ACS OMEGA

http://pubs.acs.org/journal/acsodf

Article

Eleven NanoHUB Simulation Tools Using RASPA Software To Demonstrate Classical Atomistic Simulations of Fluids and Nanoporous Materials

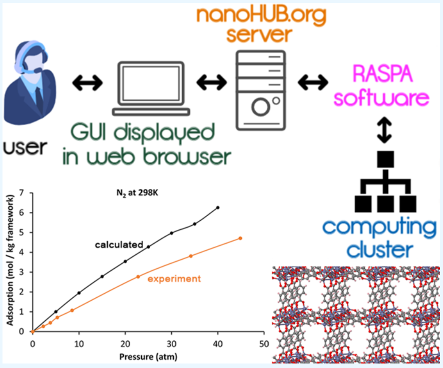
Julian C. Umeh and Thomas A. Manz*

Cite This: *ACS Omega* 2022, 7, 44470−44484

Read Online

ACCESS | Metrics & More | Article Recommendations

ABSTRACT: Eleven interactive simulation tools were created on nanoHUB to help users learn how to perform classical atomistic simulations. These tools enable users to perform classical Monte Carlo and molecular dynamics simulations using RASPA software. These tools use comparatively small numbers of production cycles to keep the runtimes short, so that users will not be discouraged by long wait times to see results. Here, we show that these tools produce results of sufficient accuracy and reproducibility for learning purposes. The 11 tools developed were as follows: (1) calculation of the self-diffusion constant of gas molecules in metal−organic frameworks (MOFs), (2) gas adsorption in MOFs using the grand canonical ensemble, (3) Henry's coefficient calculator for gas molecules in MOFs and a zeolite, (4) adsorption of a gas mixture in a MOF, (5) self-diffusion of a gas mixture in a MOF, (6) void fraction calculation for several MOFs and zeolites, (7) surface area calculation for several MOFs and zeolites, (8) calculation of radial distribution function and self-diffusion constant for several pure gases, (9) energy distribution of adsorption sites using a probe molecule in MOFs, (10) molecular dynamics simulation of pure fluids in the *NPT* ensemble, and (11) gas adsorption in MOFs using the Gibbs ensemble.



1. INTRODUCTION

In this work, we developed and deployed tools on nanoHUB with capabilities to perform atomistic simulations and to output results to a graphical user interface (GUI) using RASPA software package on the backend. NanoHUB is an online platform for science and engineering originally funded by the National Science Foundation (NSF). It contains simulation tools and resources developed and contributed by members of the community.[1] It promotes the advancement of nanoscience and nanotechnology and professional collaboration. NanoHUB is a learning resource for both undergraduate and graduate students in science and engineering.[2] NanoHUB is a product of the Network for Computational Nanotechnology, which aids research efforts in the area of nanoscience and nanotechnology. NanoHUB can be accessed through the web portal nanohub.org. It contains over 500 simulation tools and has over 2500 citations in the scientific literature.[3] NanoHUB users run these simulation tools in their web browser as applets, while the actual calculations in these simulations are run transparently on the nanoHUB backend cloud Linux cluster.

RASPA software performs classical atomistic simulations of adsorption and diffusion of molecules in nanoporous materials such as carbon nanotubes, metal−organic frameworks (MOFs), and zeolites.[4] It can also be used to compute single-phase properties as well as phase equilibrium properties such as vapor−liquid equilibrium. It uses cutting-edge algorithms to perform molecular dynamics and Monte Carlo calculations in various thermodynamic ensembles.[4] Monte Carlo and molecular dynamics methods can be combined to form more advanced simulations for specialized applications such as gas adsorption in flexible MOFs.[5]

In this work, we created interactive nanoHUB educational modules that demonstrate basic inputs, outputs, and methods for classical atomistic simulations using RASPA software. These simulation tools should catalyze the interest of new students and researchers to study diffusion constants, adsorption isotherms, and fluid properties by educating them on how to use RASPA to perform these types of calculations. To the best of our knowledge, this work is the first time RASPA has been deployed within nanoHUB tools.

Our 11 new nanoHUB simulation modules do not involve installing any software on the user's computer. They are web

Received: October 30, 2022
Accepted: November 8, 2022
Published: November 23, 2022

applets accessed and run through a normal web browser. The user enters all inputs by selecting options in a GUI that is located within a webpage on the nanoHUB.org website, and all of the simulation module's outputs are returned to the user through this same GUI running as a web applet within the webpage. In effect, these simulation modules appear to the user as interactive webpages. The only requirements for running these simulation tools is that the user must have any web browser installed on their computer, and they must sign up for a free nanoHUB.org login username. These simulation tools do not require RASPA to be installed on the user's computer because the RASPA simulations are run via the cloud on nanoHUB's own computing cluster.

These 11 new nanoHUB simulation modules do not collect any information from the user or from the user's computer, except which menu options have been selected by the user for the purposes of running the simulation. Separately, nanoHUB.org may collect certain user information as part of their privacy policy described at http://nanohub.org/legal/privacy. We reiterate that information is collected by and accessible to nanoHUB.org and not to us as the tool developers. As tool developers, we do not collect any information from or about the user or from the user's computer.

During review of this article, we received an inquiry about whether it is possible to install these modules as a "standalone software". The answer is that these modules are run as interactive webpages on the nanoHUB.org website. They are not a software program that could be installed on a user's computer.

These 11 new nanoHUB simulation modules are intended as educational tools, not as research tools. They show users how to set up and perform these types of simulations and to interpret the results. To simplify both their use and testing, these simulation modules were deliberately constructed with limited menu options. Because these menu options are very limited, these simulation tools are unsuitable for scientific research calculations. Each simulation tool has an online documentation that includes a PowerPoint presentation that briefly summarizes the choices of inputs, the classical forcefields used, the simulation methods, and how to interpret the simulation results. The online documentation also includes an example RASPA input file for each tool; this helps each user learn how to set up a RASPA calculation in case they want to later set up their own independent RASPA simulations for research purposes. In this way, we hope to promote the wider use and understanding of classical atomistic calculations for simulating gas adsorption and diffusion in porous materials and for simulating fluids. We envision the following use cases.

Use case # 1: In a college course on computational chemistry, the instructor could perform in-class demonstrations using these tools. For example, the instructor could run an in-class molecular dynamics simulation using tool 1 or tool 5 to show students how to compute gas self-diffusivities in porous materials. This helps students learn four important aspects: (a) What are the required inputs for this calculation type? (b) What computational methods and thermodynamic ensemble (if any) are used? (c) What are the raw outputs from the calculation? and (d) How are these outputs analyzed to compute the desired properties? For example, graphical analysis is performed to extract the self-diffusivity from the mean-squared-displacement (MSD) versus time data. Furthermore, the instructor could assign homework problems that use one or more of these simulation tools.

A key benefit is that these tools can be used without requiring users to install RASPA software. When teaching a course, it can be challenging to require students to install new software. First, some students in the class might not own a personal computer and instead use various on-campus computer labs. Second, different operating systems and versions present challenges to software installation. All of these difficulties are avoided because these simulation tools do not require installing any software beyond a normal web browser that is already available on the user's computer.

Variations of use case # 1 also include seminars, workshops, and short courses that last only 1−3 days rather than a full academic semester. These sometimes occur at conferences or other venues that bring together people from various industries, universities, and/or countries.

Use case # 2: A person can teach themselves the basics of classical molecular dynamics and Monte Carlo simulations by using these simulation tools. Many people who do not have access to a dedicated course on computational chemistry may wish to learn about the topic through self-study. Some of these users may not have access to high performance computing clusters. Because all of these simulations are run via the cloud on nanoHUB's own computing cluster, a person who does not have access to high performance computing can still easily run them.

Use case # 3: These simulation tools can facilitate initial training of graduate students that will subsequently progress to doing research calculations using RASPA. In this case, the tools demonstrate example inputs, settings, outputs, and data analysis for common calculation types. For research calculations, graduate students can then run simulations for other molecules and materials using the standalone RASPA program, which does not involve our simulation tools in any way. Typically, the standalone RASPA program would be installed on a Linux cluster used by the research group. Research calculations should use more initialization, equilibration (for molecular dynamics), and production cycles than what is used in our simulation tools because the numbers of cycles are purposefully kept low in our simulation tools to allow them to complete faster for demonstration purposes.

2. METHODS

2.1. Classical Forcefields Used. Forcefields are functional forms with parameters used to specify the potential energy of a system of atoms.[6−8] The basic functional form of potential energy in molecular mechanics involves the bonded and non-bonded interactions which can be expressed mathematically as

$$U = U_{\text{bonded}} + U_{\text{nonbonded}} \tag{1}$$

$$U_{\text{bonded}} = U_{\text{R}} + U_{\theta} + U_{\varphi} + U_{\omega} \tag{2}$$

$$U_{\text{nonbonded}} = U_{\text{vdw}} + U_{\text{el}} \tag{3}$$

where U_{R} is the bond stretch interaction, U_{θ} is the bond angle bending, U_{φ} is the dihedral angle torsion, U_{ω} is the inversion term, U_{vdw} is the van der Waals energy (e.g., Lennard-Jones potential, which includes short-range exchange repulsion and long-range attractive dispersion interactions), and U_{el} is the electrostatic energy.[9] The bonded terms are used to specify interactions between atoms linked by a covalent bond, while the non-bonded interactions, on the other hand, consist of the electrostatic term and the van der Waals interaction.

