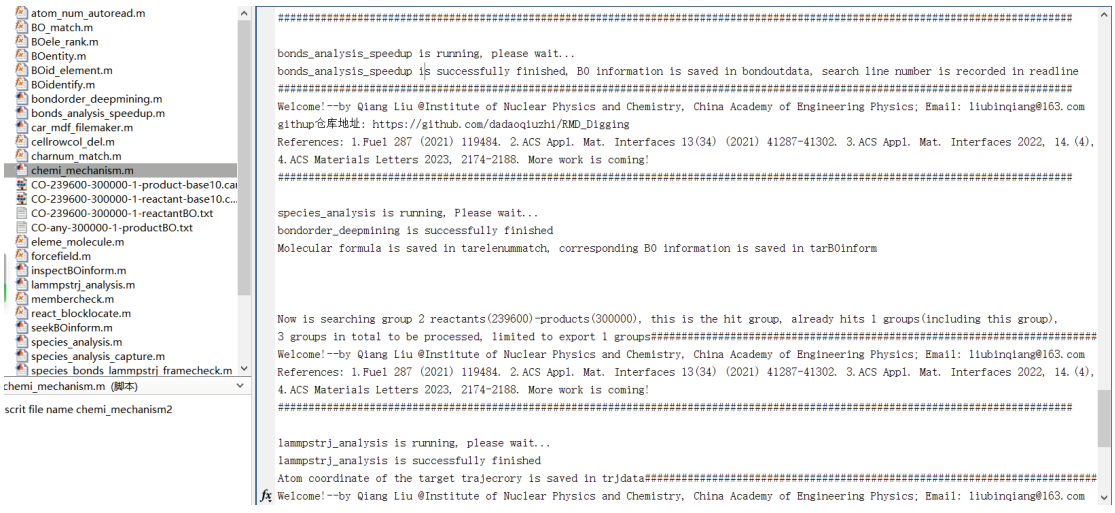

名称
atom_num_autoread.m
BO_match.m
BOele_rank.m
BOentity.m
BOid_element.m
BOidentify.m
bondorder_deepmining.m
bonds_analysis_speedup.m
car_mdf_filemaker.m
cellrowcol_del.m
charnum_match.m
chemi_mechanism.m
CO-239600-300000-1-product-base10.ca
CO-239600-300000-1-reactant-base10.c...
CO-239600-300000-1-reactantBO.txt
CO-any-300000-1-productBO.txt
eleme_molecule.m
forcefield.m
inspectBOinform.m
lammpstrj_analysis.m
membercheck.m
react_blocklocate.m
seekBOinform.m
species_analysis.m
species_analysis_capture.m
species_bonds_lammpstrj_framecheck.m
est_species.out (OUT 文件)
未提供任何详细信息
命令行窗口
10 base number system (decimalism) is recommended for atom id
Please select number system for atom id, positive integer and not less than the recommended value:
10
If to perform unwrap for atoms with coordinate affected by ghost position, causing discontinuity of structure, y/n? (Unusable Now!):
n
Please input periodic boundary condition, ON/OFF:
ON
Periodic boundary condition, eg."PBC 33.4531 33.4531 33.4531 90.0000 90.0000 90.0000 (P1)"
Periodic boundary condition, alpha, four decimal digits: 90.0000
Periodic boundary condition, eg."PBC 33.4531 33.4531 33.4531 90.0000 90.0000 90.0000 (P1)"
Periodic boundary condition, beta, four decimal digits:
90.0000
Periodic boundary condition, eg."PBC 33.4531 33.4531 33.4531 90.0000 90.0000 90.0000 (P1)"
Periodic boundary condition, gamma, four decimal digits:
90.0000
Periodic boundary condition, eg."PBC 33.4531 33.4531 33.4531 90.0000 90.0000 90.0000 (P1)"
Point group name, eg."(P1)":
(P1)
Scaled coordinate is not recommended (use "dump_modify scale no" to avoid)
Does the coordinate is scaled in the *.lammpstrj file, y/n:
n
Welcome!--by Qiang Liu @Institute of Nuclear Physics and Chemistry, China Academy of Engineering Physics; Email: liubinqiang@163.com
Repository adress of the Source code on github: https://github.com/dadaoqiuzhi/RMD_Digging


