TU Delft

Delft University of Technology

Document Version
Final published version

Licence
CC BY

Citation (APA)
Ran, Y. A., Sharma, S., Balestra, S. R. G., Li, Z., Calero, S., Vlugt, T. J. H., Snurr, R. Q., & Dubbeldam, D. (2024). RASPA3: A Monte Carlo code for computing adsorption and diffusion in nanoporous materials and thermodynamics properties of fluids. *Journal of Chemical Physics*, *161*(11), Article 114106. https://doi.org/10.1063/5.0226249

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
In case the licence states “Dutch Copyright Act (Article 25fa)”, this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership.
Unless copyright is transferred by contract or statute, it remains with the copyright holder.

Sharing and reuse
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.

The Journal of Chemical Physics

RESEARCH ARTICLE | SEPTEMBER 16 2024

RASPA3: A Monte Carlo code for computing adsorption and diffusion in nanoporous materials and thermodynamics properties of fluids

Y. A. Ran ; S. Sharma ; S. R. G. Balestra ; Z. Li ; S. Calero ; T. J. H. Vlugt ; R. Q. Snurr ; D. Dubbeldam

Check for updates

J. Chem. Phys. 161, 114106 (2024)
https://doi.org/10.1063/5.0226249

View Online
Export Citation

Articles You May Be Interested In

Efficient configurational-bias Monte-Carlo simulations of chain molecules with “swarms” of trial configurations

J. Chem. Phys. (August 2018)

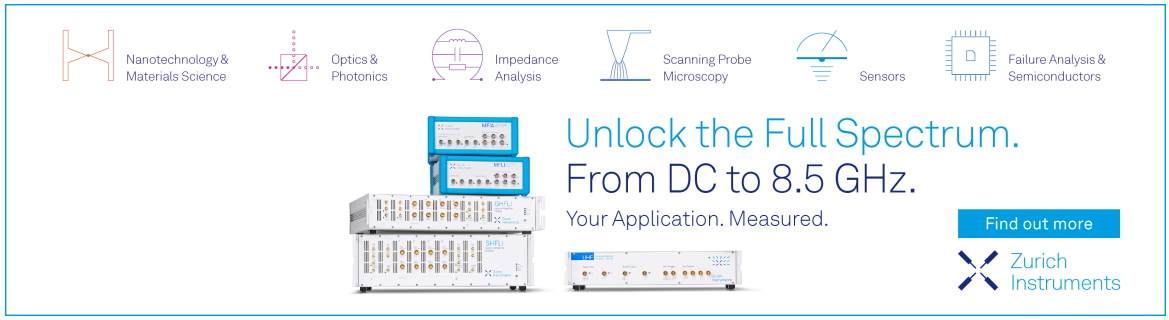
Efficient global biopolymer sampling with end-transfer configurational bias Monte Carlo

J. Chem. Phys. (January 2007)

Improving the efficiency of configurational-bias Monte Carlo: A density-guided method for generating bending angle trials for linear and branched molecules

J. Chem. Phys. (August 2014)

30 September 2024 11:38:08



AIP Publishing

The Journal of Chemical Physics ARTICLE pubs.aip.org/aip/jcp

RASPA3: A Monte Carlo code for computing adsorption and diffusion in nanoporous materials and thermodynamics properties of fluids

Cite as: J. Chem. Phys. 161, 114106 (2024); doi: 10.1063/5.0226249
Submitted: 30 June 2024 • Accepted: 2 September 2024 •
Published Online: 16 September 2024

View Online Export Citation CrossMark

Y. A. Ran,[1] S. Sharma,[2] S. R. G. Balestra,[3,a)] Z. Li,[4] S. Calero,[5] T. J. H. Vlugt,[2] R. Q. Snurr,[4] and D. Dubbeldam[1,b)]

AFFILIATIONS

[1] Van 't Hoff Institute for Molecular Sciences, University of Amsterdam, Science Park 904, 1098 XH Amsterdam, The Netherlands
[2] Engineering Thermodynamics, Process and Energy Department, Faculty of Mechanical, Maritime and Materials Engineering, Delft University of Technology, Leeghwaterstraat 39, 2628CB Delft, The Netherlands
[3] Centro de Nanociencias y Tecnologías Sostenibles (CNATS), Universidad Pablo de Olavide, ES-41013 Sevilla, Spain
[4] Department of Chemical and Biological Engineering, Northwestern University, 2145 Sheridan Road, Evanston, Illinois 60208, USA
[5] Department of Applied Physics, Eindhoven University of Technology, 5600MB Eindhoven, The Netherlands

Note: This paper is part of the JCP Special Topic on Monte Carlo Methods, 70 years after Metropolis *et al.* (1953).
[a)] **Current address:** Departamento de Física Atómica, Molecular y Nuclear, Área de Física Teórica, Universidad de Sevilla, Avenida de Reina Mercedes s/n, 41012 Sevilla, Spain.
[b)] **Author to whom correspondence should be addressed:** d.dubbeldam@uva.nl.
URL: https://www.uva.nl/profiel/d/u/d.dubbeldam/d.dubbeldam.html

ABSTRACT

We present RASPA3, a molecular simulation code for computing adsorption and diffusion in nanoporous materials and thermodynamic and transport properties of fluids. It implements force field based classical Monte Carlo/molecular dynamics in various ensembles. In this article, we introduce the new additions and changes compared to RASPA2. RASPA3 is rewritten from the ground up in C++23 with speed and code readability in mind. Transition-matrix Monte Carlo is added to compute the density of states and free energies. The Monte Carlo code for rigid molecules is based on quaternions, and the atomic positions needed in the energy evaluation are recreated from the center of mass position and quaternion orientation. The expanded ensemble methodology for fractional molecules, with a scaling parameter λ between 0 and 1, now also keeps track of analytic expressions of $\mathrm{d}U/\mathrm{d}\lambda$, allowing independent verification of the chemical potential using thermodynamic integration. The source code is freely available under the MIT license on GitHub. Using this code, we compare four Monte Carlo (MC) insertion/deletion techniques: unbiased Metropolis MC, Configurational-Bias Monte Carlo (CBMC), Continuous Fractional Component MC (CFCMC), and CB/CFCMC. We compare particle distribution shapes, acceptance ratios, accuracy and speed of isotherm computation, enthalpies of adsorption, and chemical potentials, over a wide range of loadings and systems, for the grand canonical ensemble and for the Gibbs ensemble.

© 2024 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/). https://doi.org/10.1063/5.0226249

I. INTRODUCTION

Monte Carlo (MC) codes are rarer than Molecular Dynamics (MD) codes. The MD algorithm is based on Newton's equations of motion, while MC is based on probability distributions.[1,2] MC is therefore limited to static properties on the one hand, but on the other hand, it has considerable opportunities to sample system states more efficiently than MD. MC can be easily applied to a wide variety of ensembles, such as the NPT-ensemble to compute fluid properties, the Grand-Canonical (GC) ensemble to compute adsorption, and the Gibbs ensemble to compute Vapor–Liquid Equilibrium (VLE).[3,4] MC is also very suitable for advanced biased sampling and extended ensembles, using techniques such as Configurational-Bias MC (CBMC),[4] Continuous Fractional

30 September 2024 11:38:08

© Author(s) 2024

Component MC (CFCMC),[5–9] and CB/CFCMC.[10,11] Several established MC codes and techniques were reviewed in Ref. 12, but some new codes have appeared in recent years.

GOMC[13] is open-source software for simulating molecular systems using the Metropolis Monte Carlo algorithm. The software has been written in object-oriented C++ and uses OpenMP and NVIDIA CUDA to allow for execution on multicore central processing unit (CPU) and graphics processing unit (GPU) architectures, respectively. GOMC employs the widely used simulation file types (PDB, PSF, CHARMM-style parameter files) and can be used to study vapor–liquid equilibria, adsorption in porous materials, surfactant self-assembly, and condensed phase structure for complex molecules.

A relatively new code is Brick-CFCMC.[9,14,15] This software package is designed for performing force field-based molecular simulations of single- and multi-component fluids using state of the art CFCMC techniques. Various ensembles can be combined: NVT, NPT, the Gibbs ensemble, the reaction ensemble, the grand-canonical ensemble, and the osmotic ensemble. Properties such as excess chemical potentials, fugacity coefficients, partial molar enthalpies, and partial molar volumes can be directly obtained from single simulations. Brick-CFCMC can also perform thermodynamic integration to compute free energies (including ionic systems).

RASPA2[16] is our own open source software package for simulating adsorption and diffusion of molecules in flexible nanoporous materials. The code implements the latest state-of-the-art algorithms for molecular dynamics and Monte Carlo in various ensembles, including symplectic/measure-preserving integrators, Ewald summation, configurational-bias Monte Carlo, continuous fractional component Monte Carlo, reactive Monte Carlo, and Baker's minimization. Applications of RASPA include computing coexistence properties, adsorption isotherms for single and multiple components, self- and collective diffusivities, reaction systems, and visualization. Gowers *et al.*[17] analyzed several Monte Carlo codes, RASPA,[16] Cassandra,[18] DL Monte,[19] Music,[20] and Towhee,[21] and found that their computational performance was quite comparable.

Integrating simulation code with visualization and application-based models is especially interesting for design pipelines. iRASPA[22] is a GPU-accelerated visualization package (with editing capabilities) aimed at materials science. iRASPA is available on macOS, Linux, and Windows and leverages the latest visualization technologies with high performance. iRASPA extensively utilizes GPU computing. For example, void-fractions and surface areas can be computed in a fraction of a second for small/medium structures and in a few seconds for very large unit cells. RUPTURA[23] is a free and open-source software package for (1) the simulation of gas adsorption breakthrough curves; (2) mixture prediction using methods such as the Ideal Adsorption Solution Theory (IAST), segregated-IAST, and explicit isotherm models; and (3) fitting of isotherm models on computed or measured adsorption isotherm data. The RASPA and RUPTURA software enables computation of breakthrough curves directly from adsorption simulations in the grand-canonical ensemble. The combination of the RASPA simulation code with iRASPA visualization and RUPTURA breakthrough modeling is an example of such a begin-to-end design pipeline.