To simplify computations, the non-bonded terms are often limited to pairwise interactions. The Lennard-Jones potential can be used as a model for the van der Waals interactions[10,11]

$$V_{\mathrm{LJ}} = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] \tag{4}$$

where ε is the depth of the potential well, σ is the finite distance at which the inter-particle potential is zero, and r is the distance between the two atoms or molecules. A typical potential energy curve for two interacting molecules is shown in Figure 1. r_{min} is the distance at which the potential reaches its minimum and is related to σ by

$$r_{\mathrm{min}} = 2^{1/6}\sigma \tag{5}$$

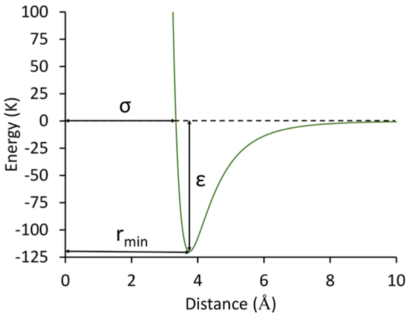


Figure 1. Lennard-Jones potential for argon. The energy unit is in Kelvin (used by RASPA); this can be multiplied by the Boltzmann constant to convert to Joules per particle.

Interactions between dissimilar non-bonded atoms require a combining rule for the part of the potential representing the van der Waals interaction.[12] For this project, the Lorentz–Berthelot[13,14] mixing rules were used, which assume the molecular size and energy parameters to be the arithmetic and geometric averages, respectively

$$\sigma_{ij} = \frac{\sigma_{ii} + \sigma_{jj}}{2} \tag{6}$$

$$\varepsilon_{ij} = \sqrt{\varepsilon_{ii}\varepsilon_{jj}} \tag{7}$$

RASPA software comes with different forcefields built into the program.[4,15]Table 1 lists the different forcefields we used for various systems in the 11 simulation tools. For IRMOF-1 and IRMOF-16, the forcefield reported by Dubbeldam et al. was used keeping the framework rigid.[16] For the other MOFs, the GenericMOFs forcefield built into RASPA was used.[15] For the zeolites, the GenericZeolites forcefield built into RASPA was used.[15]

Table 1. List of Forcefields Used in the Different NanoHUB Tools

system	forcefield	flexible	ref	tools used in
Ar	Lennard-Jones	N	[a]	1, 4, 5, 11
H_2	(GenericMOFs)[b]	N	21	1, 2, 9, 11
N_2	TraPPE	N	20	1, 2, 8, 9, 11
O_2	TraPPE	N	22	2
CO_2	TraPPE	N	20	1, 2, 8, 9, 10, 11
CH_4	TraPPE-UA	N	19	1, 2, 4, 5, 7–11
C_2H_6	TraPPE-UA	Y	19	8, 9, 10
C_3H_8	TraPPE-UA	Y	19	8, 10
C_4H_{10}	TraPPE-UA	Y	19	8
n-pentane	TraPPE-UA	Y	19	3
n-hexane	TraPPE-UA	Y	19	3
n-heptane	TraPPE-UA	Y	19	3
n-octane	TraPPE-UA	Y	19	3
n-nonane	TraPPE-UA	Y	19	3
IRMOF-1	Dubbeldam	N[c]	16	1–7, 9, 11
IRMOF-2	(GenericMOFs)[d]	N	[9,23]	7
IRMOF-3	(GenericMOFs)[d]	N	[9,23]	7
IRMOF-12	(GenericMOFs)[d]	N	[9,23]	7
IRMOF-16	Dubbeldam	N[c]	16	1–3, 6, 7, 9, 11
ITQ-1	GenericZeolites	N	[e]	6, 7
ITQ-3	GenericZeolites	N	[e]	6, 7
ITQ-7	GenericZeolites	N	[e]	6, 7
ITQ-12	GenericZeolites	N	[e]	6, 7
ITQ-29	GenericZeolites	N	[e]	6, 7
KFI	GenericZeolites	N	[e]	6, 7
LTA4A	GenericZeolites	N	[e]	6, 7
LTA5A	GenericZeolites	N	[e]	6, 7
MFI_SI	GenericZeolites	N	[e]	3

[a] Lennard-Jones parameters for argon were set to σ = 3.34 Å and ε = 119.8 K to reproduce the well depth and $r_{\mathrm{min}} = 2^{1/6}\,\sigma$ value for the dimer curve. [b] These parameters for the H_2 molecule are part of the GenericMOFs forcefield collection in RASPA. [c] This forcefield included flexibility but it was not used. [d] IRMOF-2, IRMOF-3, and IRMOF-12 used generic atom types and parameters from the UFF and Dreiding forcefields, as compiled within the GenericMOFs forcefield collection in RASPA. [e] The GenericZeolites forcefield in RASPA uses Lennard-Jones parameter values collected from different sources: (a) for Si and O atoms, the GenericZeolites forcefield uses TraPPE-zeo (ref 24) Lennard-Jones parameter values, (b) for Al atoms, the GenericZeolites forcefield used the same Lennard-Jones parameter values as for Si, (c) although these atoms are not present in any of the structures here, for Na and Ca atoms, the GenericZeolites forcefield uses UFF (ref 9) Lennard-Jones parameter values.

We used Transferrable Potential for Phase Equilibria (TraPPE) forcefields to describe the molecules. TraPPE forcefields are popular and suitable for research and industrial applications because they have reasonably high accuracy for predicting thermophysical properties of various compounds across different physical states and compositions.[17−20] TraPPE forcefields are available in RASPA in two main flavors: TraPPE-United Atom (TraPPE-UA) model and TraPPE-Explicit Hydrogen (TraPPE-EH) model. The TraPPE-UA forcefield is a united atom representation for the alkyl groups; that is, each hydrogen atom is modeled implicitly in the same interaction site as the carbon atom it is bonded to.[19] The TraPPE-EH forcefield has a separate interaction site for each hydrogen atom.[17]

2.2. Monte Carlo, Molecular Dynamics, and Widom Insertion Simulations in RASPA Software.

RASPA is a classical atomistic simulation software.[4,15] In this work, we used RASPA version 2 (i.e., RASPA2).[15] RASPA uses state-of-the-art algorithms to perform molecular dynamics and Monte Carlo calculations in various statistical ensembles.[4,25] Monte Carlo and molecular dynamics methods can be combined to form more advanced simulations for specialized applications such as adsorption on a swelling material, protein studies, lipids studies, and so on.[4] RASPA can perform simulations with flexible frameworks,[4,7,26,27] but to minimize the computational time, all frameworks in the nanoHUB tools described in this work were held rigid.

To perform a simulation in RASPA, certain parameters can be specified in the input file[15]

- The simulation type (e.g., Monte Carlo or Molecular Dynamics)
- The number of initialization, equilibration, and production cycles
- The molecules and/or framework that make up the system to be simulated
- The forcefields and mixing rules to be used
- If a framework is included, whether framework flexibility is allowed or the framework is to be held rigid
- For Monte Carlo calculations, the different kinds of trial moves and their probabilities
- Keywords to specify various quantities to be computed and printed
- Depending on the simulation type, the external temperature and/or pressure may need to be specified
- The cutoff radius for interactions between atoms

RASPA2 uses the minimum image convention. The minimum image convention assumes that each atom type "A" in a unit cell only interacts with the closest image of each atom type "B".[25] Also, interactions are neglected between atoms that are farther apart than the cutoff radius.[25] The cutoff radius must be less than half the shortest length of the unit cell.

Simulations that use a single probe particle test insertion (not a thermodynamic ensemble) do not need initialization cycles. This type of move is usually called Widom insertion.[4] Each Widom insertion move starts with no molecules present in the framework and then attempts the test insertion of a probe molecule. Even for this kind of test insertion, RASPA grows the internal configuration of flexible molecules using configurational bias Monte Carlo (CBMC).[4] In this work, the test insertion of a probe molecule was used to compute surface areas, void fractions, and Henry's coefficients.

Monte Carlo algorithms use random sampling and inferential statistics to estimate the value of an unknown quantity.[25,28,29] In RASPA, one Monte Carlo step refers to a singular trial move (e.g., translation, rotation, swap, reinsertion, etc.), whereas a Monte Carlo cycle refers to the maximum between 20 move attempts and the number of particles that make up the system.[4,25] Each attempted Monte Carlo trial move has the following sequence. First, the type of move to attempt (i.e., translation, rotation, reinsertion, etc.) is randomly selected based on their assigned probabilities. Second, a trial move of the selected type will be prepared. For a translation or rotation move, the particle on which the move acts is randomly selected, and the direction of the translation or rotation is also randomly selected. If the move is a volume change, then a particle does not need to be selected, but the size of the volume change needs to be randomly selected. Third, the energy change of the system that would be caused by the trial move is computed. Fourth, acceptance criteria are used to determine whether to accept or reject the trial move.[28,29] If the trial move is accepted, then the modified structure which includes the trial move forms the next system configuration in the Markov chain. If the trial move is rejected, then the unmodified structure which does not include the trial move forms the next system configuration in the Markov chain; in other words, the new configuration retains the configuration of the previous move.

For a simple translation or a simple rotation of a molecule, the acceptance criterion is related to the Boltzmann factor $\exp[-\Delta E/(kT)]$ computed for the attempted move. ΔE is the energy of the new configuration minus the energy of the old configuration, k is the Boltzmann constant, and T is the absolute temperature of the system. For a simple translation or a simple rotation of a molecule, the Boltzmann factor expresses the relative probability that the attempted move will be accepted. This can be implemented using the Metropolis–Hastings algorithm[30,31] in which a random number is generated from a uniform distribution between 0 and 1. If the Boltzmann factor is less than the random number, the move is rejected. If the Boltzmann factor is greater or equal to the random number, the move is accepted.