atom_num_autoread.m
BO_match.m
BOele_rank.m
BOentity.m
BOid_element.m
BOidentify.m
bondorder_deepmining.m
bonds_analysis_speedup.m
car_mdf_filemaker.m
cellrowcol_del.m
charnum_match.m
chemi_mechanism.m
CO-239600-300000-1-product-base10.ca
CO-239600-300000-1-reactant-base10.c...
CO-239600-300000-1-reactantBO.txt
CO-any-300000-1-productBO.txt
eleme_molecule.m
forcefield.m
inspectBOinform.m
lammpstrj_analysis.m
membercheck.m
react_blocklocate.m
seekBOinform.m
species_analysis.m
species_analysis_capture.m
species_bonds_lammpstrj_framecheck.m
chemi_mechanism.m (脚本)
scrit file name chemi_mechanism2
Repository adress of the Source code on github: https://github.com/dadaoqiuzhi/RMD_Digging
References: 1.Fuel 287 (2021) 119484
41302. 3.ACS Appl. Mat. Interfaces 2022, 14.(4),
4.ACS Materials Letters 2023, 2174-2
Species analasis is finished, please input the timestep of the interwsted species.
确定
species_analysis_capture is running,
User can confirm the timestep by species_analysis, species_capture program.
Please input the timestep expected to be exported, seperated by white space:
300000
Check the consistency of frame No. between species.* (outputnew), bonds.* and lammpstrj.* files.
Sometimes energy minimization causes the inconsistent record problem
species_bonds_lammpstrj_framecheck is running, please wait...
Firstly, check frame No. between bonds.* and species.* files, please wait...
The appointed frame No. in bonds.* file (5 in total) has been read
Then, check frame No. between lammpstrj.* and species.* files, please wait...
The appointed frame No. in lammpstrj.* file (5 in total) has been read
5 frame No. in the bonds.*, lammpstrj.* and species.* files are as follows:
0 100000 400
400 100400 800
800 100800 1200
1200 101200 1600
1600 101600 2000
As above, frame No. in the bonds.*, lammpstrj.* and species.* files are inconsistent,
please treat the frame No. of species.* file (outputnew) by the following method:
Please treat the frame No. of species.* file (outputnew) by the following method:
1.Substract an integer 2.Add an integer 3.No action
3


atom_num_autoread.m
BO_match.m
BOele_rank.m
BOentity.m
BOid_element.m
BOidentify.m
bondorder_deepmining.m
bonds_analysis_speedup.m
car_mdf_filemaker.m
cellrowcol_del.m
charnum_match.m
chemi_mechanism.m
CO-239600-300000-1-product-base10.ca
CO-239600-300000-1-reactant-base10.c...
CO-239600-300000-1-reactantBO.txt
CO-any-300000-1-productBO.txt
eleme_molecule.m
forcefield.m
inspectBOinform.m
lammpstrj_analysis.m
membercheck.m
react_blocklocate.m
seekBOinform.m
species_analysis.m
species_analysis_capture.m
species_bonds_lammpstrj_framecheck.m
chemi_mechanism.m (脚本)
scrit file name chemi_mechanism2
1600 101600 2000
Please treat the frame No. of species.* file (outputnew) by the following method
确定
As above, frame No. in the bonds.*,
please treat the frame No. of specie
Please treat the frame No. of specie
1.Substract an integer 2.Add an inte
3
Please input the unsigned number, eg.8, 29, or 377, etc.:
0
No action is carried out
The frame No. problem in species.* file (outputnew) has been disposed
Target species CO amount to 3 in total (original number 0)
Limit the exportation of reactants-products? If yes, program will be forced to terminate. y/n:
y
Please input limited file number, <3:
1
For high efficiency, only search the trajectory frame with changed species number
Limit the trajectory frame to be searched? For method 2 the searched trajectory frame with degressive timestep.
For method 4 is on the contrary. y/n:
y
Abstract BO information of target trajectory
Welcome!--by Qiang Liu @Institute of Nuclear Physics and Chemistry, China Academy of Engineering Physics; Email: liubinqiang@163.com
References: 1.Fuel 287 (2021) 119484. 2.ACS Appl. Mat. Interfaces 13(34) (2021) 41287-41302. 3.ACS Appl. Mat. Interfaces 2022, 14.(4),


