

Применение силового поля ANI-2x для моделирования полифениленсульфида методом классической молекулярной динамики

Л. О. Виляев, Д. В. Завьялов

Волгоградский государственный технический университет

Аннотация: В работе исследуется применимость конкретной модели силовых полей – ANI-2x – к полимеру полифениленсульфиду. Приводится порядок проведённых действий и, в качестве результатов, сравнивается радиальная функция распределения атомов серы в системах с разными температурами.

Ключевые слова: полифениленсульфид, классическая молекулярная динамика, машинное обучение, силовое поле, потенциал, полимер, численное моделирование, модель.

Введение

За последнее десятилетие проблема восстановления межатомного потенциала в классической молекулярной динамике (МД) получила значительный толчок благодаря методам, основанным на машинном обучении [1-3]. Конструирование таких силовых полей стало новым направлением исследований, ставшего популярным в вычислительном материаловедении, вычислительной химии, вычислительной физике.

Помимо моделей, создаваемых под конкретную систему, или узкий набор систем, создаются также более универсальные модели потенциалов. Они представляют возможность проводить МД гораздо более широкого спектра молекулярных систем. Некоторые из них: ANI-2x [4], AIMNet2 [5] PhysNet [6], TeaNet [7], SchNet [8]. Также существуют и разрабатываются программные пакеты и библиотеки для использования различных моделей силовых полей: OpenMM [9], Snap ML [10], TorchMD [11], TorchANI [12] и другие.

Среди органических соединений представляют интерес полимеры. Из-за сложной молекулярной структуры, моделирование даже классической молекулярной динамикой полимеров является как минимум сложной вычислительной задачей. В связи с этим, использование машинного

обучения для восстановления потенциалов в исследовании их свойств, напрашивается само собой.

Учитывая актуальность и преимущества подходов, основанных на машинном обучении, мы решили проверить их адекватность в отношении полимера. Для этого мы проведём классическую МД полифениленсульфида (ПФС), полимера, звено которого состоит из бензольного кольца и соединённого с ним атома серы (рис. 1).

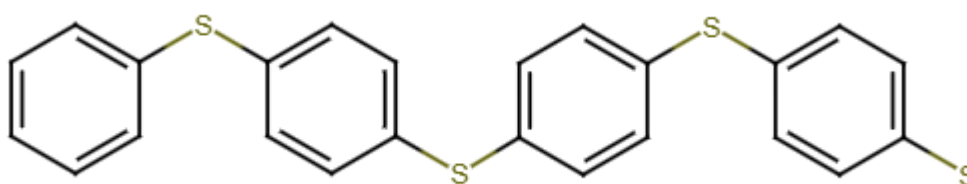


Рис. 1. – Четыре звена цепи ПФС

Порядок выполнения

Первая часть работы проводится с использованием пакета RadonPy [13]. Первым делом необходимо задать формулу SMILES, для ПФС она выглядит так: `"*c1ccc(S*)cc1"`. Звёздочки обозначают точки соединения с соседними звеньями цепи. Строка SMILES посылается в функцию создания объекта Mol, после чего производится поиск конформации (структуры).

```
from radonpy.core import utils
from radonpy.sim import qm
from radonpy.ff.gaff2_mod import GAFF2_mod

smiles = '*c1ccc(S*)cc1'
ff = GAFF2_mod()
mol = utils.mol_from_smiles(smiles)
mol, energy = qm.conformation_search(mol, ff=ff, work_dir=work_dir,
psi4_omp=omp_psi4, mpi=mpi, omp=omp, memory=mem, log_name='monomer1')
```

Затем рассчитываются электронные свойства звена полимера и конечного узла. Конечный узел – это то, чем закроется первая и последняя точки соединения в цепи полимера. Конечный узел задаётся также строкой SMILES и в данном случае выбран `"*C"`.

```
qm.assign_charges(mol, charge='RESP', opt=False, work_dir=work_dir,  
omp=omp_psi4, memory=mem, log_name='monomer1')  
qm_data = qm.sp_prop(mol, opt=False, work_dir=work_dir, omp=omp_psi4,  
memory=mem, log_name='monomer1')  
polar_data = qm.polarizability(mol, opt=False, work_dir=work_dir,  
omp=omp_psi4, memory=mem, log_name='monomer1')  
ter_smiles = '*C'  
ter = utils.mol_from_smiles(ter_smiles)  
qm.assign_charges(ter, charge='RESP', opt=True, work_dir=work_dir,  
omp=omp_psi4, memory=mem, log_name='ter1')
```