The acceptance criteria for more complicated moves (e.g., volume changes, swaps, reinsertions, etc.) is more complicated and depends on the type of biasing (if any) that is used. A detailed discussion of acceptance criteria for these moves can be found in the literature.[28,29]

Monte Carlo moves used by one or more of our 11 nanoHUB simulation tools included the following:[4,15,25]

- Translation: This move stochastically displaces (translates) a random molecule. The internal configuration of the molecule remains unchanged.
- Rotation: This move randomly selects a molecule and then randomly rotates it about the chosen starting bead. The internal configuration of the molecule remains unchanged. When a molecule is modeled as a sphere, then rotation is not needed. For example, rotation is not needed when running a Monte Carlo simulation of methane using the united-atom model, which models methane as a sphere.
- Swap: This move randomly attempts to insert or delete a random molecule from the system. An insertion is attempted 50% of the time, and a deletion is attempted the other 50% of the time.
- Reinsertion: "A full reinsertion move for the current component."[15] "The 'reinsertion' move removes a randomly selected molecule and reinserts it at a random position. For rigid molecules, it uses orientational biasing, and for chains, the molecule is fully regrown (the internal configuration is modified)."[25] "Multiple first beads are chosen, and one of these is selected according to its Boltzmann weight. The remaining part of the molecule is grown using biasing. This move is very useful, and often necessary, to change the internal configuration of flexible molecules."[15]
- Volume change: This move isotopically changes the volume of the simulation box.
- Gibbs ensemble calculations use two boxes.[28,29] In these computations, a "Gibbs swap" moves a molecule from one box to the other; a new internal configuration is regrown for the molecule using CBMC in the second box.

In RASPA, every Monte Carlo simulation is first initialized at the beginning of the simulation. The initialization cycles warm up the Markov chain.[4,15,30] Data are collected during the production cycles that follow the initialization cycles. Monte Carlo methods can be applied to a variety of molecular studies such as adsorption calculations,[32] vapor−liquid equilibrium calculations,[33] chemical reactions,[34] and so on.

Molecular dynamics calculates the time-dependent behavior of a system. The trajectories of atoms and molecules are calculated numerically by solving Newton's equations of motion for interacting particles.[28,29,35] The potential energy and interaction forces between atoms and molecules are calculated using the forcefield. The timestep is the time interval between adjacent configurations. A 0.5−1.0 femtosecond timestep is appropriate for molecules containing light moveable atoms; a somewhat larger timestep can be used for rigid molecules or if all of the atoms are heavy or if the forcefield is coarse-grained.[36] (We used a timestep of 0.5 femtosecond per MD cycle in tools # 1, 5, 8, and 10.) Three types of cycles are used in a molecular dynamics calculation: initialization, equilibration, and production. In RASPA, initialization cycles are performed at the beginning using Monte Carlo moves to get a statistically relevant configuration of particle positions. Then, equilibration cycles are performed using molecular dynamics moves to get a statistically relevant configuration of particle velocities and positions. Finally, data are collected during the production cycles. Figure 2 shows the types of cycles involved in Monte Carlo and molecular dynamics simulations.

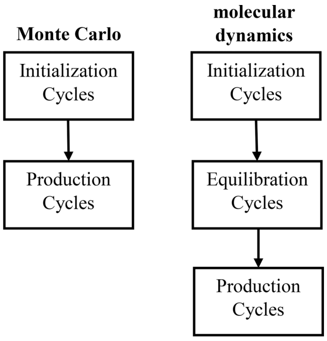


Figure 2. Monte Carlo and molecular dynamics cycles.

Molecular dynamics can be used to study diffusion coefficients (both self-diffusion and collective diffusion),[37,38] interactions between different molecules,[39] material characteristics,[40] protein folding,[41] and so on. In this work, we computed self-diffusion constants rather than collective (also called transport or Fickian) diffusion constants.[38]

Common thermodynamic ensembles used for Monte Carlo and molecular dynamics include the following:[25,28,29]

- Microcanonical ensemble (*NVE*)—This statistical ensemble holds the number of particles *N*, the volume *V*, and the energy *E* constant during the course of the simulation.
- Canonical ensemble (*NVT*)—This ensemble holds the number of particles *N*, the volume *V*, and the average temperature *T* constant.
- Grand Canonical ensemble (μVT)—This ensemble holds the chemical potential μ, the volume *V*, and the average temperature *T* constant.
- Isobaric−isothermal ensemble (*NPT*)—This ensemble holds the number of particles *N*, the average pressure *P*, and the average temperature *T* constant.
- Isoenthalpic−isobaric ensemble (*NPH*)—This ensemble holds the number of particles *N*, the average pressure *P*, and the enthalpy *H* constant.

2.3. Creating NanoHUB Tools That Run RASPA Simulations.

Figure 3 depicts the process flow diagram of nanoHUB tool creation. The nanoHUB infrastructure used to develop tools in this project include the workspace, Rappture toolkit, middleware, and repository.

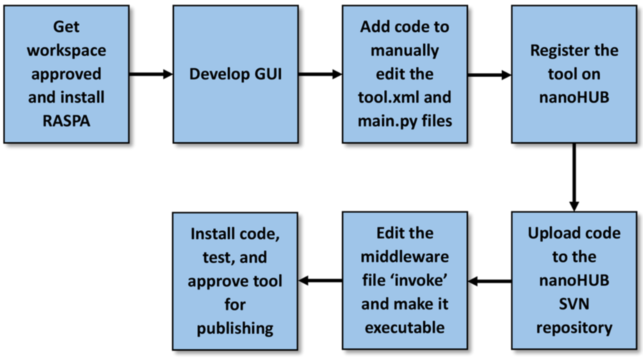
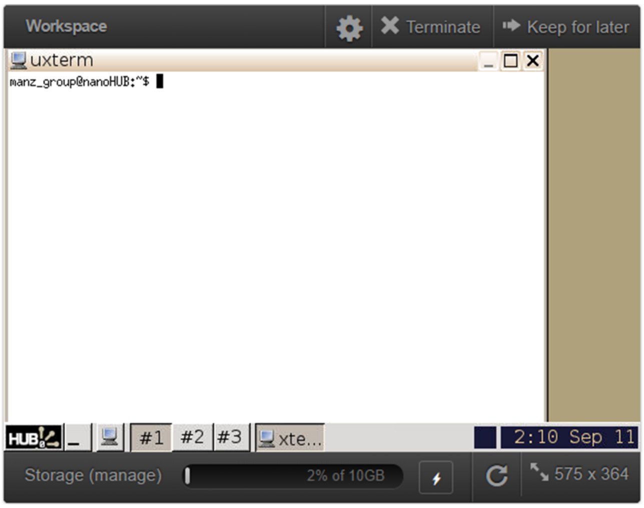

Get workspace approved and install RASPA
Develop GUI
Add code to manually edit the tool.xml and main.py files
Register the tool on nanoHUB
Upload code to the nanoHUB SVN repository
Edit the middleware file 'invoke' and make it executable
Install code, test, and approve tool for publishing

Figure 3. Flow diagram illustrating the process to create a new nanoHUB tool.

As shown in Figure 4, the workspace is an in-browser Linux command line terminal that provides access to computational resources on the Network for Computational Nanotechnology. This workspace can be used to develop code, test, compile, and debug new simulation tools before they are ready to be deployed on nanoHUB. A new developer of nanoHUB simulation tools must submit a special request to the support team to get permission to access this workspace. The nanoHUB support team also installed the RASPA software on the backend computing cluster.



```
manz_group@nanoHUB:~$
```

Figure 4. NanoHUB workspace.

Rappture is an acronym for "rapid application infrastructure". The Rappture toolkit was deployed inside the workspace by typing "rappture -builder" (without quotes) at the Linux command line prompt. Rappture facilitates nanoHUB tool development using drag and drop functionality. It makes tool development a quick and easy process by allowing users to drag and drop from the "Object Types," as shown in Figure 5, to the "Tool Interface" input and output. This allows the developer to add input fields such as dropdown menus, radio buttons, text boxes, and so forth and output fields such as text boxes, graphs, and so forth to the GUI. This was used to develop the GUI for each tool.

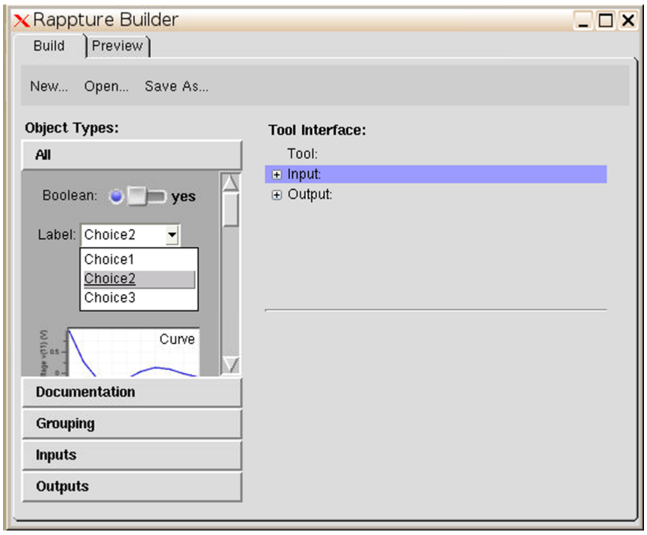
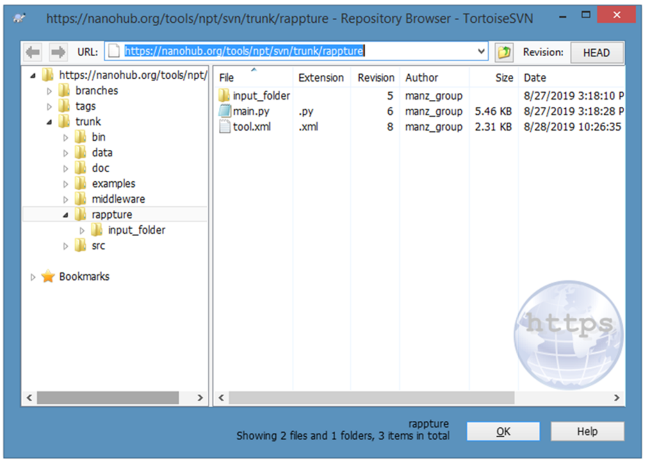


Figure 5. Rappture builder GUI.