The fundamental concept in adsorption science is the adsorption isotherm, i.e., the equilibrium relation between the quantity of the adsorbed material and the pressure or concentration in the bulk fluid phase at a constant temperature.[24] One of the main applications of RASPA is to compute adsorption isotherms and to understand the atomic-level mechanism of adsorption and separations. This type of information is very hard to obtain experimentally, but in simulations, the positions of the molecules can be examined and "kinks" in isotherms can be explained.[25–27] Figure 1 shows a volume-rendered picture of the density of methane in ZSM-5 (MFI-type) zeolite and that of CO_2 in the Cu-BTC-type metal–organic framework (MOF). MFI is a three-dimensional silicon/oxygen structure with interconnected straight and zigzag ten membered rings.[28] Copper benzene-1,3,5-tricarboxylate (Cu-BTC) is a pocket/channel structure, with a

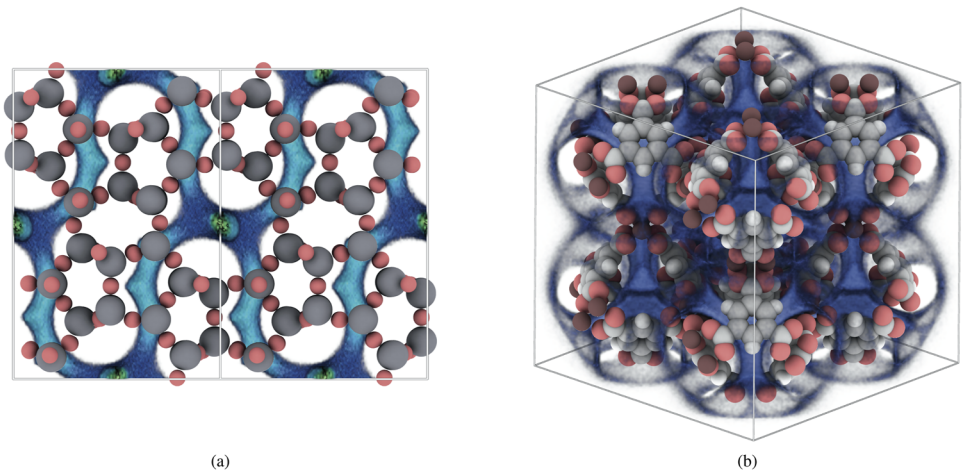


FIG. 1. Volume rendering of the density grid (blue) computed for (a) methane in MFI zeolite at 300 K and 1 bar and (b) CO_2 in Cu-BTC at 300 K and 1 bar. Atoms are colored by atom type, showing oxygen (red), silicon (gray), carbon (light gray), hydrogen (white), and copper (dark red). Images rendered with iRASPA.[22]

30 September 2024 11:38:08

© Author(s) 2024

pore network that has a simple cubic symmetry.[29] The density of adsorbates at a certain temperature and pressure can be analyzed to identify the location of strong adsorption. However, competition and segregation can also be elucidated.[30]

RASPA2 has previously been used extensively in molecular simulation studies of adsorption and diffusion in nanoporous materials.[16] The Grand-Canonical Monte Carlo (GCMC) simulation engine is used to computationally predict the adsorption behavior and can be used stand-alone or in support of performed experimental work. As such, one could validate and explain the adsorption of small gas molecules in synthesized porous materials.[31–37] Structural and thermodynamic analyses can be carried out to analyze molecular (binding) geometries.[38–42] Large-scale screening efforts are supported by the easy featurization of porous materials into surface area, pore volume, and adsorption isotherms.[43–48] More recently, machine learning efforts have been using these featurizations at the core of dataset design for training generative models.[49–52] With all these applications in mind, it is essential to improve RASPA's user experience, computational efficiency, and interface.

In this paper, we present the new version of RASPA. RASPA3 is rewritten from the ground up in C++23, with large improvements in user experience and computational efficiency. In Sec. II, we describe our design decisions and programming details, followed by the new features in Sec. III. In Sec. IV, we validate RASPA3 using adsorption studies of methane and CO_2 in MFI, an equimolar mixture of CO_2 and N_2 in the zeolite MFI, and CO_2 in the Cu-BTC MOF, and VLE Gibbs simulations of methane. We show that RASPA3 reproduces the results from RASPA2, but RASPA3 is a factor of 3–6 times faster compared to RASPA2. Using RASPA3, we will show that the modern CBMC technique is a significant improvement over the standard conventional unbiased Metropolis MC technique and that recent techniques, such as CFCMC and CB/CFCMC, have overcome the downside of standard MC and CBMC, namely the drop in insertion/deletion rates in open ensembles at low temperatures and high densities.

II. DESIGN CHOICES

RASPA3 is redesigned and rewritten from the ground up in C++23. This is a major improvement from the functional C code in RASPA2, both in terms of computational efficiency and in terms of a reduction in technical debt. Technical debt is caused by suboptimally written and designed code that builds over time and severely limits the extensibility of the code. User experience has been improved through a variety of changes in input- and output files as well as the definition of an Application Programming Interface (API) for the user to interact with.

Major improvements in efficiency result from the implementation of C++ code based on composition and value semantics (in contrast to inheritance and reference semantics). Composition is a structural approach in which complex objects are built from simple building blocks. Value semantics avoid sharing mutable state and uphold the independence of values to support local reasoning. References are only used implicitly, at function boundaries, and are never stored. This style of coding has many advantages. Avoiding complex inheritance hierarchies and reducing the need for virtual functions lead to major code simplifications. It encourages code clarity and local reasoning. With composition, components are directly included in an object, removing the necessity for pointers, manual memory allocations, and indirections. This leads to straightforward memory patterns and access safety. Lifetimes of objects are coupled with composition, eliminating the need for explicit memory management. These lead to better compiler optimization and, therefore, performance of the code.

Value semantics make the code much easier to write and much easier to understand. The coding style is essentially C-like but uses convenient C++ abstractions and features from the C++ standard library. The standard library offers containers such as `std::vector` and `std::set`, which avoid any manual memory allocations. Containers offer convenient functionality for memory operations, such as insertion, appending, deletion, swapping, and sorting. The main data item in the code is the `Atom` structure, which contains the position of an atom, the type, the component-identifier, the molecule-identifier, and a group-identifier (for thermodynamic integration). The atoms are stored in a `std::vector<Atom>`. Rigid molecules are handled using center of mass positions and an orientation. We therefore also need a vector of molecules, `std::vector<Molecule>`, containing `Molecule` objects, which hold a center of mass position and a quaternion (to describe the orientations). After generating the `Molecule` structure, the atom positions for the molecule are generated since the computation of the energy is based on potentials that operate on the positions of the atoms. The order in the atom and molecule vectors are framework atoms first and the molecule atoms of the components one after the other.

The C++20 and C++23 language additions have recently offered substantial opportunities to write better code. The first is `std::span`. Spans offer views into data, such as a vector, and are a much better alternative than passing a memory reference and length. Spans have reference semantics but are only used as function arguments and not as data members. It is dangerous to keep references to vector elements around as any memory insertion or deletion to the vector object might cause an internal reallocation. Using spans, the routines of computing the energy can be quite generic and based on spans into the atom- and molecule-vectors. For example, the same functions can be used to compute framework–molecule energies of all components and to compute the interaction of the framework with a single component, as the function takes only the span as argument. Molecule–molecule energy computation differs fundamentally from framework–molecule energy calculations in that for molecule–molecule interactions, self-exclusion must be applied, i.e., a particle does not interact with itself. In C++23, span is extended to the multi-dimensional span `std::mdspan`. Data in a container, such as a vector or an array, can be described as multi-dimensional. This circumvents the manual mapping from 1D to multi-dimensional data. Mdspan is a non-owning view of a contiguous sequence of elements in memory, interpreted as a multi-dimensional array.

The second new addition, included in C++17, is `std::optional`. This is convenient in the routine to compute the difference in energy between a particle at its old position and at its new position. First, the energy is computed at the new position, but when an overlap is detected, the code bails out early by returning `std::nullopt`. When no overlap is detected, the routines return the energy value. Optional return values are therefore a very convenient way of stating that two cases can occur:

30 September 2024 11:38:08

© Author(s) 2024

(1) either the computation has failed (e.g., overlap detected) or (2) the computation succeeded and the value is returned.

Another C++20 feature employed are modules. Modules eliminate or reduce many of the problems associated with the use of header files. Modules do away with header files and offer significant compile-time improvements and isolation of macros. Modules are only imported once and therefore indifferent to which order the modules are imported.

III. NEW FEATURES

A. GitHub

GitHub is an online software development platform, owned by Microsoft since 2018, for software development and version control. GitHub is centered around the Git version control system, providing all the features of Git for managing versions and source code. The system ensures that all the team members are working on the latest version of the code and can work simultaneously on the project. If users want to contribute to the project, they can use a "fork and pull request" workflow. GitHub Actions is a continuous integration and continuous delivery (CI/CD) platform that allows automation of the build, test, and deployment pipeline. Using GitHub Actions, after a `git push` command or a pull request, the unit tests are automatically run. Similarly, using GitHub Actions, the rpm-, deb-, and tzst-packages for Red Hat-, Debian-, and Arch-based Linux distributions are automatically generated (see Fig. S1). The RASPA3 package is available (MIT license) from GitHub at https://github.com/iraspa/RASPA3.

B. JSON

The input of RASPA3 has been changed to JSON format.[53] JSON stands for JavaScript Object Notation. It is a lightweight

```
{
  "SimulationType" : "MonteCarlo",
  "NumberOfCycles" : 25000,
  "NumberOfInitializationCycles" : 10000,
  "NumberOfEquilibrationCycles" : 10000,
  "PrintEvery" : 1000,

  "Systems" :
  [
    {
      "Type" : "Box",
      "BoxLengths" : [30.0, 30.0, 30.0],
      "ExternalTemperature" : 240.0,
      "ChargeMethod" : "Ewald",
      "GibbsVolumeMoveProbability" : 0.01
    },
    {
      "Type" : "Box",
      "BoxLengths" : [30.0, 30.0, 30.0],
      "ExternalTemperature" : 240.0,
      "ChargeMethod" : "Ewald",
      "GibbsVolumeMoveProbability" : 0.01
    }
  ],

  "Components" :
  [
    {
      "Name" : "CO2",
      "ThermodynamicIntegration" : true,
      "TranslationProbability" : 0.5,
      "RotationProbability" : 0.5,
      "ReinsertionProbability" : 0.0,
      "GibbsCFCMCSwapProbability" : 1.0,
      "CreateNumberOfMolecules" : [256, 256]
    }
  ]
}
```

General options.

System options

Component options

FIG. 2. JSON input example: Gibbs ensemble simulation at 240 K using two boxes, one for the gas phase and one for the liquid phase.

30 September 2024 11:38:08

© Author(s) 2024

data-interchange format in plain text written in JavaScript Object Notation. It is language independent. Figure 2 shows an example input for a Gibbs ensemble simulation at 240 K using two boxes, one for the gas phase and one for the liquid phase.

As can be seen in Fig. 2 for the Gibbs ensemble input, the input for components and the systems is coupled. There are two systems (see `Systems-keyword`), and one therefore needs to specify how many molecules one would like for system 0 and for system 1, respectively, defined by the `CreateNumberOfMolecules-keyword`. Using JSON, all data are immediately available to inspect and this cross-type of input is easier to parse in JSON than a top-down approach, which requires several sequential parses.

Similarly, the JSON format is now also used for the output. Where previously all initial data (e.g., hardware info and unit conversion factors) and statistics (e.g., CPU timings and MC move statistics) were written to a text file, these are now written as a nested dictionary to a JSON file. We also support the universal standard archive file-format for writing out adsorption data.[54]

C. CMake

CMake is likely the most popular and most powerful build system generator for C++. The build system generator CMake is platform- and compiler-independent. When porting a project from one system to another, the code does not have to be rewritten for CMake. There is a language called the CMake language, which allows embedding of intricate logic into CMake scripts.