名称
atom_num_autoread.m
BO_match.m
BOele_rank.m
BOentity.m
BOid_element.m
BOidentify.m
bondorder_deepmining.m
bonds_analysis_speedup.m
car_mdf_filemaker.m
cellrowcol_del.m
charnum_match.m
chemi_mechanism.m
CO-239600-300000-1-product-base10.ca
CO-239600-300000-1-reactant-base10.c...
CO-239600-300000-1-reactantBO.txt
CO-any-300000-1-productBO.txt
eleme_molecule.m
forcefield.m
inspectBOinform.m
lammpstrj_analysis.m
membercheck.m
react_blocklocate.m
seekBOinform.m
species_analysis.m
species_analysis_capture.m
species_bonds_lammpstrj_framecheck.m
chemi_mechanism.m (脚本)
scrit file name chemi_mechanism2
命令行窗口
References: 1.Fuel 287 (2021) 119484. 2.ACS Appl. Mat. Interfaces 13(34) (2021) 41287-41302. 3.ACS Appl. Mat. Interfaces 2022, 14.(4),
4.ACS Materials Letters 2023, 2174-2188. More work is coming!
bonds_analysis_speedup is running, please wait...
bonds_analysis_speedup is successfully finished, BO information is saved in bondoutdata, search line number is recorded in readline
Welcome!--by Qiang Liu @Institute of Nuclear Physics and Chemistry, China Academy of Engineering Physics; Email: liubinqiang@163.com
github仓库地址: https://github.com/dadaoqiuzhi/RMD_Digging
species_analysis is running, Please wait...
bondorder_deepmining is successfully finished
Molecular formula is saved in tarelenummatch, corresponding BO information is saved in tarBOinform
Target species number(3) is consistent with that of species recordation(3)
Successfully delete the unexpected BO information in tarBOinform
lammpstrj_analysis is running, please wait...
lammpstrj_analysis is successfully finished
Atom coordinate of the target trajecrory is saved in trjdata
New frame is found 289600, species number 2 is less than that of last frame 300000, which has 3


atom_num_autoread.m
BO_match.m
BOele_rank.m
BOentity.m
BOid_element.m
BOidentify.m
bondorder_deepmining.m
bonds_analysis_speedup.m
car_mdf_filemaker.m
cellrowcol_del.m
charnum_match.m
chemi_mechanism.m
CO-239600-300000-1-product-base10.ca
CO-239600-300000-1-reactant-base10.c...
CO-239600-300000-1-reactantBO.txt
CO-any-300000-1-productBO.txt
eleme_molecule.m
forcefield.m
inspectBOinform.m
lammpstrj_analysis.m
membercheck.m
react_blocklocate.m
seekBOinform.m
species_analysis.m
species_analysis_capture.m
species_bonds_lammpstrj_framecheck.m
chemi_mechanism.m (脚本)
scrit file name chemi_mechanism2
New frame is found 289600, species number 2 is less than that of last frame 300000, which has 3
Welcome!--by Qiang Liu @Institute of Nuclear Physics and Chemistry, China Academy of Engineering Physics; Email: liubinqiang@163.com
References: 1.Fuel 287 (2021) 119484. 2.ACS Appl. Mat. Interfaces 13(34) (2021) 41287-41302. 3.ACS Appl. Mat. Interfaces 2022, 14.(4),
4.ACS Materials Letters 2023, 2174-2188. More work is coming!
bonds_analysis_speedup is running, please wait...
bonds_analysis_speedup is successfully finished, BO information is saved in bondoutdata, search line number is recorded in readline
Welcome!--by Qiang Liu @Institute of Nuclear Physics and Chemistry, China Academy of Engineering Physics; Email: liubinqiang@163.com
github仓库地址: https://github.com/dadaoqiuzhi/RMD_Digging
species_analysis is running, Please wait...
bondorder_deepmining is successfully finished
Molecular formula is saved in tarelenummatch, corresponding BO information is saved in tarBOinform
Group 1 reactants(289600)-products(300000) hit nothing,already hit 0 groups, continue to search the next frame of reactants
New frame is found 239600, species number 1 is less than that of last frame 289600, which has 2