Developing a GUI is the first half of building a simulation tool; the other half is writing code within the simulator to access inputs, run calculations, and generate results. Rappture generates two skeleton files to which code is manually added by the developer. The first generated file is tool.xml. This is an extensible markup language file that contains all the input and output fields selected and predefined in the Rappture builder during the process of creating the GUI. The other file generated by Rappture builder is "main.ext"; where "ext" is the extension of the file based on the chosen programming language (e.g., main.py for Python). This file acts as the main program in the simulator. This file contains commands that retrieve the input data from the GUI, generates the RASPA input file, submits the calculation in RASPA, extracts the results from the RASPA output files, and finally sends back the results to the GUI. After these two files (i.e., tool.xml and main.py) are auto generated, the developer opens each of these two files in a text editor and manually adds the desired code lines to perform the different tasks they want. For this project, the main.py script creates the RASPA input file titled "simulation.input" based on the user input specified in the GUI. Then main.py submits the RASPA job and extracts the relevant data from the RASPA output files and returns these to the GUI. Umeh's master's thesis contains several examples of the tool.xml, main.py, and simulation.input files.[42]

After making necessary changes to both the tool.xml and main.py files, the tool is ready to be registered on nanoHUB. Tool registration requires that the developer provide information on the tool name, repository host [e.g., Subversion (SVN) or Github], whether the tool will be public or private, licensing options, and so forth. It is recommended that SVN be used as the repository host for developers working on nanoHUB. SVN tracks and saves all changes made to each file and can revert to a previous version if so desired.

Once the tool is registered, it automatically triggers the creation of a repository for the tool. Figure 6 shows the TortoiseSVN repository browser. After the repository is created, the developer must manually upload the main.py and tool.xml files to the repository.


https://nanohub.org/tools/npt/svn/trunk/rappture - Repository Browser - TortoiseSVN
URL:
https://nanohub.org/tools/npt/svn/trunk/rappture
Revision:
HEAD
https://nanohub.org/tools/npt/
branches
tags
trunk
bin
data
doc
examples
middleware
rappture
input_folder
src
Bookmarks
https
OK
Help

File	Extension	Revision	Author	Size	Date
input_folder		5	manz_group		8/27/2019 3:18:10 P
main.py	.py	6	manz_group	5.46 KB	8/27/2019 3:18:28 P
tool.xml	.xml	8	manz_group	2.31 KB	8/28/2019 10:26:35

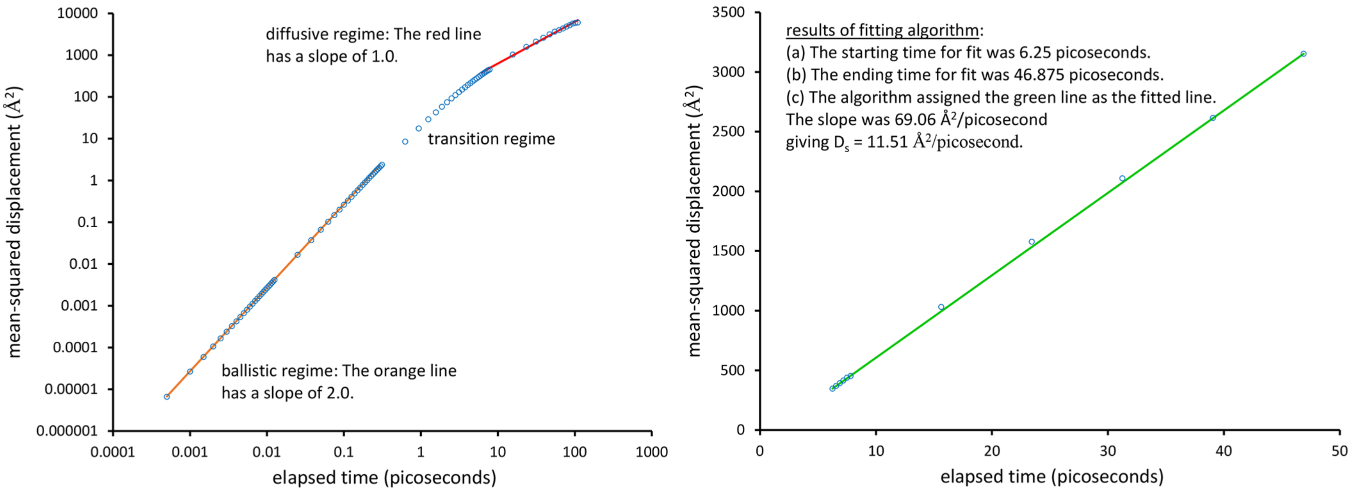
rappture
Showing 2 files and 1 folders, 3 items in total

Figure 6. TortoiseSVN repository browser.

In the newly created repository, the middleware folder contains a file named "invoke" that acts as the interface between the tool.xml and main.py files and the RASPA software. Middleware is the software layer that lies between the operating system and the applications on each side of a distributed computing system in a network. To set up the invoke file, all we had to specify was the simulation tool name (e.g., rdf for the radial distribution function tool) and the RASPA software pathname, and the nanoHUB middleware handles all the interfacing details. For further details, see the master's thesis of Umeh.[42] This invoke file must be made executable.

After the source code has been uploaded and the middleware file edited, the tool is ready to be installed, tested, and published. Clicking the hyperlink "I've committed new code. Please install the latest version for testing and approval." on the nanoHUB tool page alerts support and lets them know that the tool is ready for installation. After the tool is installed, tests should be performed to ensure that it runs as expected. If there are needs for changes to the tool, the changes should be made to the source code on the developer's local repository on their PC and the source code should be recommitted to the repository and the new code installed. This cycle should be repeated until the developer is satisfied with the performance of the tool. Finally, once the tool is working to the satisfaction of the developer, the developer can approve the tool for publishing.

In the tools developed, nanoHUB typically saves the results of the simulation in a cache such that if a simulation is later rerun with exactly the same input options as before, it will display the already computed results rather than redo the same calculation. Because molecular dynamics and Monto Carlo simulations in RASPA use a random number generator, we desire that a new set of randomly generated numbers be used for each replicate run in RASPA. To ensure that a new set of randomly generated numbers is always used rather than displaying nanoHUB's cached results of a prior calculation, we included a line of code in main.py that purges the simulation



diffusive regime: The red line has a slope of 1.0.

transition regime

ballistic regime: The orange line has a slope of 2.0.

results of fitting algorithm:
(a) The starting time for fit was 6.25 picoseconds.
(b) The ending time for fit was 46.875 picoseconds.
(c) The algorithm assigned the green line as the fitted line.
The slope was 69.06 Å²/picosecond
giving D_s = 11.51 Å²/picosecond.

Figure 7. Left panel: A log−log plot of mean-squared displacement versus time that shows the separate ballistic, transition, and diffusive regimes. Right panel: A linear−linear plot of mean-squared displacement versus time using the starting and ending times selected by the algorithm that extracts the self-diffusion constant D_s from a data subset within the diffusive regime.

Table 2. Self-Diffusion of N_2 in IRMOF-16 at 298 K Was Computed Using Tool 1

	initialization cycles	equilibration cycles	production cycles	self-diffusion (Å²/picosecond)
nanoHUB	5000	5000	30,000	11 ± 2
SDSC	50,000	50,000	250,000	11.1 ± 0.6

tool's output cache at the beginning of each nanoHUB simulation.

2.4. Computing a Molecule's Self-Diffusivity. Tools # 1, 5, and 8 calculate and display the self-diffusion constant D_s using Einstein's equation[29]

$$\mathrm{MSD} = 6D_s t + \mathrm{C} \tag{8}$$

where C is the fitted intercept, and MSD is the mean-squared displacement of a molecule during time t. eq 8 describes the relationship between MSD and time in the diffusive regime.

As shown in Figure 7, a plot of MSD versus time can be separated into the following regimes.[38,43] When the time elapsed is much shorter than the mean free time between particle collisions, this kind motion is called ballistic transport (aka "ballistic regime"). In the ballistic regime, log[MSD] versus log[t] has a slope of approximately 2 because each individual particle (e.g., individual molecule) tends to travel at approximately constant speed between collisions, even though the speeds of different particles will be different. The diffusive regime occurs when the time elapsed is much longer than the mean free time between particle collisions. In this case, the trajectory of each particle (i.e., each molecule) resembles a random walk (e.g., Brownian motion) that was studied by Einstein to derive eq 8. Between the ballistic and diffusive regimes is a transition regime in which the time elapsed is approximately similar to the mean free time between collisions.

To compute D_s, we used the following algorithm to find a segment of the MSD versus time curve within the diffusive regime. For each time t_j, a moving average slope of log[MSD] versus log[t] was computed as

$$\mathrm{avg\_slope}_j = \frac{\log[\mathrm{MSD}_{(j+10)}] - \log[\mathrm{MSD}_j]}{\log[t_{(j+10)}] - \log[t_j]} \tag{9}$$

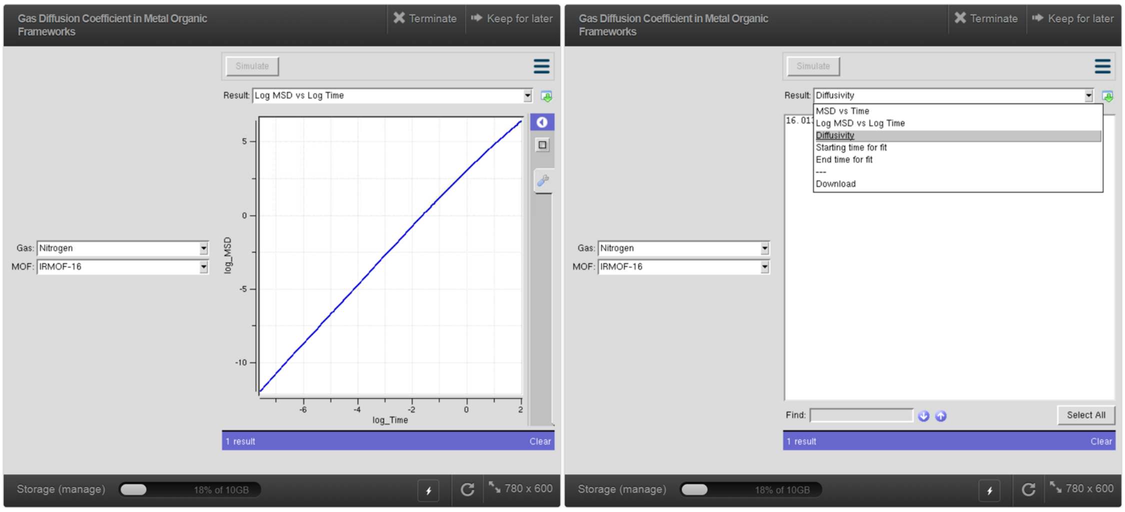
This average slope starts with value 2.00 in the ballistic regime. As t_j increases into the transition regime, avg_slope$_j$ changes until it reaches a value of ∼1 near the start of the diffusive regime. Now, the cutoff of how close avg_slope$_j$ must be to 1.0 to mark the start of the diffusive regime is a judgment call. In this work, the following heuristic algorithm was used to define the start of a curve segment for fitting within the diffusive regime:

1. Loop over j = 1, 2 to (N-20), where N is the total number of datapoints. (The 20 points at the end are reserved to ensure that there will always be a large enough curve segment remaining for us to perform an adequate fit to extract the D_s.) During each iteration of this loop, if avg_slope$_j$ >1.25, then set $k = j$.