CMake 3.28 and later support C++ modules. C++20 named modules are now supported by the Ninja Generator 1.11 or newer in combination with the LLVM/Clang 16.0 and newer, MSVC toolset 14.34 and newer, or GCC 14 and newer. We recommend LLVM/Clang 18 or higher for compiling RASPA3. This version has support for `std::format`, `std::print`, and `std::jthread`. The output-files of RASPA3 are in UTF-8 encoding and contain Unicode characters (e.g., for Å).

CMake contains the CPack system, which allows the creation of cross-platform packages and installers for Linux, Windows, and macOS. These packages are automatically created in GitHub upon release, using Docker-images for each operating system, and the CPack system; see Fig. 3 and Fig. S1.

D. Doxygen documentation

Doxygen is a documentation generator and static analysis tool for software source tree documentation. Information is extracted from specially formatted comments within the code when used as a documentation generator. When Doxygen is used for analysis, it uses its parse tree to generate diagrams and charts of the code structure. Documentation and code can be compared in Doxygen, making it easy for users when interpreting API documentation with code side by side.

Doxygen has built-in support for generating inheritance diagrams for C++ classes and composition visualization (see Fig. 4). Doxygen can use the `graphviz` tool to generate these advanced diagrams and graphs. Graphviz is an open-source, cross-platform graph drawing toolkit. The RASPA3 documentation can be found at https://iraspa.github.io/RASPA3/.

E. Unit tests

Software testing helps ensure code quality and is an important part of software development. Unit tests test the smallest functional

30 September 2024 11:38:08

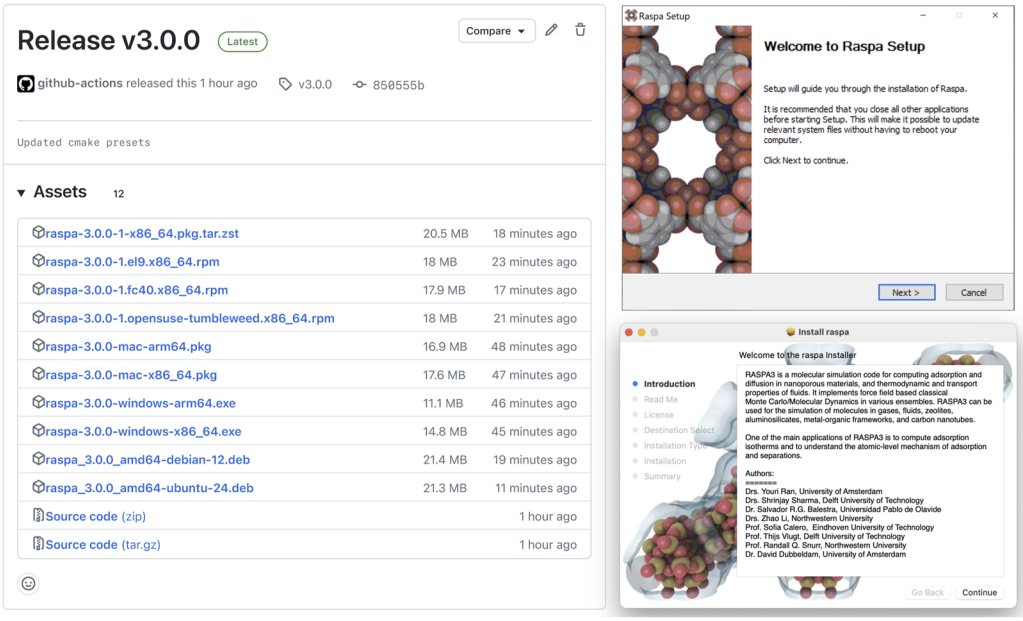


FIG. 3. GitHub release: binary packages are automatically created with CPack and GitHub Actions using docker-images for each operating system. We have provided installers for Windows (×86-64 and arm64) and macOS (×86-64 and Apple silicon).

© Author(s) 2024

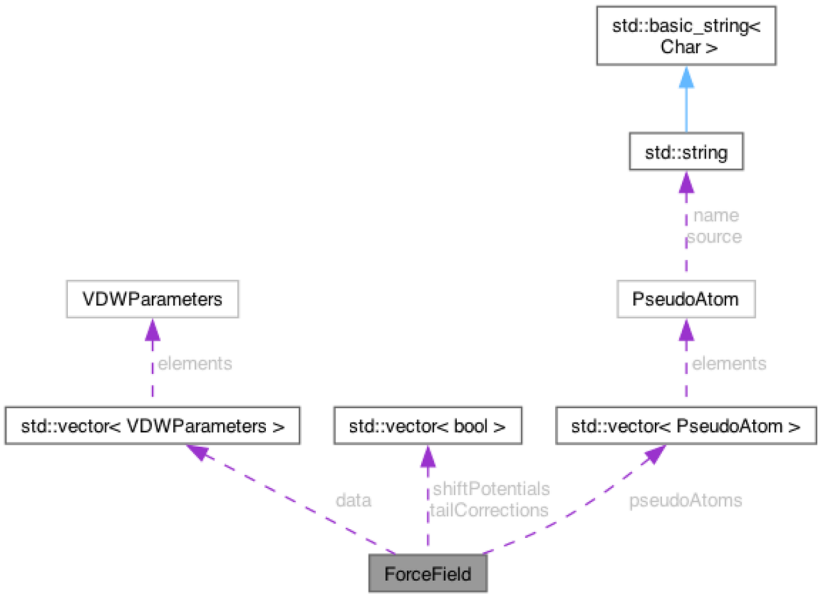


FIG. 4. Doxygen composition graph drawn with the graphviz tool. This graph shows how the code relates the force field object to objects for the van der Waals parameters and objects for the defined pseudo-atoms.

units of code and prevent developers breaking existing code when refactoring or adding new code. The GoogleBenchmark framework is used for benchmarking,[55] and the GoogleTest framework is used for testing and mocking.[56] It is a platform-independent device that automates testing. Using GitHub Actions, the code is automatically built and tested (see Fig. 5).

The unit tests in RASPA3 are arranged hierarchically. Atoms are placed at several distances, and the Lennard-Jones potentials are tested and compared with the analytically computed energy. Then, molecules are placed within a zeolite framework at predetermined positions and compared to the energies computed with RASPA2. The various energy routines used in biased-sampling are tested by comparing to the general routines. The various routines for the gradients are tested by comparing the computed gradients to the values computed by finite difference schemes based on the energy. Likewise, the strain-derivative tensor (related to the pressure) is tested by comparing to the values computed by finite difference schemes based on the energy of a strained cell. Some NIST reference calculations of intermolecular energy for the SPC/E water system have also been included in the unit tests (see Table S1).

F. Python interface

The RASPA3 C++ simulation engine is made available to Python via a Pybind11 interface.[57] For use in Python, RASPA3 is built as a shared library, allowing its functions to be used by Python users and enabling seamless interactions with the simulation routines. Through the API, the library allows for invocation of RASPA3's simulation routines from Python scripts, calling the same simulation routines as via the JSON input. This execution directly from Python enables the ability to prototype simulation settings and incorporate RASPA3 into existing workflows.

Moreover, the RASPA3 library functions can directly be called from Python, enabling more advanced scripting and building pipelines. Such pipelines, for example, can be built by having modules interact with the simulation results and runtime settings. As such, one could device feedback loops that check convergence or check on multiple simulations simultaneously. In addition, in line with modern (generative) Artificial Intelligence (AI) applications, it allows the users to have an external agent modify the current state of the system.

As of publication (2024), current pip and Pybind11 build systems do not yet support C++23 modules, necessitating the build to be carried out as a shared library from CMake. While this does not impact functionality, it does limit the option to build the RASPA3 library in line with PEP-518 requirements. Examples demonstrating the functionality of the Python interface and pipelines are implemented and documented in the examples directory on GitHub. These examples serve as practical guides, illustrating how to effectively utilize the interface. An example of a simulation carried out with interfaced RASPA3 code is shown in Fig. S2.

G. Cube-file output

The Gaussian software package defines the cube file format for describing volumetric data and atom positions. The file has a header that contains the atom's information, box vectors, and size of the volumetric data. The volumetric data follow one scalar per voxel element. All aspects of the file are text (human readable). Originally, the numerical values were five characters wide for integers that started each header line, and floating point values were formatted 12.6 (12 characters wide with six decimal places). The file contains one line for each atom consisting of five numbers: the first is the atom number, the second is the charge, and the last three are the x, y, z

30 September 2024 11:38:08

© Author(s) 2024

GitHub Actions / RASPA Tests
succeeded 2 hours ago in 0s

50 passed, 0 failed and 0 skipped

tests 50 passed

✅ /home/runner/work/raspa3/raspa3/testResults.xml

50 tests were completed in 1s with 50 passed, 0 failed and 0 skipped.

Test suite	Passed	Failed	Skipped	Time
cbmc_interactions	2✅			3ms
dudlambda	7✅			35ms
energy_decomposition	3✅			106ms
Ewald	8✅			133ms
gradients	9✅			710ms
insertion_deletion	4✅			22ms
MC	1✅			11ms
MC_strain_tensor	2✅			1ms
pbc	1✅			0ms
rigid_gradient	3✅			15ms
spacegroup	7✅			6ms
static_energy	3✅			22ms

FIG. 5. GitHub unit tests: automatic code building and running of unit tests using GitHub Actions.

coordinates of the atom center. This is followed by a list of N^3 floating point numbers for each specified voxel that gives the normalized number density at that point.

This file format is used in RASPA3 to output the average density measured on the voxelgrid in a simulation. These densities can be used to visualize the density of an adsorbate in a nanoporous material as in Fig. 1, where methane is shown in MFI and CO_2 is shown in Cu-BTC. RASPA3 stores the data in a unit-cube format using fractional positions and hence can handle triclinic cells (viewable with iRASPA).

H. HDF5 output

RASPA3 includes a Hierarchical Data Format version 5 (HDF5) writer for logging data produced during the simulation.[58] It offers many benefits compared to the previously used text output. HDF5 is a file format that uses a file system-like hierarchy to structure data outputs. The advantages include data compression, parallel I/O support, and easy accessibility from both python (with h5py and xarray) and other languages using the HDF5 library. Crucially, the dataset can be accessed without loading the full dataset into memory.

An HDF5 file is hierarchically structured as represented in the graph in Fig. 6, where the data are split into groups, datasets, and

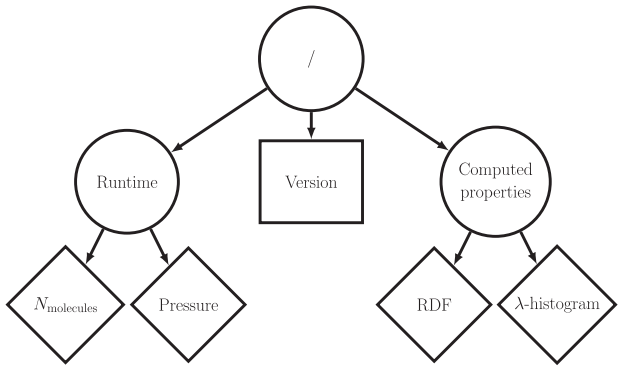


FIG. 6. HDF5 file layout graph: each file has a local graph structure that can be represented by groups (circles), metadata (rectangles), and datasets (diamonds).

metadata. Groups act like directories, which hold multi-dimensional arrays called datasets and named attributes, herein named metadata. Accessing, for example, the RDF dataset can be done in a POSIX-like syntax by selecting `/computed_properties/rdf`.