Следующим этапом стоит создание полимерной цепи. Создаётся цепь, состоящая из 1000 звеньев. После чего, полимерной цепи присваивается силовое поле GAFF2_mod.

```
from radonpy.core import poly  
  
dp = poly.calc_n_from_num_atoms(mol, 1000, terminal1=ter)  
homopoly = poly.polymerize_rw(mol, dp, tacticity='atactic')  
homopoly = poly.terminate_rw(homopoly, ter)  
result = ff.ff_assign(homopoly)
```

Далее, генерируется область симуляции, содержащая несколько цепей, созданных ранее. Мы выбрали область, содержащую 10 полимерных цепочек, с плотностью 0,05.

```
ac = poly.amorphous_cell(homopoly, 10, density=0.05)
```

Заключительным этапом в этой части является трёхэтапное уравнивание и проверка релаксации.

```
eqmd = eq.EQ21step(ac, work_dir=work_dir)  
ac = eqmd.exec(temp=temp, press=1.0, mpi=mpi, omp=omp, gpu=gpu)  
analy = eqmd.analyze()  
prop_data = analy.get_all_prop(temp=temp, press=1.0, save=True)  
result = analy.check_eq()
```

Если проверка релаксации провалена, то запускается дополнительное уравнивание: повторение до 4 раз NpT симуляции на 5 нс при 300 К и 1 атмосфере.

```
for i in range(4):  
    if result: break  
    eqmd = eq.Additional(ac, work_dir=work_dir)  
    ac = eqmd.exec(temp=temp, press=press, mpi=mpi, omp=omp, gpu=gpu)
```

```
analy = eqmd.analyze()  
prop_data = analy.get_all_prop(temp=temp, press=press, save=True)  
result = analy.check_eq()
```

Второй частью работы является проведение МД с использованием пакета OpenMM-torch – плагина для OpenMM [9]. Здесь будет использована модель потенциалов ANI-2x [4]. Развивая свою предыдущую модель [14], авторы реализовали поддержку семи элементов: H, C, N, O, F, Cl, S. По сравнению с проверенными методами квантовой механики, ANI-2x показывает точность, аналогичную DFT (теория функционала плотности).

Для использования OpenMM необходимо два входных файла: .pdb файл, содержащий названия атомов, координаты атомов, виды атомов, номер звена; .xml файл силового поля, содержащий связи атомов в повторяющемся звене и в конечных узлах. Первый файл создается внутри RadonPy с помощью встроенной функции конвертации Mol-объекта в .pdb файл – `utils.MolToPDBFile()`. Полученный файл форматируется с помощью рукописной программы так, чтобы в одном звене не было одинаковых названий атомов. Имена атомов выбраны произвольным образом.

Второй файл создается другой рукописной программой из .pdb файла.

Созданные файлы загружаются в OpenMM, и создается система, названная "pps", на основе топологии .pdb файла. Также из .pdb файла извлекается список атомных чисел.

```
from openmm.app import PDBFile, ForceField  
  
pdb = PDBFile('333K_edited.pdb')  
ff = ForceField('fakeff.xml')  
pps = ff.createSystem(pdb.topology)  
atomic_numbers = [atom.element.atomic_number for atom in pdb.topology.atoms()]
```

Затем создается класс потенциалов с помощью TorchANI [12]. Выбранную модель ANI-2x можно ускорить с помощью пакета NNPOps.

```
import torch as pt  
from torchani.models import ANI2x  
from NNPOps import OptimizedTorchANI
```

```
class NNP(pt.nn.Module):  
  
    def __init__(self, atomic_numbers):  
        super().__init__()  
        self.atomic_numbers = pt.tensor(atomic_numbers).unsqueeze(0)  
        self.model = ANI2x(periodic_table_index=True)  
        self.model = OptimizedTorchANI(self.model, self.atomic_numbers)  
  
    def forward(self, positions):  
        positions = positions.unsqueeze(0).float() * 10 # nm --> Å  
        result = self.model((self.atomic_numbers, positions))  
        energy = result.energies[0] * 2625.5 # Hartree --> kJ/mol  
        return energy  
  
nnp = NNP(atomic_numbers)
```

Созданная модель потенциалов сохраняется в файл. В последующих моделированиях шаг создания потенциала опускается, поскольку модель уже ГОТОВА.

```
pt.jit.script(nnp).save('model.pt')
```

Сохранённая модель силового поля загружается с помощью OpenMM-torch и добавляется в созданную в начале систему "pps".

```
from openmmtorch import TorchForce  
  
force = TorchForce('model.pt')  
pps.addForce(force)
```