2. D_s is computed as

$$D_s = \frac{\log[\mathrm{MSD}_{(k+15)}] - \log[\mathrm{MSD}_{(k+5)}]}{\log[t_{(k+15)}] - \log[t_{(k+5)}]} \tag{10}$$

When N is sufficiently large, step # 1 above sets k equal to the last point within the transition regime for which avg_slope$_j$ > 1.25 such that avg_slope$_{(k+1)} \leqslant 1.25$. However, if N is relatively small, then k may equal N-20, which is less than ideal but still ensures that enough datapoints are reserved to perform a fit to compute D_s. Step # 2 above then performs the fit starting with index (k + 5) and continuing to index (k + 15). The rational is that once avg_slope$_j$ drops below 1.25 [i.e., index value (k + 1)], then adding +4 to the index [i.e., (k + 5)] should have the effect of choosing a starting point with a slope near 1. For example, Figure 7 shows the first of three runs at the San Diego Supercomputing Center that was used to compute the self-diffusivity of N_2 in IRMOF-16 at 298 K, as listed in Table 2. Using this algorithm, the selected start time for the fit of eq 8 had avg_slope = 1.10, which is sufficiently close to 1 to give a meaningful D_s.



Result: Log MSD vs Log Time

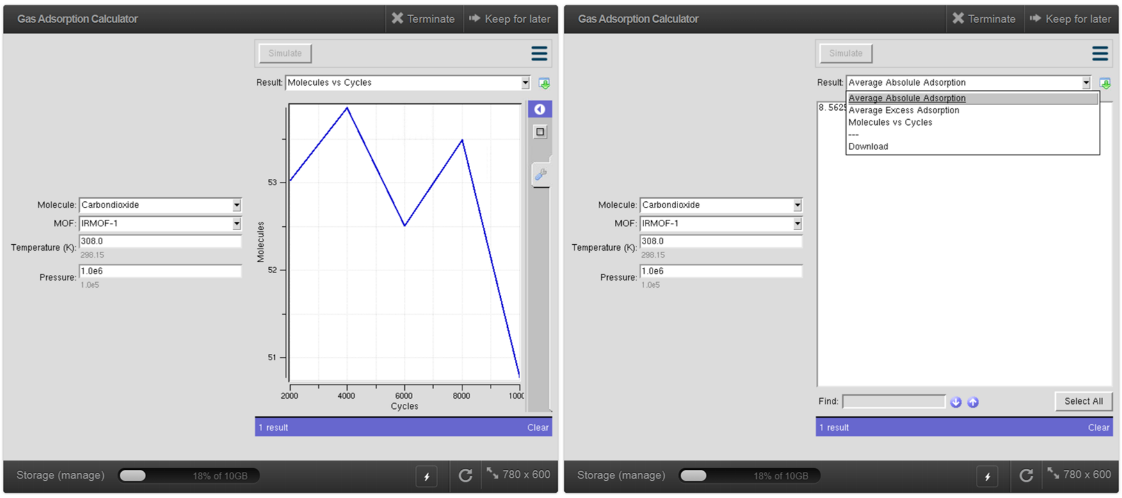
Gas: Nitrogen

MOF: IRMOF-16

Result: Diffusivity

16.01

Figure 8. Screenshots of Tool 1 for calculating the self-diffusion constant of a gas in a MOF.



Result: Molecules vs Cycles

Molecule: Carbondioxide

MOF: IRMOF-1

Temperature (K): 308.0 (298.15)

Pressure: 1.0e6 (1.0e5)

Result: Average Absolute Adsorption

8.5625

Figure 9. Screenshots of Tool 2 for calculating the absolute and excess adsorption of a gas in a MOF.

3. SIMULATION TOOL DESCRIPTIONS

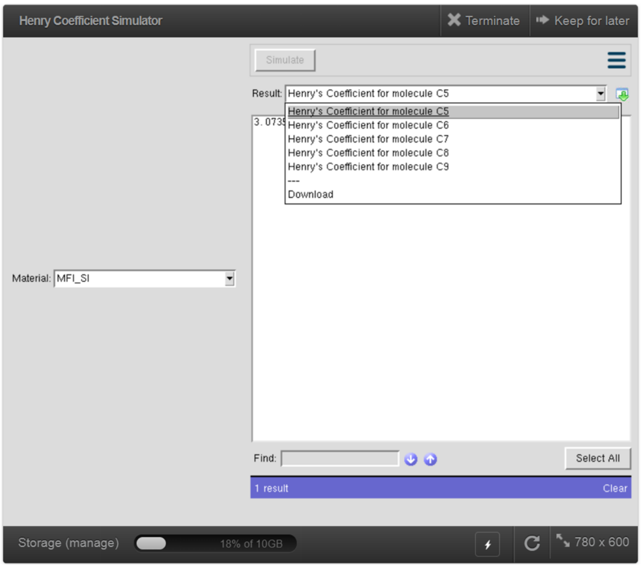
Note: The graphs of molecules versus number of cycles in tools # 2, 4, and 11 were prepared as follows. The production cycles were divided into five blocks of equal sizes, and the average number of molecules in each phase was computed by RASPA for each block. For each phase, this gives a total of five data points that correspond to the average number of molecules during 0–20% (aka block 1), 20–40% (aka block 2), 40–60% (aka block 3), 60–80% (aka block 4), and 80–100% (aka block 5) of the production cycles. For the mixed gas adsorption tool 4, separate molecules versus cycles plots are computed and displayed for methane and argon.

3.1. Gas Diffusion Coefficient in MOFs (Tool 1). This tool calculates the self-diffusion constant of gas molecules through a MOF using molecular dynamics in the *NVT* ensemble. Gases used for this tool include argon, hydrogen (H_2), nitrogen (N_2), methane, and carbon dioxide. MOFs used for this tool are IRMOF-1 and IRMOF-16. The simulation temperature is 298 K. The user selects the gas and the MOF from drop lists. Sixteen of the chosen molecules are included in the simulation box along with the chosen MOF. To minimize the effects of periodic boundary conditions on the computed self-diffusion constant, the simulation box contains 27 (i.e., 3 × 3 × 3) MOF unit cells. The tool outputs a graph of the MSD versus time (t) in both linear–linear and log–log scales. This tool uses the algorithm described in Section 2.4 to calculate the gas molecule's self-diffusion constant D_s. The value of D_s and the start and end times for the linear fit to compute D_s are displayed to the user. Figure 8 shows screenshots for this gas diffusion calculation tool.

3.2. Gas Adsorption Calculator (Tool 2). This tool calculates the average absolute and excess adsorption of gas molecules in a MOF using Monte Carlo in the grand canonical ensemble (μVT). Gases used for this tool include hydrogen (H_2), nitrogen (N_2), oxygen (O_2), methane, and carbon dioxide. MOFs used for this tool are IRMOF-1 and IRMOF-16. The user selects the gas and the MOF from drop lists. The user types in the temperature (in K) and pressure (in Pa) for the simulation. The tool outputs the average absolute adsorption, the average excess adsorption, and a graph of molecules vs cycles. This graph of molecules versus cycles helps demonstrate that the calculation has reached equilibrium. Figure 9 shows screenshots of this tool.

3.3. Henry's Coefficients Simulator (Tool 3). This tool uses Widom insertion to simulate Henry's coefficients of *n*-

alkane (*n*-pentane to *n*-nonane) for several nanoporous materials. To run a simulation, users are required to select one of the following nanoporous materials from a drop list: IRMOF-1, IRMOF-16, or the zeolite MFI_SI. The temperature used for the simulation is 573 K. Using a pre-calculated ideal gas Rosenbluth weight for each molecule, the tool outputs the Henry's coefficients for the above-mentioned *n*-alkanes, as depicted in Figure 10. The procedure this tool uses to compute the Henry's coefficient is analogous to an example described in the RASPA 2.0 manual.[15]


Henry Coefficient Simulator
✖ Terminate
➠ Keep for later
Simulate
Henry's Coefficient for molecule C5
Henry's Coefficient for molecule C6
Henry's Coefficient for molecule C7
Henry's Coefficient for molecule C8
Henry's Coefficient for molecule C9

Download
Find:
Select All
Clear
Storage (manage)
18% of 10GB
780 x 600

Material: MFI_SI

Result: Henry's Coefficient for molecule C5

3. 0735

1 result

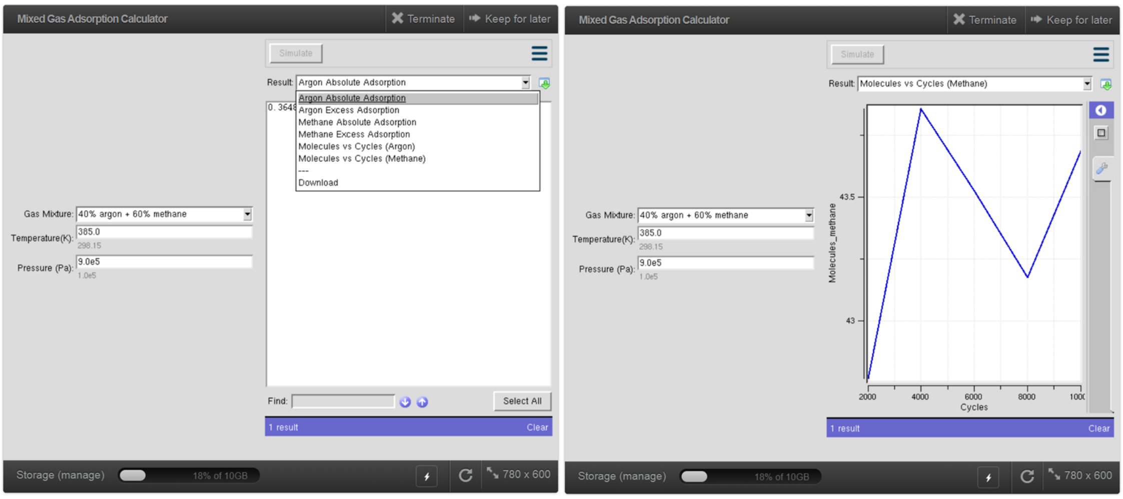
Figure 10. Screenshot of Tool 3 for calculating the Henry's coefficients of *n*-alkanes in porous materials.