In RASPA3, the output data are split into several groups to maintain clarity and ease of access. The primary groups created are `runtime` and `computed_properties`, containing data generated during the run and statistics of computed properties, respectively. Both of these groups hold datasets, which hold multi-dimensional data. The size of the dataset is determined by the sampling frequency and the dimensionality of the data itself. For example, a dataset `BoxLengths` in the group `runtime_log` will have size (N, 3, 3) for the current 3 × 3 box matrix at each of the N write steps.

I. Quaternions

Flexible molecules have no internal constraints and are specified using internal potentials, such as bond-, bend-, and torsion-potentials. For efficiency reasons, many small molecules are modeled as rigid. Examples are water, CO_2, N_2, CO, O_2, and CH_4. Rigid molecules are described by a center of mass and an orientation.

Integration of Newton's equations of motion is most conveniently performed using the center of mass and quaternions. In the classical mechanics of rigid body motion, the unit quaternion, $\{q_0, q_1, q_2, q_3\}$, is introduced in order to generate a minimal, non-singular representation of the rotation matrix from a space-fixed system, denoted "s," to a body-fixed coordinate system, denoted "b,"

$$r^{(b)} = A(q)r^{(s)}, \tag{1}$$

$$r^{(s)} = A(q)^T r^{(b)}, \tag{2}$$

where $A(q)$ is a rotation matrix computed from the quaternion q. In the body-fixed coordinate system, the moment of inertia tensor I is diagonal. Upon reading the molecule file, the moment of inertia tensor is computed and then diagonalized. The positions are then recomputed in the body-fixed coordinate system.

Monte Carlo moves on rigid molecules have been re-implemented to make use of quaternions. Translation moves change the center of mass, while rotation moves operate on the

30 September 2024 11:38:08

© Author(s) 2024

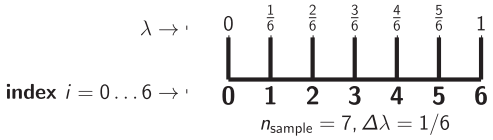


FIG. 7. Schematic representation of the λ value mapping from index to λ value used for modeling discrete λ-values for the fractional molecules defined in CFCMC simulations.

quaternion. Trial positions are then generated using Eq. (2). The advantage is that MD/MC hybrid moves are now consistent between MC and MD. Moreover, it is no longer necessary to compute quaternions from atomic positions. The MD integration of the equations of motions for rigid molecules in RASPA3 is based on the quaternion implementation of Miller *et al.*[59]

J. Expanded ensembles

Measuring chemical potentials in open ensembles at high densities is difficult due to the lack of sufficient insertions and deletions of particles. The Widom-insertion methodology similarly fails at high densities and/or low temperatures.[11] As single-step insertions are not efficient, it is natural to consider insertions in multiple steps in such a way that the surrounding of a molecule that is inserted can simultaneously adapt. This is the central idea that is applied in expanded-ensemble methods. In the expanded ensemble methods in general, interactions of a so-called fractional molecule are scaled with a coupling parameter λ in such a way that interactions between the fractional molecule and the surrounding "whole" molecules (either other adsorbates or framework atoms) vanish for $\lambda = 0$ and that those interactions are fully developed for $\lambda = 1$.[10,11] By including Monte Carlo trial moves for sampling λ, different (expanded) ensembles, such as the NPT-, grand-canonical-, reaction-, and Gibbs ensemble, can be simulated. The CFCMC method considerably improves the insertion or deletion of molecules while allowing for a direct computation of chemical potentials and partial molar properties.[8,10,14]

RASPA3 uses a discrete λ-sampling implementation, as opposed to the continuous λ-sampling often employed. Using n_{sample} sample points, with $\Delta\lambda = \frac{1}{n_{\text{sample}}-1}$, we can map the indices $i = 0 \ldots n_{\text{sample}}$ to λ-values,

$$\lambda_i = \Delta\lambda \times i, \qquad i = 0 \ldots n. \tag{3}$$

The first sample index 0 corresponds to $\lambda = 0$, and the last sample-index n corresponds to $\lambda = 1$. Figure 7 shows a graphical representation of the mapping for $n_{\text{sample}} = 7$. It is preferable that an odd number of sample points are used and, hence, the numbers of intervals is even, because then $\lambda = \frac{1}{2}$ is explicitly defined at sample index $(n_{\text{sample}} - 1)/2$.

K. Thermodynamic integration

Thermodynamic integration is a method used to calculate the difference in free energy between two given states, for example, to calculate the free energy of inserting a molecule into a system.[8] The integration runs from state A, with $\lambda = 0$, to state B, with $\lambda = 1$, where the interactions of the inserted molecule are gradually scaled from 0 to that of a full molecule. The change in free energy between states A with $\lambda = 0$ and state B with $\lambda = 1$, i.e., the insertion of a particle, can be computed from the integral of the ensemble averaged derivatives of the potential energy over the coupling parameter λ,

$$\Delta F(A_{\lambda=0} \rightarrow B_{\lambda=1}) = \int_{\lambda=0}^{\lambda=1} \left\langle \frac{\partial U(\lambda)}{\partial \lambda} \right\rangle_{\lambda} \mathrm{d}\lambda. \tag{4}$$

This is performed by defining a potential energy function $U(\lambda)$, sampling the ensemble of equilibrium configurations at a series of λ values, calculating the ensemble-averaged derivative of $U(\lambda)$ with respect to λ at a series of λ values from 0 to 1, and finally computing the integral over the ensemble-averaged derivatives.

In general, interaction types have different optimal scaling from non-interacting to full interaction, because their functional forms and interaction range differ. Van der Waals and electrostatic interactions each have their own scaling factors λ_{vdw} and λ_{el}. Increasing λ_{vdw} smoothly scales from 0 to 1 between $\lambda \in [0, \frac{1}{2}]$ and λ_{el} smoothly scales from 0 to 1 in $\lambda \in [\frac{1}{2}, 1]$, ensuring that when inserting a molecule, the van der Waals interactions are first activated and then the Coulomb interactions. For the Coulomb interactions, λ_{el} scales the charges, whereas for the scaled Lennard-Jones, a soft-core potential is used to avoid divergence at $r = 0$ when $\lambda \rightarrow 0$. The value of $\partial U/\partial\lambda$ is computed using the chain rule,[15]

$$\frac{\partial U}{\partial \lambda} = \frac{\partial U_{\text{vdw}}}{\partial \lambda_{\text{vdw}}} \frac{\partial \lambda_{\text{vdw}}}{\partial \lambda} + \frac{\partial U_{\text{el}}}{\partial \lambda_{\text{el}}} \frac{\partial \lambda_{\text{el}}}{\partial \lambda}, \tag{5}$$

where $\frac{\partial \lambda_{\text{vdw}}}{\partial \lambda}$ and $\frac{\partial \lambda_{\text{el}}}{\partial \lambda}$ depend on the switching function being used.

The discrete λ-implementation of RASPA3 has the advantage that the sample point probes the exact λ value, instead of an average over the bin. All states, including the states $\lambda = 0$ and $\lambda = 1$, are *explicitly* sampled. Moreover, Simpson's integration rules for thermodynamic integration can then be conveniently applied.[60] The Simpson's 1/3 rule assumes a quadratic behavior for $f(\lambda_i)$,

$$I \approx \frac{\Delta\lambda}{3}\Big[\underbrace{f(0) + 4f(1) + f(2)} + \underbrace{f(2) + 4f(3) + f(4)} + \cdots + \underbrace{f(n-2) + 4f(n-1) + f(n)}\Big], \tag{6}$$

which reduces to

$$I \approx \frac{\Delta\lambda}{3}\big[f(0) + 4f(1) + 2f(2) + 4f(3) + 2f(4) + \cdots + 2f(n-2) + 4f(n-1) + f(n)\big]. \tag{7}$$

The Simpson's 3/8 rule assumes a cubic behavior,

$$I \approx \frac{3\Delta\lambda}{8}\Big[\underbrace{f(0) + 3f(1) + 3f(2) + f(3)} + \underbrace{f(3) + 3f(4) + 3f(5) + f(6)} + \cdots + \underbrace{f(n-3) + 3f(n-2) + 3f(n-1) + f(n)}\Big], \tag{8}$$

30 September 2024 11:38:08

© Author(s) 2024

which reduces to

$$I \approx \frac{3\Delta\lambda}{8}\big[f(0)+3f(1)+3f(2)+2f(3)+3f(4)+3f(5)+2f(6)+\cdots+2f(n-3)+3f(n-2)+3f(n-1)+f(n)\big]. \quad (9)$$

Simpson's 3/8 rule is about twice as accurate as the 1/3 rule.

L. Transition-matrix Monte Carlo

Subcritical gas adsorption in which capillary phase transitions are present can be simulated using the grand canonical transition-matrix Monte Carlo (GC-TMMC).[61,62] The goal of GC-TMMC is to calculate the particle number probability distribution (PNPD) instead of direct calculation of ensemble averages. A biasing scheme encourages accurate examination of metastable states. At its core, the method computes the transition matrix $\boldsymbol{P}$ with probabilities of changing particle number. This transition matrix can be used to obtain the PNP,

$$\ln\Pi(N+1;\mu,V,T) = \ln\Pi(N;\mu,V,T) + \ln\left[\frac{P(N\to N+1)}{P(N+1\to N)}\right] \quad (10)$$

at state (μ, V, T), where Π is the probability of observing N particles. The transition matrix $\boldsymbol{P}$ can be computed through a biased sampling, where a biasing factor η is added to the acceptance criterion to sample uniformly from particle number N,

$$p_{\text{bias}}(o\to n) = \min\left[1, \frac{\exp[\eta(N_n)]}{\exp[\eta(N_o)]}\frac{\alpha(\text{n}\to\text{o})\pi_n}{\alpha(\text{o}\to\text{n})\pi_o}\right], \quad (11)$$

where $\alpha(\text{o}\to\text{n})$ is the probability of generating the new configuration n from o and π_o is the probability of being in state o. The biasing function η is iteratively optimized to match $\eta(N) = -\ln\Pi(N;\ \mu,V,T)$. The histogram reweighting of the PNPD at one thermodynamic state point can be used to generate that of another state point without running an additional simulation, using

$$\ln\Pi(N;\mu,V,T) = \ln\Pi(N;\mu_0,V,T) + \beta N(\mu-\mu_0) + C. \quad (12)$$

The newly generated PNPD can be used to generate any of the ensemble averages and thermophysical properties that were derived from the initial PNPD. Thus, a single GC-TMMC simulation can produce an entire adsorption isotherm while also identifying the conditions of phase coexistence, limits of stability, and other properties of interest, at a particular temperature.