atom_num_autoread.m
BO_match.m
BOele_rank.m
BOentity.m
BOid_element.m
BOidentify.m
bondorder_deepmining.m
bonds_analysis_speedup.m
car_mdf_filemaker.m
cellrowcol_del.m
charnum_match.m
chemi_mechanism.m
CO-239600-300000-1-product-base10.ca
CO-239600-300000-1-reactant-base10.c...
CO-239600-300000-1-reactantBO.txt
CO-any-300000-1-productBO.txt
eleme_molecule.m
forcefield.m
inspectBOinform.m
lammpstrj_analysis.m
membercheck.m
react_blocklocate.m
seekBOinform.m
species_analysis.m
species_analysis_capture.m
species_bonds_lammpstrj_framecheck.m
chemi_mechanism.m (脚本)
scrit file name chemi_mechanism2
bonds_analysis_speedup is running, please wait...
bonds_analysis_speedup is successfully finished, BO information is saved in bondoutdata, search line number is recorded in readline
Welcome!--by Qiang Liu @Institute of Nuclear Physics and Chemistry, China Academy of Engineering Physics; Email: liubinqiang@163.com
github仓库地址: https://github.com/dadaoqiuzhi/RMD_Digging
References: 1.Fuel 287 (2021) 119484. 2.ACS Appl. Mat. Interfaces 13(34) (2021) 41287-41302. 3.ACS Appl. Mat. Interfaces 2022, 14.(4),
4.ACS Materials Letters 2023, 2174-2188. More work is coming!
species_analysis is running, Please wait...
bondorder_deepmining is successfully finished
Molecular formula is saved in tarelenummatch, corresponding BO information is saved in tarBOinform
Now is searching group 2 reactants(239600)-products(300000), this is the hit group, already hits 1 groups(including this group),
3 groups in total to be processed, limited to export 1 groups
Welcome!--by Qiang Liu @Institute of Nuclear Physics and Chemistry, China Academy of Engineering Physics; Email: liubinqiang@163.com
References: 1.Fuel 287 (2021) 119484. 2.ACS Appl. Mat. Interfaces 13(34) (2021) 41287-41302. 3.ACS Appl. Mat. Interfaces 2022, 14.(4),
4.ACS Materials Letters 2023, 2174-2188. More work is coming!
lammpstrj_analysis is running, please wait...
lammpstrj_analysis is successfully finished
Atom coordinate of the target trajecrory is saved in trjdata
Welcome!--by Qiang Liu @Institute of Nuclear Physics and Chemistry, China Academy of Engineering Physics; Email: liubinqiang@163.com