3.4. Mixed Gas Adsorption Calculator (Tool 4). This tool simulates the adsorption of mixed gases (argon and methane) in IRMOF-1 using a Monte Carlo algorithm with the grand canonical ensemble (μVT). The GUI prompts the user to select the composition of the gas mixture from a drop list and then to enter the simulation temperature and pressure. Compositions available are (10 mol % argon + 90% methane) to (90 mol % argon + 10% methane) in 10% increments. This tool outputs the absolute and excess adsorption for the argon and methane components. The GUI also plots molecules versus the number of cycles separately for argon and methane to demonstrate how closely the calculation has approached equilibrium. Figure 11 shows the developed tool interface for this mixed adsorption calculation.

3.5. Mixed Gas Diffusion Calculator (Tool 5). This tool simulates the self-diffusion of gas mixtures through a MOF using molecular dynamics with the *NVT* ensemble. To minimize the effects of periodic boundary conditions on the computed self-diffusion constants, the simulation box contains 27 (i.e., 3 × 3 × 3) MOF unit cells. The gases used in this tool are argon and methane, and the MOF used is IRMOF-1. The GUI prompts the user to select the composition of the gas mixture from a drop list and then to enter the simulation temperature. Compositions available are (10 mol % argon + 90% methane) to (90 mol % argon + 10% methane) in 10% increments. The tool outputs a graph of the MSD versus time (t) in both linear–linear and log–log scales separately for argon and methane. This tool applies the algorithm described in Section 2.4 to separately compute D_s for methane and argon. The D_s values are displayed to the user along with the start and end times for the linear fits used to compute these D_s values. Figure 12 depicts the developed tool interface for mixed gas diffusion calculation.

3.6. Void Fraction Calculator (Tool 6). The void fraction calculator tool simulates the pore volume fraction of nanoporous materials. The nanoporous materials being calculated include both MOFs and zeolites: IRMOF-1, IRMOF-16, ITQ-1, ITQ-3, ITQ-7, ITQ-12, ITQ-29, KFI, LTA4A, and LTA5A. The tool requires the user to select (from a drop list) the material for which they want to calculate the void fraction. The void fraction is computed using Widom insertion of a helium atom, as described by Dubbeldam et al.[4]Figure 13 shows the developed tool interface for void fraction calculation.

3.7. Surface Area Calculator (Tool 7). This tool simulates the surface area of nanoporous materials. Users have the options to select the type of material and the probe distance (either σ or $r_{\min}$ as illustrated in Figure 1). The materials used in this simulation include both MOFs and zeolites: IRMOF-1, IRMOF-2, IRMOF-3, IRMOF-12,



Gas Mixture: 40% argon + 60% methane

Temperature(K): 385.0

Pressure (Pa): 9.0e5

Result: Argon Absolute Adsorption

0. 3648

1 result

Gas Mixture: 40% argon + 60% methane

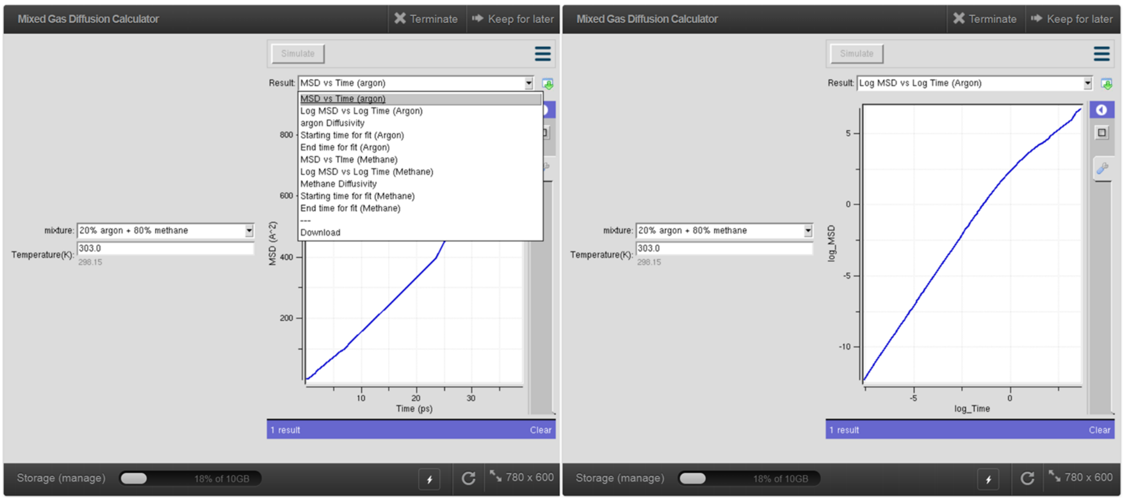
Temperature(K): 385.0

Pressure (Pa): 9.0e5

Result: Molecules vs Cycles (Methane)

1 result

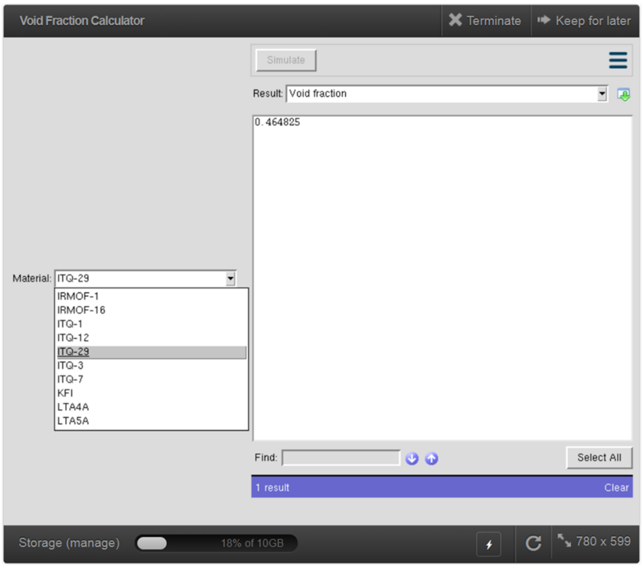
Figure 11. Screenshots of Tool 4 for calculating adsorption of a gas mixture in a porous material.



Result: MSD vs Time (argon)
mixture: 20% argon + 80% methane
Temperature(K): 303.0
298.15

Result: Log MSD vs Log Time (Argon)
mixture: 20% argon + 80% methane
Temperature(K): 303.0
298.15

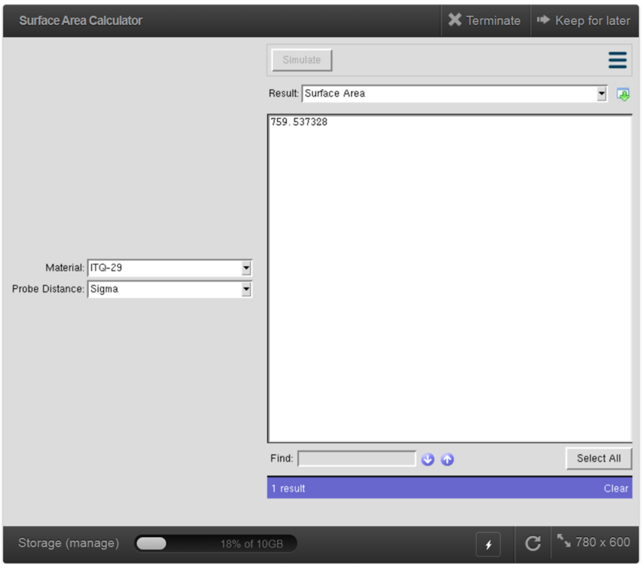
Figure 12. Screenshots of Tool 5 for calculating diffusion of a gas mixture in a porous material.



Result: Void fraction
Material: ITQ-29

```
0.464825
```

Figure 13. Screenshot of Tool 6 for calculating the void fraction of a porous material.



Result: Surface Area
Material: ITQ-29
Probe Distance: Sigma

```
759.537328
```

Figure 14. Screenshot of Tool 7 for calculating the surface area of a porous material.

IRMOF-16, ITQ-1, ITQ-3, ITQ-7, ITQ-12, ITQ-29, KFI, LTA4A, and LTA5A. This tool simulates surface area of the material using a single methane probe molecule at 298 K. Figure 14 shows the developed tool interface for surface area calculation.