M. Radial distribution function

The Radial Distribution Function (RDF), $g(r)$, is one of the most useful tools to measure the structure of a fluid at molecular length scales. It is often defined as the ratio of the average local number density of particles $\langle\rho(r)\rangle$ at a distance r from a given particle to the bulk density of particles ρ,

$$g(r) = \frac{\langle\rho(r)\rangle}{\rho}. \quad (13)$$

The pair distribution function in a uniform classical fluid is equivalent to the one-body density when one particle is fixed.[63] Mathematically, the one-body density distribution is defined as

$$\rho(\mathbf{r}) = \left\langle \sum_i \delta(\mathbf{r}-\mathbf{r}_i) \right\rangle, \quad (14)$$

where the sum runs over all particles, δ indicates the Dirac distribution, and $\mathbf{r}_i$ is the position of particle i. The standard particle-based approach to sample $\rho(\mathbf{r})$ is to discretize the Dirac function and to count events in a histogram, labeled by position r and with bins of a certain size ΔV. Normalization by ΔV and by the number of sampling sweeps ensures the correct normalization $\int\rho(\mathbf{r})d\mathbf{r} = N$, where N is the total number of particles. The problem of histogram-based (i.e., "binning") estimates is that the variance diverges as the bin size h decreases.[64]

It is not so much the density that matters but rather the variations in density with respect to distance. The idea behind force-sampling strategies is to sample these variations using estimators involving the force acting on the atoms.[65,66] Borgis *et al.*[65] proposed a virial-like estimator as an alternative to Eq. (14). The variance of this force-based estimator is significantly reduced compared to histograms. Force sampling was shown to be ~5 times more accurate than counting.[66] The scheme can be seen as an integration of $g'(r) = dg/dr$ from $r\to\infty$, where $g = 1$. Note that this is different

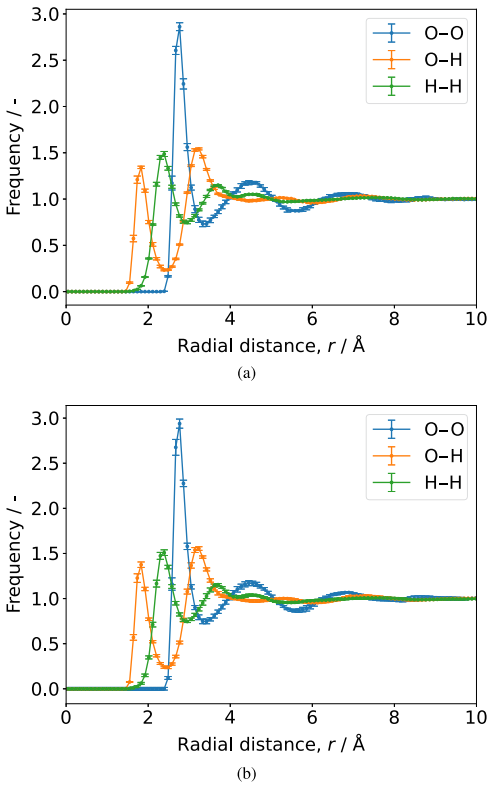


FIG. 8. Radial distribution function (RDF) of water at 298 K and 1000 kg m^{-3}, modeled using the Tip5p water model: (a) conventional RDF from normalized distance histograms; (b) RDF from force averaged method.

30 September 2024 11:38:08

© Author(s) 2024

from the standard method where the plateau value differs from 1 by a $\mathcal{O}(N^{-1})$ correction.[64]

Figure 8 shows the RDF for water using the counting method vs the force-based method. For a rigid molecule, $\mathbf{f}_i$ is the sum of forces acting on particle i and all particles participating in a constraint with i (in particular, those belonging to the same rigid molecule as i) and where the average is made over configurations satisfying all constraints. In practice, the gradient of the water oxygen (respectively, hydrogen) density is computed by assigning the total force acting on each molecule to the position of the O atom (respectively, H atoms).[64] Both RDF methods give the same results within statistical error.

IV. RESULTS

A. Validation

1. Adsorption

In adsorption studies, one would like to know the amount of molecules adsorbed as a function of pressure and temperature of the reservoir with which the adsorbent is in contact. Therefore, the natural ensemble to use is the grand-canonical ensemble (or μ, V, T ensemble). In this ensemble, the temperature T, the volume V, and the chemical potential μ are fixed. The equilibrium conditions are that the temperature and chemical potential of the gas inside and outside the adsorbent must be equal. The imposed chemical potential μ can be related to the fugacity f computed directly from the equation of state of the vapor in the reservoir. RASPA3 was checked by comparing the computation of isotherms to RASPA2 in several systems: methane adsorption in MFI, CO_2 in MFI, an equimolar mixture of methane and CO_2 in MFI, and CO_2 adsorption in Cu-BTC. The used force fields and settings are listed in the supplementary material (Tables S2, S9, S16, and S20). For all systems, 1×10^6 cycles were used for the production time.

The adsorption results for all systems show statistically identical results for absolute adsorption loadings (Tables S3, S10, S17, and S21), enthalpies of adsorption (Tables S6, S11, S18, and S24), and chemical potentials (Tables S7 and S25) over a wide range of pressures. The chemical potentials are known in advance, since the chemical potential is directly related to the imposed chemical potential or imposed fugacity. In Tables S8, S15, and S26, we show that the measured fugacity is statistically identical to the imposed fugacity, which is a very good indicator of the consistency and correctness of the results.

In Fig. 9, we show that the particle distributions, closely resembling the PNPD, sampled with CB/CFCMC[10] are identical in RASPA2 and RASPA3. We compared the results of four insertion/deletion methods: conventional unbiased insertion/deletion, CBMC, CFCMC, and CB/CFCMC. The particle distributions produced by all four used Monte Carlo insertion/deletion methods show statistically identical particle distributions as shown in Figs. S6 and S11. The distribution and the fluctuations are related to the slope of the adsorption isotherm.[67] The steeper the slope, the larger the amount of fluctuations, which makes steep inflections (e.g., water) hard to sample. Even though the particle distributions might seem similar, properties that are computed based on these fluctuations do show differences depending on what insertion/deletion method is used. For example, the conventional insertion/deletion method shows larger error-bars in the enthalpy of adsorption compared to CBMC or CFCMC methods. This shows that the conventional swap becomes inadequate at higher loadings to predict the adsorption behavior of these systems.

Similarly to the enthalpy of adsorption, conventional or even biased Widom insertions are only valid to compute the chemical potential at low densities and fail at high loadings. We computed the chemical potential using three methods: (1) using the logarithm of the ensemble average of the Rosenbluth weights, (2) using $\partial U/\partial\lambda$ with Simpson's rule as described in Sec. III K, and (3) using the difference in probabilities $\beta^{-1}\log(p(\lambda=1)/p(\lambda=0))$, as described in Ref. 68. The calculated chemical potential for methane in MFI is shown in Fig. 10, and it shows that for higher densities, the calculations based on the λ particles with CFCMC result in lower errors. A similar trend can be seen for the chemical potential of CO_2 in Cu-BTC in Fig. S7. The results for the chemical potential and error are shown in Tables S7 and S25.

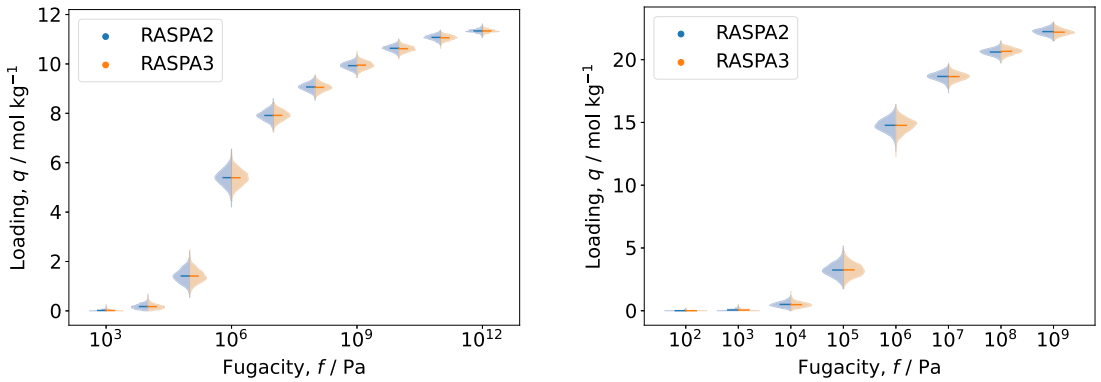


FIG. 9. Particle distributions of GCMC adsorption simulations with CB/CFCMC insertions in RASPA2 (blue) and RASPA3 (orange) simulated with CBMC/CFCMC at 300 K, for (a) methane in MFI and (b) CO_2 in Cu-BTC. The distributions shown closely resemble the PNPD and show the variance in the particle number.

30 September 2024 11:38:08

© Author(s) 2024

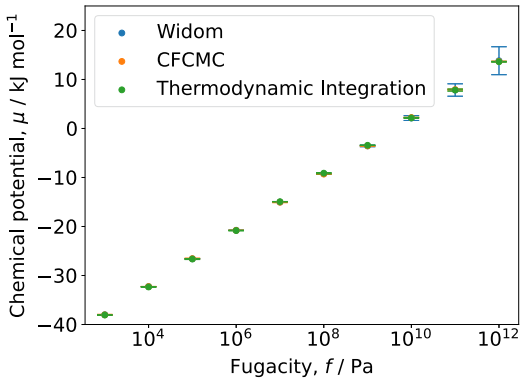


FIG. 10. Chemical potential of methane insertion in MFI at 300 K measured as a function of fugacity, computed with Widom particle insertion as $\langle \exp(-\beta U^{+}) \rangle$ (blue), the ensemble average ratio of CFCMC lambda probability $\langle \ln \frac{p(\lambda=1)}{p(\lambda=0)} \rangle$ (orange), and thermodynamic integration as defined in Sec. III K.

We also compared the acceptance rates of the insertion moves and see that for high fugacities, the acceptance rates for the conventional and CBMC swap methods drop off exponentially. An acceptance rate as low as 10^{-4} is observed for the CBMC swap method at saturation pressures as shown in Figs. S3 and S8. Acceptance rates this low cause the effective correlation time to increase and lead to large errors in the predicted observables.

2. Gibbs-ensemble for VLE

Several phases can coexist when the following equations are satisfied:[4]

$$T^{\text{vapor}} = T^{\text{liquid}}, \tag{15}$$

$$p^{\text{vapor}} = p^{\text{liquid}}, \tag{16}$$

$$\mu_i^{\text{vapor}} = \mu_i^{\text{liquid}}, \qquad i = 1, \ldots, m, \tag{17}$$

where T is the temperature, p is the pressure, μ_i is the chemical potential of component i, and m is the number of components. Simulations in the Gibbs Ensemble (GE) are frequently used to study Vapor–Liquid Equilibrium (VLE) of pure components and mixtures.[69] The vapor and liquid phases are simulated using two boxes, which for single components, can change individual volumes (while keeping the total volume constant), leading to equal pressure in both boxes. In addition, particle transfer between the boxes is performed to obtain equal chemical potentials in both boxes. For mixtures, the volume move operates on the individual boxes. VLE-GE is routinely used to develop force fields, because the force field parameters are very sensitive to changes in environment. If a model successfully reproduces VLE data, it reproduces fluid densities over a wide range of pressures and temperatures. The TraPPE model[70–75] is an example of such a force field fitted to VLE data.