Следующим шагом является подготовка симуляции. Сначала создаётся интегратор, в нашем случае, интегратор Ланжевена.

```
import sys  
from openmm import LangevinMiddleIntegrator, Platform  
from openmm.app import Simulation, StateDataReporter  
from openmm.unit import kelvin, picosecond, femtosecond  
  
temperature = 333 * kelvin  
frictionCoeff = 1 / picosecond  
timeStep = 1 * femtosecond  
integrator = LangevinMiddleIntegrator(temperature, frictionCoeff, timeStep)
```

После этого задаётся симуляция с позициями, скоростями и интегратором.

```
simulation = Simulation(pdb.topology, pps, integrator)
```

```
simulation.context.setPositions(pdb.positions)
```

Заключительным шагом является запуск симуляции на количество шагов, указанных в интеграторе. Здесь для всех симуляций мы выбрали 100 пс (100000 фс).

```
simulation.step(100000)
```

Результаты моделирования

По результатам моделирования для температур 333 К, 353 К и 373 К были получены файлы с дампами траекторий, по которым можно производить расчёты некоторых тепловых свойств системы. Температуры моделирования были выбраны в районе экспериментальной точки стеклования полимера с тем, чтобы проверить применимость поля ANI-2x с помощью анализа поведения модели при переходе через точку стеклования. Самой простой характеристикой, изменяющейся при стекловании, является удельный объём (или, что по сути тоже самое, плотность) системы. Обычно полимеры при температурах выше температуры стеклования находятся в пластичном состоянии, а при температурах ниже температуры стеклования – в твёрдом. И плотность в пластичном состоянии меньше, чем в твёрдом. Если поле ANI-2x приемлемо описывает взаимодействие атомов в полимере, то такое же поведение при переходе через точку стеклования должно быть и при моделировании.

Так как при моделировании использовалась одна и та же система (менялись только температуры), то её масса не менялась. Значит, достаточно проследить, как меняется объём системы – он должен расти при повышении температуры. В качестве критерия для оценки поведения объёма мы решили взять радиальную функцию распределения (РФР) [15] атомов серы, а не сам непосредственно объём системы, так как корректно посчитать объём для клубка полимерных молекул не является такой же тривиальной задачей, как для однородных материалов, состоящих из небольших молекул.

РФР определяет вероятность обнаружения частицы на расстоянии r от другой, меченой частицы. Задаётся формулой:

$$g(r) = \frac{dn_r}{dV_r \cdot \rho} \approx \frac{dn_r}{4\pi r^2 dr \cdot \rho},$$

где dn_r – количество частиц в сферической оболочке толщиной dr , $dV_r \approx 4\pi r^2 dr$ – объём оболочки, ρ – средняя плотность всей системы.

Радиальная функция распределения атомов серы, полученная по результатам моделирования для температур 333 К, 353 К и 373 К, изображена на рис. 2.

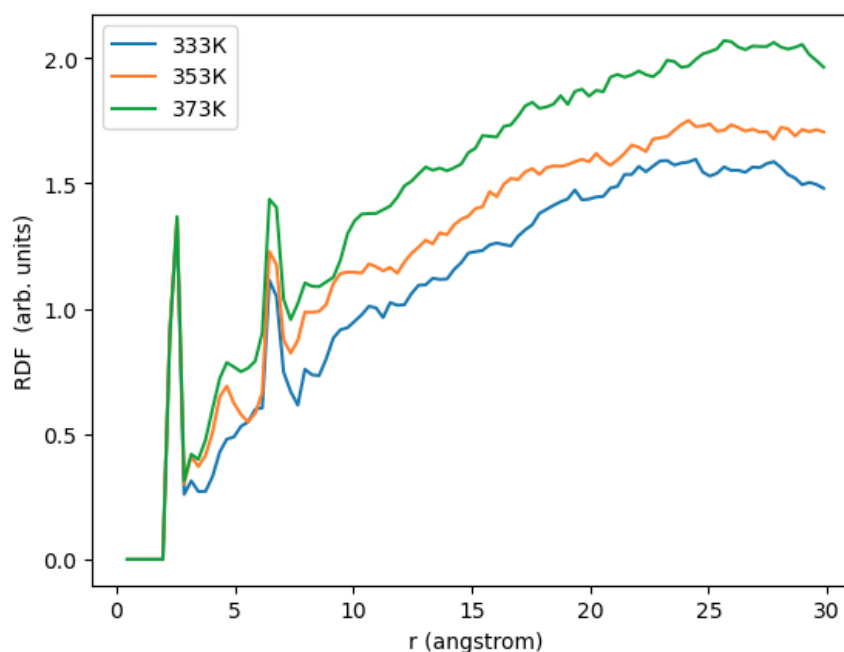


Рис. 2. – РФР атомов серы

Из рисунка видно, что с ростом температуры всё большая часть атомов располагается на больших расстояниях друг от друга, то есть объём должен расти. Таким образом можно утверждать, что качественно поле ANI-2x верно описывает ситуацию при переходе через температуру стеклования. Для её количественного определения по результатам моделирования необходим, например, расчёт теплоёмкости в зависимости от температуры (так называемый калориметрический метод), что требует большого количества вычислительных ресурсов и будет продолжением этой работы.