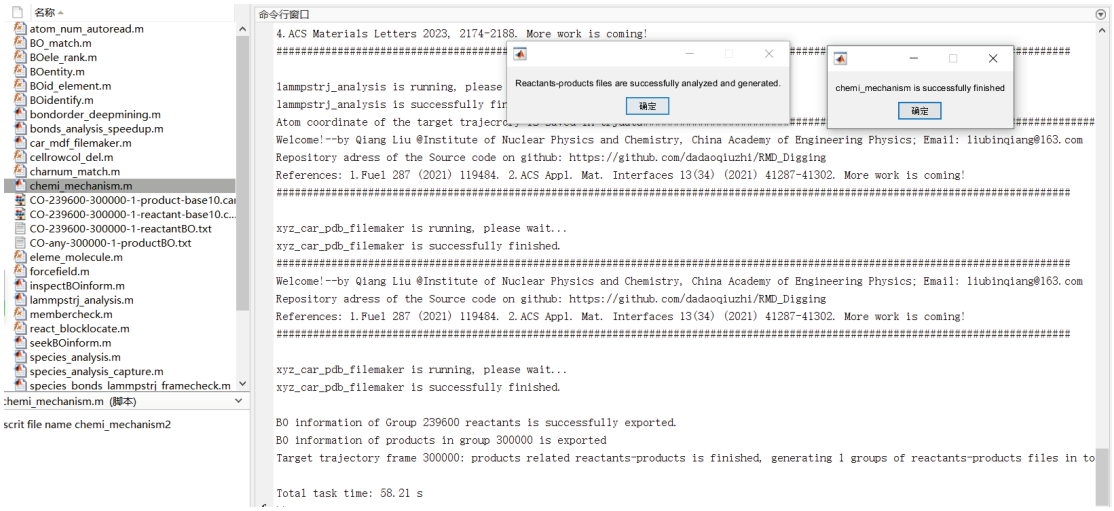


Figure 12 Examples of running chemi_mechanism code of the chemi_mechanism module.

Here a brief introduction on the further analysis and visualization of the generated files is given. The generated files in the last tutor are shown in Figure 13. We can see the CO2 product in the 348000 frame can be formed by different reaction pathways, indicated by the reactants that occurred in the timestep of 262000, 299200, 314000, etc. The number after 348000 denotes the number of reactions in this frame that is related to the formation of the interesting product. The files with the file extension "car" can be visualized and analyzed by Materials Studio. For each product-reactant pair, the bond information is exported for subsequent use (see **-reactantBO.txt*). The bond information of the interesting product specified by the users in the very beginning is also exported, whose name has "any" character (see **-productBO.txt*). One can combine these files (**-reactantBO.txt, *-productBO.txt* and **.car*) with the knowledge of software operation to analyze the detailed reaction pathways.

Figure 13 Files generated by task 2 in the chemi_mechanism code of the chemi_mechanism module.

3 Reference[1-7]

(1) Liu, Q.; Liu, S.; Lv, Y.; Hu, P.; Huang, Y.; Kong, M.; Li, G. Atomic-scale insight into the pyrolysis of polycarbonate by ReaxFF-based reactive molecular dynamics simulation. *Fuel* **2021,** *287*, 119484, DOI: https://doi.org/10.1016/j.fuel.2020.119484.

(2) Liu, Q.; Huang, W.; Liu, B.; Wang, P.-C.; Chen, H.-B. Gamma Radiation Chemistry of Polydimethylsiloxane Foam in Radiation-Thermal Environments: Experiments and Simulations. *ACS Appl. Mat. Interfaces* **2021,** *13* (34), 41287-41302, DOI: 10.1021/acsami.1c10765.

(3) Chenoweth, K.; Cheung, S.; van Duin, A. C.; Kober, E. M. Simulations on the thermal decomposition of a poly(dimethylsiloxane) polymer using the ReaxFF reactive force field. *J. Am. Chem. Soc.* **2005,** *127* (19), 7192-202.

(4) Vashisth, A.; Ashraf, C.; Zhang, W.; Bakis, C. E.; Duin, A. C. T. V. Accelerated ReaxFF Simulations for Describing the Reactive Cross-Linking of Polymers. *J. Phys. Chem. A 122* (32), acs.jpca.8b03826-.

(5) Yoon, K.; Rahnamoun, A.; Swett, J. L.; Iberi, V.; Cullen, D. A.; Vlassiouk, I. V.; Belianinov, A.; Jesse, S.; Sang, X.; Ovchinnikova, O. S.; Rondinone, A. J.; Unocic, R. R.; van Duin, A. C. T. Atomistic-Scale Simulations of Defect Formation in Graphene under Noble Gas Ion Irradiation. *ACS Nano* **2016,** *10* (9), 8376-8384, DOI: 10.1021/acsnano.6b03036.

(6) Liang, T.; Yun, K. S.; Cheng, Y. T.; Yilmaz, D. E.; Vishnu, K. G.; Verners, O.; Zou, C.; Phillpot, S. R.; Sinnott, S. B.; Duin, A. C. T. V. Reactive Potentials for Advanced Atomistic Simulations. *Annual Review of Materials Research* **2013,** *43* (43), 109-129.

(7) Farah, K.; Müller-Plathe, F.; Böhm, M. C. Classical reactive molecular dynamics implementations: state of the art. *Chemphyschem* **2012,** *13* (5), 1127-1151.