Similar to simulations in the grand-canonical ensemble, GE simulations rely on sufficiently large acceptance probabilities for particle exchanges between the simulation boxes. However, the acceptance probability for particle exchange can be very low when molecules are large or when the densities are high. When the acceptance probability for insertion/deletion is low, it is not straightforward to verify if the two phases have reached chemical equilibrium and that the chemical potentials of a certain component are equal in the simulation boxes. To overcome this problem, one possible solution is applying expanded ensemble methods.[76–78] The CFCMC method, recently introduced by Shi and Maginn,[5,6] is one of the most commonly used expanded ensemble approaches. In the CFCMC GE, there are fractional particles with reduced interactions with the surrounding molecules (instead of inserting whole molecules at once). The strength of this interaction is controlled by a coupling parameter lambda (λ) over which we integrate in the partition function, hence the name expanded ensemble. Poursaeidesfahani *et al.* introduced a more efficient formulation of GE combined with the CFCMC technique.[8] In this formulation, there is only a single fractional component per molecule type that can be in either one of the boxes. This increases the acceptance ratio for particle insertion/deletion. The advantage of the CFCMC method lies within the fact that it does not depend on the occurrence of spontaneous cavities in the system (as in CBMC). The molecules are inserted gradually in the system instead of a single trial move. In the new formulation of the CFCMC GE, the chemical potential can be computed directly without any extra calculations.

In the supplementary material, we have listed the simulation data. The force field was listed in S27. From Table S28, we see that RASPA2 and RASPA3 give similar results, for both the CBMC and Poursaeidesfahani CFCMC method. Tables S29 and S30 show that we indeed have obtained mechanical and chemical equilibrium, since the pressure and the chemical potentials of the vapor and liquid phase are equal to within statistical error. Table S31 shows the acceptance ratios for the CFCMC-GE scheme. The fractional particle indeed shows near equal λ-occupancy in the vapor and liquid phases, due to the biasing applied. The *swap* move swaps a randomly chosen integer molecule of another box with the current fractional molecule. The acceptance of this move for this system is higher than 40%. The move to change the λ value is scaled to achieve 50% acceptance. The *shuffle* move to shuffle the fractional move to one of the other boxes is around 10%–15%. The *shuffle* move is very effective at low λ values, while the *swap* move has high acceptance at near unity λ values. Finally, Table S32 shows that the chemical potentials computed from Widom particle insertion and from the CFCMC λ-histograms and thermodynamic integration are very similar for this simple system. For more difficult systems, Widom particle integration will start to fail, while the CFCMC-methodology remains effective.[15]

3. Molecular dynamics

Molecular Dynamics (MD) mimics nature by computing the positions $\boldsymbol{x}(t)$ and velocities $\boldsymbol{v}(t)$ of particles as a function of time t according to the equations of motion of Newton:[3] $\boldsymbol{f} = m\boldsymbol{a}$, which relates the force $\boldsymbol{f}$ to the mass m times the acceleration $\boldsymbol{a}$. Newton's equations of motion yield a constant energy ensemble. To extend MD to other ensembles, the system of equations needs to be extended. The Nosé–Hoover equations of motion introduce temperature control.[79] Martyna and Tuckerman and co-workers derived equations of motion for temperature and pressure control.[79–82] For each case, a conserved quantity can be derived that can be used to

30 September 2024 11:38:08

© Author(s) 2024

check the integration error. Figures S12 and S13 show the excellent energy conservation for an NVT simulation of over 100 ns of methane and CO_2, respectively, in $2 \times 2 \times 2$ unit cells of IRMOF-1. Even though the Nosé–Hoover equations of motion perturb the motions, the changes are gentle enough that the obtained values are similar to the values obtained from NVE MD at the same average temperature. For diffusion, no significant influence is observed over a wide range of thermostat masses.[4]

In an equilibrium molecular dynamics (EMD) simulation, the self-diffusion coefficient D_α^S of component α is computed by taking the slope of the mean-squared displacement (MSD) at long times,

$$D_\alpha^S = \frac{1}{2dN_\alpha} \lim_{t\to\infty} \frac{d}{d\,t} \left\langle \sum_{i=1}^{N_\alpha} \left(r_i^\alpha(t) - r_i^\alpha(0) \right)^2 \right\rangle, \tag{18}$$

where N_α is the number of molecules of component α, d is the spatial dimension of the system, t is the time, and r_i^α is the center-of-mass of molecule i of component α. The order-n algorithm for measuring MSDs conveniently and efficiently captures correlations over short, medium, and long times.[83] In Fig. S14, we show simulated diffusivities of small gasses in IRMOF-1 at 298 K, compared to previous results of results of Skoulidas and Sholl.[84]

B. Timings

The same simulations used in Sec. IV A were used to measure comparative efficiency of the simulation codes. Figure 11 shows that a significant speedup is observed. The speedup for the simulation of methane in MFI (non-Ewald) is 4–6×, and the speedup of the CO_2 in Cu-BTC is about 2–4×. This reduces the total core hours necessary to do these isotherm calculations from 170 to 36 core hours for methane in MFI and from 408 to 113 core hours for CO_2 in Cu-BTC (with CBMC/CFCMC).

In both cases, a higher speedup is observed in the higher fugacity ranges. In these ranges, a higher density is expected, and therefore more particles. Specifying the speedup per type of Monte Carlo move in Figs. S4 and S9 shows that the speedup is most noticeable in the CBMC Widom and reinsertion moves, with speedups of around a factor of 7. This likely explains the fugacity dependence of the speedup. Also promising is the factor 3.5 speedup of the translation move, which is a key move in all MC simulations.

These results clearly show that the speedup is a result of the design choices described in Sec. II. The improved memory layout for the atom lists and the value-based semantics make accessing the particles faster and more reliable. Hence, we see an improved performance for every Monte Carlo move. These results greatly improve the capacity to quickly screen porous materials by computing adsorption isotherms.

C. Parallelization

The OpenMP API defines a portable, scalable model with a simple and flexible interface for developing parallel applications in C/C++ and Fortran. By adding some additional pragma statements,

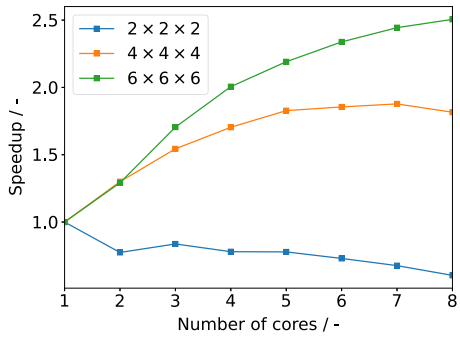


FIG. 12. Multi-threading speedup results as a function of the number of cores for methane in MFI at 300 K.

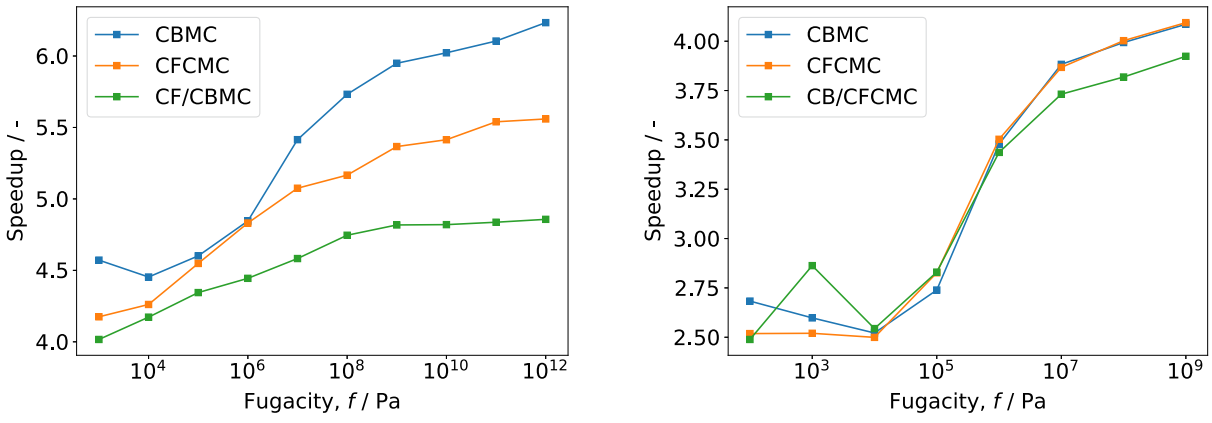


FIG. 11. Relative simulation time of a GCMC adsorption simulation of the RASPA2 implementation compared to the RASPA3 implementation for (a) methane in MFI and (b) CO_2 in Cu-BTC.

30 September 2024 11:38:08

© Author(s) 2024

these OpenMP directives can exploit shared memory parallelism by defining various types of parallel regions. Figure 12 shows some preliminary results on parallelization of the framework–molecule energy computation used in the CBMC routine. As can be seen, parallelization for small systems can be detrimental because not enough work can be given to the cores to overcome the cost of spawning the threads and combining the results. For sufficiently large systems, it starts to pay off, but the gains are marginal. However, for obtaining an equilibrated system, this might be acceptable. Once an equilibrated system is available, task-farming parallelism can be exploited. GPU-implementations show substantial speedups compared to serial CPU implementations of Monte Carlo, and also compared to thread-based/OpenMP CPU parallelization.[62] Currently, RASPA3 has no GPU acceleration.

V. CONCLUSIONS

RASPA3 is a highly efficient Monte Carlo code with a particular emphasis on simulations of adsorption in porous materials and thermodynamic properties of fluids. It represents a significant upgrade to state-of-the-art technologies, such as CMake, unit-tests, and Doxygen documentation. The code was re-implemented in modern C++23, resulting in a factor of 3–6 speedup compared to RASPA2 while doing more work like the computation and bookkeeping of $dU/d\lambda$. A further quick speedup of a factor of 1.5–2 can be achieved by multi-threading using OpenMP, but only for relatively large systems. The code is freely available at GitHub and licensed under the MIT license.

Using the RASPA3 code, we have shown that the modern CBMC technique is a significant improvement over the standard conventional unbiased Metropolis MC technique and that recent techniques, such as CFCMC and CB/CFCMC, have overcome the downside of standard MC and CBMC, namely the drop in insertion/deletion rates in open ensembles at low temperatures and high densities. Even though the sampled particle distributions using all four methods look similar, there are significant differences observed at high densities for properties that are computed via fluctuation formulas and for the chemical potential. We recommend the use of CB/CFCMC for adsorption and vapor–liquid equilibrium computations.

SUPPLEMENTARY MATERIAL

The supplementary material contains simulation results on the adsorption of methane and CO_2 in MFI, an equimolar mixture of CO_2 and N_2 in MFI, and CO_2 in Cu-BTC; a Gibbs VLE simulation of methane; and diffusivity results on small gas molecules in IRMOF-1.