3.8. Radial Distribution Function Calculator (Tool 8).

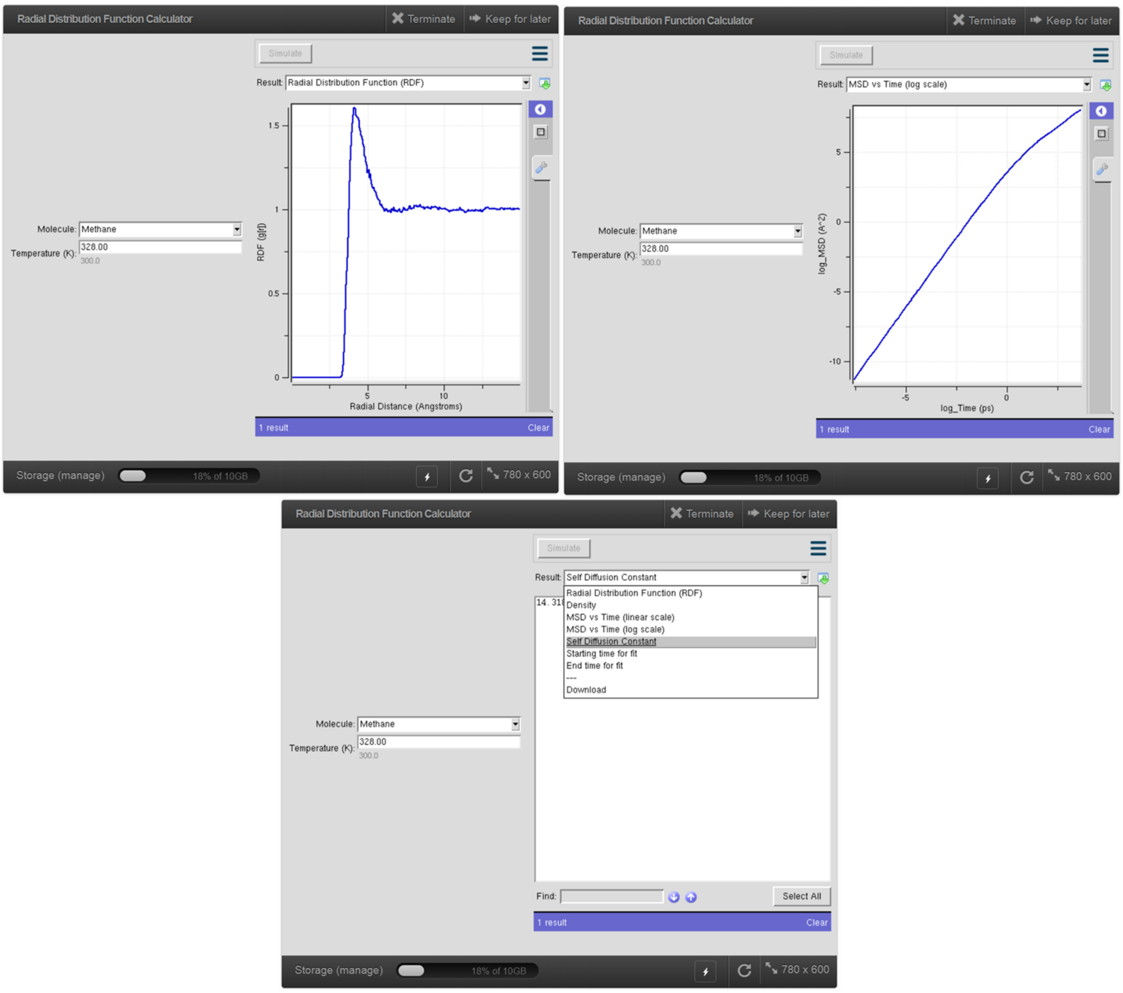
Figure 15 shows the developed tool interface for radial distribution function calculation. The GUI allows the user to select (from a drop list) which gas is to be used for the simulation: methane, ethane, propane, butane, nitrogen (N_2), or carbon dioxide. The GUI also allows the user to enter the simulation temperature. The simulation box is 30 Å × 30 Å × 30 Å and contains 100 molecules. Using molecular dynamics with the *NVT* ensemble, the tool performs the simulation and outputs a graph of the radial distribution function. It also outputs the density and self-diffusion graphs (i.e., MSD versus time for both linear−linear and log−log scales). This tool uses the algorithm described in Section 2.4 above to calculate the gas molecule's self-diffusion constant D_s. The value of D_s and the start and end times for the linear fit to compute D_s are displayed to the user.

3.9. Adsorption Energy Calculator (Tool 9).

This tool calculates the histogram of adsorption energies onto different sites within a MOF using Monte Carlo moves with one molecule per simulation box. No adsorbate−adsorbate interactions appear. This tool requires the user to select the kind of material to be used for the simulation, the gas molecule, and the temperature. Figure 16 shows the developed tool interface for adsorption energy calculation. The MOFs used for this tool include IRMOF-1 and IRMOF-16. The gas molecules are methane, ethane, hydrogen (H_2), nitrogen (N_2), and carbon dioxide. The tool outputs the energy histogram of the gas molecule as well as the average adsorption energy.

3.10. *NPT* Simulator (Tool 10).

This tool uses molecular dynamics with the isobaric−isothermal ensemble (*NPT*) to simulate the properties of gas molecules. To run a simulation, users are required to select the molecule from a drop list and enter the simulation temperature and pressure. The molecules


Radial Distribution Function Calculator

Molecule: Methane

Temperature (K): 328.00 (300.0)

Result: Radial Distribution Function (RDF)

Result: MSD vs Time (log scale)

Result: Self Diffusion Constant

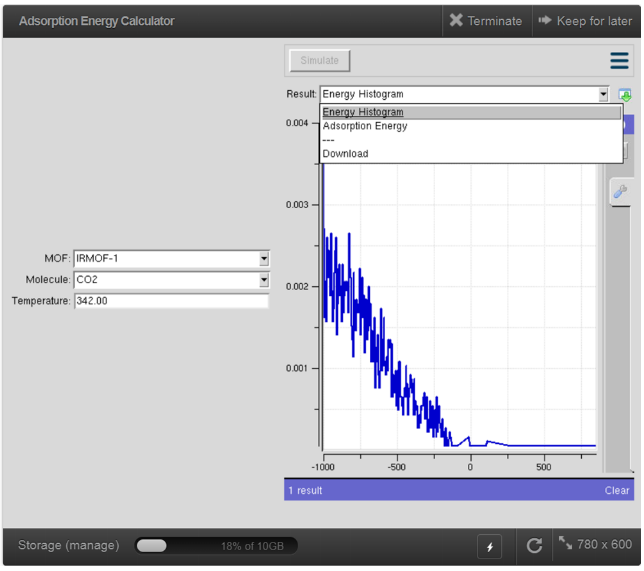
14. 31

1 result

Storage (manage) 18% of 10GB

780 x 600

Figure 15. Screenshots of Tool 8 for calculating the radial distribution function of a gas.


Adsorption Energy Calculator

MOF: IRMOF-1

Molecule: CO2

Temperature: 342.00

Result: Energy Histogram

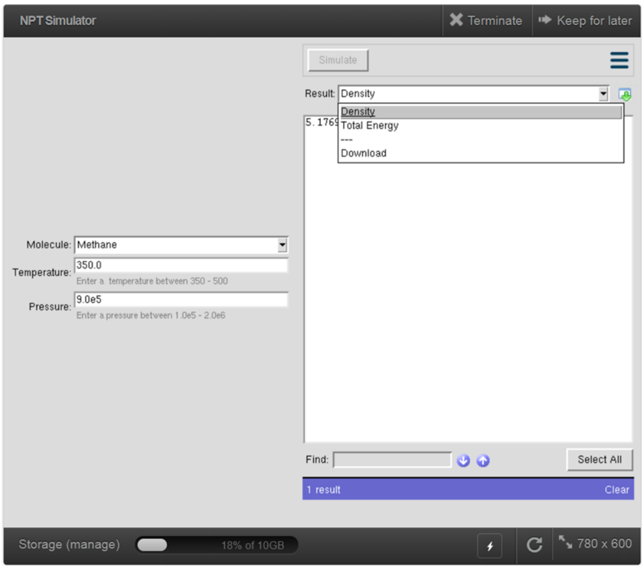
1 result

Storage (manage) 18% of 10GB

780 x 600

Figure 16. Screenshot of Tool 9 for calculating the adsorption energy histogram.

used in the tool include methane, ethane, propane, and carbon dioxide. Fifty molecules of the user-selected compound are simulated inside the simulation box. Figure 17 shows the interface for this *NPT* simulation tool. The tool outputs the density and total energy of the gas at the given conditions.


NPT Simulator

Molecule: Methane

Temperature: 350.0

Enter a temperature between 350 - 500

Pressure: 9.0e5

Enter a pressure between 1.0e5 - 2.0e6

Result: Density

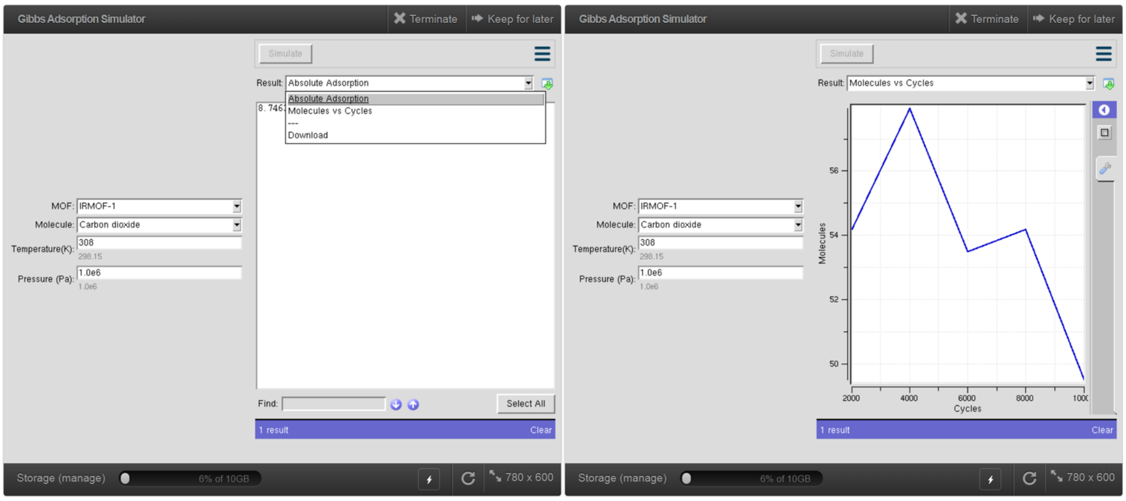
5. 1765

1 result

Storage (manage) 18% of 10GB

780 x 600

Figure 17. Screenshot of Tool 10 for calculating the density and total energy of a gas in the *NPT* ensemble.