Литература (References)

1. Pinheiro M., Ge F., Ferré N., Dral P. O., Barbatti M. Choosing the right molecular machine learning potential // *Chemical Science*. – 2021. – Vol. 12, № 43. – pp. 14396-14413.

2. Mueller T., Hernandez A., Wang C. Machine learning for interatomic potential models // *The Journal of Chemical Physics*. – 2020. – Vol. 152, № 5. – pp. 050902.

3. Mishin Y. Machine-learning interatomic potentials for materials science // *Acta Materialia*. – 2021. – Vol. 214. – P. 116980.

4. Devereux C., Smith J. S., Huddleston K. K., Barros K., Zubatyuk R., Isayev O., Roitberg A. E. Extending the Applicability of the ANI Deep Learning Molecular Potential to Sulfur and Halogens // *Journal of Chemical Theory and Computation*. – 2020. – Vol. 16, № 7. – pp. 4192-4202.

5. Anstine D., Zubatyuk R., Isayev O. AIMNet2: A Neural Network Potential to Meet your Neutral, Charged, Organic, and Elemental-Organic Needs//. – 2023.doi.org/10.26434/chemrxiv-2023-296ch.

6. Unke O. T., Muwly M. PhysNet: A Neural Network for Predicting Energies, Forces, Dipole Moments, and Partial Charges // *Journal of Chemical Theory and Computation*. – 2019. – Vol. 15, № 6. – pp. 3678-3693.

7. Takamoto S., Izumi S., Li J. TeaNet: Universal neural network interatomic potential inspired by iterative electronic relaxations // *Computational Materials Science*. – 2022. – Vol. 207. – pp. 111280.

8. Schütt K. T., Sauceda H. E., Kindermans P. J., Tkatchenko A., Müller K. R. SchNet – A deep learning architecture for molecules and materials // *The Journal of Chemical Physics*. – 2018. – Vol. 148, № 24. – pp. 241722.

9. Eastman P., Galvelis R., Peláez R. P., Abreu C. R. A., Farr S. E., Gallicchio E., Gorenko A., Henry M. M., Hu F., Huang J., Krämer A., Michel J., Mitchell J. A., Pande V. S., Rodrigues J. P., Rodriguez-Guerra J., Simmonett A.



C., Singh S., Swails J., Turner P., Wang Y., Zhang I., Chodera J. D., De Fabritiis G., Markland T. E. OpenMM 8: Molecular Dynamics Simulation with Machine Learning Potentials // The Journal of Physical Chemistry B. – 2024. – Vol. 128, № 1. – pp. 109-116.

10. Mendler-Dünner C., Parnell T., Sarigiannis D., Ioannou N., Anghel A., Pozidis H. Snap ML: A Hierarchical Framework for Machine Learning // ArXiv. – 2018. doi.org/10.48550/arXiv.1803.06333.

11. Doerr S., Majewski M., Pérez A., Krämer A., Clementi C., Noe F., Giorgino T., De Fabritiis G. TorchMD: A Deep Learning Framework for Molecular Simulations // Journal of Chemical Theory and Computation. – 2021. – Vol. 17, № 4. – pp. 2355-2363.

12. Gao X., Ramezanghorbani F., Isayev O., Smith J. S., Roitberg A. E. TorchANI: A Free and Open Source PyTorch-Based Deep Learning Implementation of the ANI Neural Network Potentials // Journal of Chemical Information and Modeling. – 2020. – Vol. 60, № 7. – pp. 3408-3415.

13. Hayashi Y., Shiomi J., Morikawa J., Yoshida R. RadonPy: automated physical property calculation using all-atom classical molecular dynamics simulations for polymer informatics // npj Computational Materials. – 2022. – Vol. 8, № 1. – P. 222.

14. Smith J. S., Isayev O., Roitberg A. E. ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost // Chemical Science. – 2017. – Vol. 8, № 4. – pp. 3192-3203.

15. Levine B. G., Stone J. E., Kohlmeyer A. Fast analysis of molecular dynamics trajectories with graphics processing units – Radial distribution function histogramming // Journal of Computational Physics. – 2011. – Vol. 230, № 9. – pp. 3556-3569.

Дата поступления: 11.03.2024

Дата публикации: 18.04.2024