ACKNOWLEDGMENTS

R.Q.S. gratefully acknowledges support from the U.S. Department of Energy, Office of Basic Energy Sciences, Division of Chemical Sciences, Geosciences and Biosciences under Award No. DE-SC0023454. S.R.G.B. acknowledges Grant No. RYC2022-036070-I funded by MICIU/AEI/10.13039/501100011033 and by "ESF+".

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Y. A. Ran: Investigation (equal); Software (equal); Validation (equal); Visualization (equal); Writing – original draft (equal). **S. Sharma**: Methodology (equal); Software (equal). **S. R. G. Balestra**: Methodology (equal); Software (equal). **Z. Li**: Methodology (equal); Software (equal). **S. Calero**: Methodology (equal); Software (equal); Supervision (equal). **T. J. H Vlugt**: Methodology (equal); Software (equal); Supervision (equal). **R. Q. Snurr**: Methodology (equal); Software (equal); Supervision (equal). **D. Dubbeldam**: Investigation (equal); Methodology (equal); Software (equal); Supervision (equal).

DATA AVAILABILITY

All our software is freely available from GitHub, and the simulations results are written down in the supplementary material.

REFERENCES

[1] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Equation of state calculations by fast computing machines," J. Chem. Phys. **21**, 1087–1092 (1953).
[2] M. Rosenbluth and A. Rosenbluth, "Monte Carlo calculation of the average extension of molecular chains," J. Chem. Phys. **23**, 356–359 (1955).
[3] M. Allen and D. Tildesley, *Computer Simulation of Liquids*, 2nd ed. (Clarendon Press, Oxford, 2017).
[4] D. Frenkel and B. Smit, *Understanding Molecular Simulation*, 3rd ed. (Academic Press, London, UK, 2023).
[5] W. Shi and E. J. Maginn, "Continuous fractional component Monte Carlo: An adaptive biasing method for open system atomistic simulations," J. Chem. Theory Comput. **3**, 1451–1463 (2007).
[6] W. Shi and E. J. Maginn, "Improvement in molecule exchange efficiency in Gibbs ensemble Monte Carlo: Development and implementation of the continuous fractional component move," J. Comput. Chem. **29**, 2520–2530 (2008).
[7] T. W. Rosch and E. J. Maginn, "Reaction ensemble Monte Carlo simulation of complex molecular systems," J. Chem. Theory Comput. **7**, 269–279 (2011).
[8] A. Poursaeidesfahani, A. Torres-Knoop, D. Dubbeldam, and T. J. H. Vlugt, "Direct free energy calculation in the continuous fractional component Gibbs ensemble," J. Chem. Theory Comput. **12**, 1481–1490 (2016).
[9] A. Rahbari, R. Hens, M. Ramdin, O. A. Moultos, D. Dubbeldam, and T. J. H. Vlugt, "Recent advances in the continuous fractional component Monte Carlo methodology," Mol. Simul. **47**, 804–823 (2021).
[10] A. Torres-Knoop, S. P. Balaji, T. J. H. Vlugt, and D. Dubbeldam, "A comparison of advanced Monte Carlo methods for open systems: CFCMC vs CBMC," J. Chem. Theory Comput. **10**, 942–952 (2014).
[11] B. J. Sikora, Y. J. Colón, and R. Q. Snurr, "Continuous fractional component Monte Carlo simulations of high-density adsorption in metal-organic frameworks," Mol. Simul. **41**, 1339–1347 (2015).
[12] D. Dubbeldam, A. Torres-Knoop, and K. S. Walton, "On the inner workings of Monte Carlo codes," Mol. Simul. **39**, 1253–1292 (2013).
[13] Y. Nejahi, M. Soroush Barhaghi, J. Mick, B. Jackman, K. Rushaidat, Y. Li, L. Schwiebert, and J. Potoff, "GOMC: GPU optimized Monte Carlo for the simulation of phase equilibria and physical properties of complex fluids," SoftwareX **9**, 20–27 (2019).
[14] R. Hens, A. Rahbari, S. Caro-Ortiz, N. Dawass, M. Erdős, A. Poursaeidesfahani, H. S. Salehi, A. T. Celebi, M. Ramdin, O. A. Moultos, D. Dubbeldam, and

30 September 2024 11:38:08

© Author(s) 2024

T. J. H. Vlugt, "Brick-CFCMC: Open source software for Monte Carlo simulations of phase and reaction equilibria using the continuous fractional component method," J. Chem. Inf. Model. **60**, 2678–2682 (2020).
[15]H. M. Polat, H. S. Salehi, R. Hens, D. O. Wasik, A. Rahbari, F. de Meyer, C. Houriez, C. Coquelet, S. Calero, D. Dubbeldam, O. A. Moultos, and T. J. H. Vlugt, "New features of the open source Monte Carlo software Brick-CFCMC: Thermodynamic integration and hybrid trial moves," J. Chem. Inf. Model. **61**, 3752–3757 (2021).
[16]D. Dubbeldam, S. Calero, D. E. Ellis, and R. Q. Snurr, "RASPA: Molecular simulation software for adsorption and diffusion in flexible nanoporous materials," Mol. Simul. **42**, 81–101 (2016).
[17]R. Gowers, A. Farmahini, D. Friedrich, and L. Sarkisov, "Automated analysis and benchmarking of gcmc simulation programs in application to gas adsorption," Mol. Simul. **44**, 309–321 (2018).
[18]J. K. Shah, E. Marin-Rimoldi, R. G. Mullen, B. P. Keene, S. Khan, A. S. Paluch, N. Rai, L. L. Romanielo, T. W. Rosch, B. Yoo, and E. J. Maginn, "Cassandra: An open source Monte Carlo package for molecular simulation," J. Comput. Chem. **38**, 1727–1739 (2017).
[19]A. V. Brukhno, J. Grant, T. L. Underwood, K. Stratford, S. C. Parker, J. A. Purton, and N. B. Wilding, "DL_MONTE: A multipurpose code for Monte Carlo simulation," Mol. Simul. **47**, 131–151 (2019).
[20]S. Chempath, T. Düren, and R. Snurr, "Experiences with the publicly available multipurpose simulation code, Music," Mol. Simulat. **39**, 1223–1232 (2013).
[21]M. G. Martin, "MCCCS Towhee: A tool for Monte Carlo molecular simulation," Mol. Simul. **39**, 1212–1222 (2013).
[22]D. Dubbeldam, S. Calero, and T. J. H. Vlugt, "iRASPA: GPU-accelerated visualization software for materials scientists," Mol. Simul. **44**, 653–676 (2018).
[23]S. Sharma, S. R. G. Balestra, R. Baur, U. Agarwal, E. Zuidema, M. S. Rigutto, S. Calero, T. J. H. Vlugt, and D. Dubbeldam, "RUPTURA: Simulation code for breakthrough, ideal adsorption solution theory computations, and fitting of isotherm models," Mol. Simul. **49**, 893–953 (2023).
[24]A. Dabrowski, "Adsorption - From theory to practice," Adv. Colloid Interface Sci. **93**, 135–224 (2001).
[25]T. J. H. Vlugt, W. Zhu, F. Kapteijn, J. A. Moulijn, B. Smit, and R. Krishna, "Adsorption of linear and branched alkanes in the zeolite silicalite-1," J. Am. Chem. Soc. **120**, 5599–5600 (1998).
[26]T. J. H. Vlugt, R. Krishna, and B. Smit, "Molecular simulations of adsorption isotherms for linear and branched alkanes and their mixtures in silicalite," J. Phys. Chem. B **103**, 1102–1118 (1999).
[27]K. S. Walton, A. R. Millward, D. Dubbeldam, H. Frost, J. J. Low, O. M. Yaghi, and R. Q. Snurr, "Understanding inflections and steps in carbon dioxide adsorption isotherms in metal-organic frameworks," J. Am. Chem. Soc. **130**, 406–407 (2008).
[28]H. van Koningsveld, H. van Bekkum, and J. C. Jansen, "On the location and disorder of the tetrapropylammonium (TPA) ion in zeolite ZSM-5 with improved framework accuracy," Acta Crystallogr., Sect. B: Struct. Sci. **43**, 127–132 (1987).
[29]S. Chui, S. Lo, J. Charmant, A. Orpen, and I. Williams, "A chemically functionalizable nanoporous material $[Cu_3(TMA)_2(H_2O)_3]_n$," Science **283**, 1148–1150 (1999).
[30]D. Dubbeldam, C. G. K. Walton, D. Ellis, and R. Snurr, "Separation and molecular-level segregation of complex alkane mixtures in metal–organic frameworks," J. Am. Chem. Soc. **130**, 10884–10885 (2008).
[31]Y. Hu, Y. Jiang, J. Li, L. Wang, M. Steiner, R. F. Neumann, B. Luan, and Y. Zhang, "New-generation anion-pillared metal–organic frameworks with customized cages for highly efficient CO_2 capture," Adv. Funct. Mater. **33**, 2213915 (2023).
[32]S. J. Datta, A. Mayoral, N. Murthy Srivatsa Bettahalli, P. M. Bhatt, M. Karunakaran, I. D. Carja, D. Fan, P. Graziane M Mileo, R. Semino, G. Maurin, O. Terasaki, and M. Eddaoudi, "Rational design of mixed-matrix metal-organic framework membranes for molecular separations," Science **376**, 1080–1087 (2022).
[33]C. Yu, H. Li, Y. Wang, J. Suo, X. Guan, R. Wang, V. Valtchev, Y. Yan, S. Qiu, and Q. Fang, "Three-dimensional triptycene-functionalized covalent organic frameworks with hea net for hydrogen adsorption," Angew. Chem., Int. Ed. **61**, e202117101 (2022).
[34]Z. Chen, P. Li, R. Anderson, X. Wang, X. Zhang, L. Robison, L. R. Redfern, S. Moribe, T. Islamoglu, D. A. Gómez-Gualdrón, T. Yildirim, J. F. Stoddart, and O. K. Farha, "Balancing volumetric and gravimetric uptake in highly porous materials for clean energy," Science **368**, 297–303 (2020).
[35]L. Liang, C. Liu, F. Jiang, Q. Chen, L. Zhang, H. Xue, H.-L. Jiang, J. Qian, D. Yuan, and M. Hong, "Carbon dioxide capture and conversion by an acid-base resistant metal-organic framework," Nat. Commun. **8**, 1233 (2017).
[36]A. Pulido, L. Chen, T. Kaczorowski, D. Holden, M. A. Little, S. Y. Chong, B. J. Slater, D. P. McMahon, B. Bonillo, C. J. Stackhouse, A. Stephenson, C. M. Kane, R. Clowes, T. Hasell, A. I. Cooper, and G. M. Day, "Functional materials discovery using energy–structure–function maps," Nature **543**, 657–664 (2017).
[37]Y. Chen, Z. Qiao, H. Wu, D. Lv, R. Shi, Q. Xia, J. Zhou, and Z. Li, "An ethane-trapping MOF PCN-250 for highly selective adsorption of ethane over ethylene," Chem. Eng. Sci. **175**, 110–117 (2018).
[38]Q. Ke, X. Gong, S. Liao, C. Duan, and L. Li, "Effects of thermostats/barostats on physical properties of liquids by molecular dynamics simulations," J. Mol. Liq. **365**, 120116 (2022).
[39]X. Wang, Q. Lyu, T. Tong, K. Sun, L.-C. Lin, C. Y. Tang, F. Yang, M. D. Guiver, X. Quan, and Y. Dong, "Robust ultrathin nanoporous MOF membrane with intra-crystalline defects for fast water transport," Nat. Commun. **13**, 266 (2022).
[40]H. Wang, L. Chen, Z. Qu, Y. Yin, Q. Kang, B. Yu, and W.-Q. Tao, "Modeling of multi-scale transport phenomena in shale gas production—A critical review," Appl. Energy **262**, 114575 (2020).
[41]M. Liu, L. Zhang, M. A. Little, V. Kapil, M. Ceriotti, S. Yang, L. Ding, D. L. Holden, R. Balderas-Xicohténcatl, D. He, R. Clowes, S. Y. Chong, G. Schütz, L. Chen, M. Hirscher, and A. I. Cooper, "Barely porous organic cages for hydrogen isotope separation," Science **366**, 613–620 (2019).
[42]D. A. Gómez-Gualdrón, P. Z. Moghadam, J. T. Hupp, O. K. Farha, and R. Q. Snurr, "Application of consistency criteria to calculate BET areas of micro- and mesoporous metal–organic frameworks," J. Am. Chem. Soc. **138**, 215–224 (2016).
[43]Y. G. Chung, E. Haldoupis, B. J. Bucior, M. Haranczyk, S. Lee, H. Zhang, K. D. Vogiatzis, M. Milisavljevic, S. Ling, J. S. Camp, B. Slater, J. I. Siepmann, D. S. Sholl, and R. Q. Snurr, "Advances, updates, and analytics for the computation-ready, experimental metal–organic framework database: CoRE MOF 2019," J. Chem. Eng. Data **64**, 5985–5998 (2019).
[44]A. H. Farmahini, S. Krishnamurthy, D. Friedrich, S. Brandani, and L. Sarkisov, "Performance-based screening of porous materials for carbon capture," Chem. Rev. **121**, 10666–10741 (2021).
[45]S. M. Moosavi, A. Nandy, K. M. Jablonka, D. Ongari, J. P. Janet, P. G. Boyd, Y. Lee, B. Smit, and H. J. Kulik, "Understanding the diversity of the metal-organic framework ecosystem," Nat. Commun. **11**, 4068 (2020).
[46]A. Ahmed, S. Seth, J. Purewal, A. G. Wong-Foy, M. Veenstra, A. J. Matzger, and D. J. Siegel, "Exceptional hydrogen storage achieved by screening nearly half a million metal-organic frameworks," Nat. Commun. **10**, 1568 (2019).
[47]B. J. Bucior, N. S. Bobbitt, T. Islamoglu, S. Goswami, A. Gopalan, T. Yildirim, O. K. Farha, N. Bagheri, and R. Q. Snurr, "Energy-based descriptors to rapidly predict hydrogen storage in metal–organic frameworks," Mol. Syst. Des. Eng. **4**, 162–174 (2019).
[48]Y. J. Colón, D. A. Gómez-Gualdrón, and R. Q. Snurr, "Topologically guided, automated construction of metal–organic frameworks and their evaluation for energy-related applications," Cryst. Growth Des. **17**, 5801–5810 (2017).
[49]Y. Kang, H. Park, B. Smit, and J. Kim, "A multi-modal pre-training transformer for universal transfer learning in metal-organic frameworks," Nat. Mach. Intell. **5**, 309 (2023).
[50]Z. Yao, B. Sánchez-Lengeling, N. S. Bobbitt, B. J. Bucior, S. G. H. Kumar, S. P. Collins, T. Burns, T. K. Woo, O. K. Farha, R. Q. Snurr, and A. Aspuru-Guzik, "Inverse design of nanoporous crystalline reticular materials with deep generative models," Nat. Mach. Intell. **3**, 76–86 (2021).
[51]N. Beauregard, M. Pardakhti, and R. Srivastava, "In silico evolution of high-performing metal organic frameworks for methane adsorption," J. Chem. Inf. Model. **61**, 3232–3239 (2021).
[52]P. Chen, R. Jiao, J. Liu, Y. Liu, and Y. Lu, "Interpretable graph transformer network for predicting adsorption isotherms of metal–organic frameworks," J. Chem. Inf. Model. **62**, 5446–5456 (2022).
[53]N. Lohmann, *json*, https://github.com/nlohmann/json.

30 September 2024 11:38:08

© Author(s) 2024

[54]J. D. Evans, V. Bon, I. Senkovska, and S. Kaskel, "A universal standard archive file for adsorption data," Langmuir **37**, 4222–4226 (2021).
[55]Google, Google Benchmark, https://github.com/google/benchmark (2013).
[56]Google, Google Test, https://github.com/google/googletest (2008).
[57]W. Jakob, J. Rhinelander, and D. Moldovan, *pybind11 – seamless operability between c++11 and python*, https://github.com/pybind/pybind11.
[58]The HDF Group, Hierarchical Data Format, version 5, https://github.com/HDFGroup/hdf5.
[59]T. Miller III, M. Eleftheriou, P. Pattnaik, A. Ndirango, D. Newns, and G. Martyna, "Symplectic quaternion scheme for biophysical molecular dynamics," J. Chem. Phys. **116**, 8649–8659 (2002).
[60]M. Jorge, N. Garrido, A. Queimada, I. Economou, and E. Macedo, "Effect of the integration method on the accuracy and computational efficiency of free energy calculations using thermodynamic integration," J. Chem. Theory Comput. **6**, 1018–1027 (2010).
[61]D. W. Siderius and V. K. Shen, "Use of the grand canonical transition-matrix Monte Carlo method to model gas adsorption in porous materials," J. Phys. Chem. C **117**, 5861–5872 (2013).
[62]Z. Li, K. Shi, D. Dubbeldam, M. Dewing, C. Knight, A. V. Mayagoitia, and R. Snurr, "Efficient implementation of Monte Carlo algorithms on graphical processing units for simulation of adsorption in porous materials," J. Chem. Theory Comput. (submitted).
[63]J. L. Lebowitz and J. K. Percus, "Asymptotic behavior of the radial distribution function," J. Math. Phys. **4**, 248–254 (1963).
[64]B. Rotenberg, "Use the force! reduced variance estimators for densities, radial distribution functions, and local mobilities in molecular simulations," J. Chem. Phys. **153**, 150902 (2020).
[65]D. Borgis, R. Assaraf, B. Rotenberg, and R. Vuilleumier, "Computation of pair distribution functions and three-dimensional densities with a reduced variance principle," Mol. Phys. **111**, 3486–3492 (2013).
[66]D. de las Heras and M. Schmidt, "Better than counting: Density profiles from force sampling," Phys. Rev. Lett. **120**, 218001 (2018).
[67]R. Q. Snurr, R. L. June, A. T. Bell, and D. N. Theodorou, "Molecular simulations of methane adsorption in silicalite," Mol. Simul. **8**, 73–92 (1991).
[68]A. Rahbari, R. Hens, D. Dubbeldam, and T. J. H. Vlugt, "Improving the accuracy of computing chemical potentials in CFCMC simulations," Mol. Phys. **117**, 3493–3508 (2019).
[69]S. M. Walas, *Phase Equilibria in Chemical Engineering* (Butterworth, Boston, 1985).
[70]M. G. Martin and J. I. Siepmann, "Transferable potentials for phase equilibria. 1. United-atom description of *n*-alkanes," J. Phys. Chem. B **102**, 2569 (1998).
[71]M. G. Martin and J. I. Siepmann, "Novel configurational-bias Monte Carlo method for branched molecules. Transferable potentials for phase equilibria. 2. United-atom description of branched alkanes," J. Phys. Chem. B **103**, 4508 (1999).
[72]N. Rai and J. I. Siepmann, "Transferable potentials for phase equilibria. 9. Explicit hydrogen description of benzene and five-membered and six-membered heterocyclic aromatic compounds," J. Phys. Chem. B **111**, 10790 (2007).
[73]J. M. Stubbs, J. J. Potoff, and J. I. Siepmann, "Transferable potentials for phase equilibria. 6. United-atom description for ethers, glycols, ketones, and aldehydes," J. Phys. Chem. B **108**, 17596 (2004).
[74]C. D. Wick, M. G. Martin, and J. I. Siepmann, "Transferable potentials for phase equilibria. 4. United-atom description of linear and branched alkenes and alkylbenzenes," J. Phys. Chem. B **104**, 8008 (2000).
[75]C. D. Wick, J. M. Stubbs, N. Rai, and J. I. Siepmann, "Transferable potentials for phase equilibria. 7. Primary, secondary, and tertiary amines, nitroalkanes and nitrobenzene, nitriles, amides, pyridine, and pyrimidine," J. Phys. Chem. B **109**, 18974 (2005).
[76]K. S. Rane, S. Murali, and J. R. Errington, "Monte Carlo simulation methods for computing liquid-vapor saturation properties of model systems," J. Chem. Theory Comput. **9**, 2552–2566 (2013).
[77]F. A. Escobedo and J. J. de Pablo, "Monte Carlo simulation of the chemical potential of polymers in an expanded ensemble," J. Chem. Phys. **103**, 2703–2710 (1995).
[78]A. P. Lyubartsev, A. A. Martsinovski, S. V. Shevkunov, and P. N. Vorontsov-Velyaminov, "New approach to Monte Carlo calculation of the free energy: Method of expanded ensembles," J. Chem. Phys. **96**, 1776–1783 (1992).
[79]M. Tuckerman, *Statistical Mechanics: Theory and Molecular Simulations* (Oxford University Press, New York, 2010).
[80]G. J. Martyna, M. Tuckerman, D. J. Tobias, and M. L. Klein, "Explicit reversible integrators for extended systems dynamics," Mol. Phys. **87**, 1117–1157 (1996).
[81]M. E. Tuckerman, J. Alejandre, R. Lopez-Rendon, A. L. Jochim, and G. J. Martyna, "A Liouville-operator derived measure-preserving integrator for molecular dynamics simulations in the isothermal–isobaric ensemble," J. Phys. A: Math. Gen. **39**, 5629–5651 (2006).
[82]T.-Q. Yu, J. Alejandre, R. Lopez-Rendon, G. J. Martyna, and M. E. Tuckerman, "Measure-preserving integrators for molecular dynamics in the isothermal-isobaric ensemble derived from the Liouville operator," Chem. Phys. **370**, 294–305 (2010).
[83]D. Dubbeldam, D. Ford, D. Ellis, and R. Snurr, "A new perspective on the order-*n* algorithm for computing correlation functions," Mol. Simul. **35**, 1084–1097 (2009).
[84]A. Skoulidas and D. Sholl, "Self-diffusion and transport diffusion of light gases in metal-organic framework materials assessed using molecular dynamics simulations," J. Phys. Chem. B **109**, 15760–15768 (2005).

30 September 2024 11:38:08

© Author(s) 2024