Gibbs Adsorption Simulator

MOF: IRMOF-1

Molecule: Carbon dioxide

Temperature(K): 308

Pressure (Pa): 1.0e6

Result: Absolute Adsorption

8. 746

Gibbs Adsorption Simulator

MOF: IRMOF-1

Molecule: Carbon dioxide

Temperature(K): 308

Pressure (Pa): 1.0e6

Result: Molecules vs Cycles

Figure 18. Screenshots of Tool 11 for calculating gas adsorption in a MOF using the Gibbs ensemble.

Table 3. Absolute and Excess Adsorption of CO_2 in IRMOF-1 at 308 K and 10 Bar Was Computed Using Tool 2 in the Grand Canonical Ensemble[a]

			tool 2 (grand canonical ensemble)		tool 11 (Gibbs ensemble)
	initialization cycles	production cycles	absolute adsorption (mol/kg framework)	excess adsorption (mol/kg framework)	absolute adsorption (mol/kg framework)
nanoHUB	5000	10,000	8.7 ± 0.2	8.0 ± 0.2	8.5 ± 0.2
SDSC	25,000	50,000	8.56 ± 0.05	7.91 ± 0.05	8.85 ± 0.01

[a] For comparison, the last column lists the absolute adsorption from Tool 11 in the Gibbs ensemble.

Table 4. Henrys' Coefficients for *n*-Pentane to *n*-Nonane in MFI_SI Zeolite at 573 K Was Computed Using Tool 3[a]

	production cycles	C5	C6	C7	C8	C9
nanoHUB	5000	30.2 ± 0.7	60 ± 1	123 ± 1	243 ± 10	446 ± 13
SDSC	25,000	28.8 ± 0.2	57.83 ± 0.04	116 ± 1	223 ± 6	410 ± 9

[a] Units are 10^{-7} mol/(kg Pa).

Table 5. Absolute and Excess Adsorption of 40% Argon and 60% Methane Mixture at 385 K and 9 Bar Was Computed Using Tool 4

	initialization cycles	production cycles	argon absolute adsorption (mol/kg framework)	argon excess adsorption (mol/kg framework)	methane absolute adsorption (mol/kg framework)	methane excess adsorption (mol/kg framework)
nanoHUB	5000	10,000	0.366 ± 0.002	0.203 ± 0.002	0.8849 ± 0.0013	0.6414 ± 0.0013
SDSC	25,000	50,000	0.3654 ± 0.0005	0.203 ± 0.001	0.883 ± 0.002	0.640 ± 0.002

Table 6. Self-Diffusion Constant of 20% Argon and 80% Methane Mixture in IRMOF-1 at 303 K Was Computed Using Tool 5

	initialization cycles	equilibration cycles	production cycles	argon self-diffusion ($Å^2$/picosecond)	methane self-diffusion ($Å^2$/picosecond)
nanoHUB	50,000	50,000	100,000	2.4 ± 0.9	3.7 ± 0.3
SDSC	50,000	50,000	250,000	3.4 ± 0.5	3.2 ± 0.5

3.11. Gibbs Adsorption Simulator (Tool 11). This tool simulates the adsorption of gas molecules onto a MOF. The difference between adsorption simulation using the Gibbs ensemble versus using the grand canonical ensemble (μVT) is that Gibbs ensemble adsorption uses two boxes for the simulation. The framework is contained in one box, and a gas phase is in the other box. The simulation contains a total of 100 molecules, and these can move between the two boxes. The gas-phase box can change volume. This simulation computes absolute adsorption using the forcefield without requiring a fugacity coefficient or an equation of state (Adsorption calculations using the grand canonical ensemble require a fugacity coefficient that is not computed self-consistently from the forcefield). The MOFs used for this tool are IRMOF-1 and IRMOF-16. The gas molecules include methane, argon, hydrogen (H_2), nitrogen (N_2), and carbon dioxide. This tool is similar to the grand canonical adsorption tool in terms of input parameters; that is, the user specifies a temperature and a pressure. The tool outputs the average absolute adsorption of the gas molecule, as shown in Figure 18. This tool also outputs a graph of molecules versus cycles in one box to show how closely the calculation has reached equilibrium.

4. RESULTS AND DISCUSSION

4.1. Convergence Tests. Each simulation tool was systematically tested for convergence. First, the menu options to be tested were randomly selected. For tools that had multiple choices of material, one material was selected at random. For tools that had choices of temperature, pressure, and/or composition as input, a random selection was made. Each of these nanoHUB tools has fixed numbers of initialization, equilibration, and/or production cycles that are used every time the tool runs. The numbers of these cycles are listed in Tables 2345678910.

Table 7. Void Fraction (Tool 6 for 300 K) and Surface Area (Tool 7 Using Sigma Probe Radius) of ITQ-29 Zeolite

	production cycles	void fraction	surface area (m^2/g)
nanoHUB	2000	0.4623 ± 0.0013	758.9 ± 0.2
SDSC	10,000	0.4634 ± 0.0003	759.26 ± 0.06

Table 8. Self-Diffusion Constant of Methane at 328 K and a Density of 98.66 kg/m^3 Was Computed Using Tool 8

	initialization cycles	equilibration cycles	production cycle	self-diffusion ($Å^2/ps$)
nanoHUB	50,000	50,000	100,000	14.4 ± 0.2
SDSC	50,000	50,000	250,000	14.8 ± 0.7

Table 9. Adsorption Energy of CO_2 in IRMOF-1 at 342 K Was Computed Using Tool 9

	initialization cycles	production cycles	adsorption energy (K)
nanoHUB	5000	10,000	−1173 ± 2
SDSC	25,000	50,000	−1168 ± 2

Three simulations were performed for each tool, and the mean and standard deviation for the main numeric results were computed. All three simulations used the same simulation conditions, except a different seed value was used to initialize the random number generator. The same conditions were run on clusters at the San Diego Supercomputing Center (SDSC) using a larger number of cycles and different random seed values to test for numerical convergence. As shown in Tables 2345678910, results with the larger number of cycles were similar to those computed using our nanoHUB tools. This demonstrates a reasonable level of convergence. Table 11 lists the times required to run each nanoHUB tool.

4.2. Comparing Simulations to Experiments. Here, we compared the experimental results from various literature sources to the simulated results obtained from two nanoHUB tools. These tools are gas adsorption calculator (tool 2) and surface area calculator (tool 7).

Table 12 compares experimentally extracted and calculated surface areas for IRMOF-2 and IRMOF-3. As shown in Table 12, surface areas computed using the "sigma" and "r_{min}" probe distances were larger than experimentally extracted Brunauer−Emmett−Teller (BET) surface areas and smaller than surface areas computed using the Connolly method with the probe distance set equal to the N_2 kinetic diameter.

Table 11. Time Required to Run Each Nanohub Tool[a]

tool	run time (min)	tool	run time (min)
1	160 ± 30	7	120.7 ± 0.2
2	24.6 ± 0.7	8	8 ± 2
3	32 ± 1	9	6.5 ± 0.4
4	33.0 ± 0.2	10	4.77 ± 0.01
5	153 ± 2	11	28.0 ± 0.5
6	0.676 ± 0.008		

[a] The average and standard deviation of three runs is listed.

Finally, Figure 19 compares calculated and experimental adsorption isotherms of nitrogen (N_2), methane, and carbon dioxide gases adsorbed in IRMOF-1 at 298 K. These simulated adsorption isotherms predicted slightly more adsorption than the experimental isotherms. This could be due to imperfections in the forcefield or due to impurities (e.g., residual molecules) or defects in the MOF crystals that partially block adsorption sites.

5. CONCLUSIONS

Eleven simulation modules were successfully developed and published on nanoHUB. These tools were created to teach users how to use classical atomistic simulations to compute the following: (i) gas adsorption and diffusion in porous materials, (ii) radial distribution functions and self-diffusivity of pure fluids, and (iii) surface areas and void fractions of porous materials. These modules included a GUI on the front end to accept user inputs and to display results back to the user. Middleware communicated information between the GUI front end and the RASPA software that performed the simulations. These tools are accessible worldwide through an online web browser without requiring the user to install any software. These simulations run on nanoHUB's computing cluster without requiring the user to supply computational resources. Each simulation tool has an associated set of PowerPoint slides that briefly explain how the tool works. An example RASPA input file is also provided for each tool. This arrangement allows users to quickly learn basic simulation principles.

In this article, we also reported tests demonstrating these tools give useful results. We report runtimes, which were purposefully kept short to facilitate easier learning. We show the computational precision is reasonable and compares favorably to longer runs that included more production cycles. For a couple of the tools, computed results were compared to published experimental data. As shown in Table 12, surface areas computed using the "sigma" and "r_{min}" probe distances were larger than experimentally extracted BET surface areas and smaller than surface areas computed using the Connolly method with the probe distance set equal to the N_2 kinetic diameter. For the adsorption of N_2, methane, and CO_2 in IRMOF-1 at 298 K shown in Figure 19, we compared

Table 10. Properties of Methane at 350 K and 9 Bar Was Computed in the *NPT* Ensemble Using Tool 10

	initialization cycles	equilibration cycles	production cycle	density (kg/m^3)	system energy (K)
nanoHUB	5000	5000	10,000	5.11 ± 0.14	−428 ± 20
SDSC	25,000	25,000	50,000	5.01 ± 0.07	−447 ± 4

Table 12. Comparison between Experimentally Extracted and Calculated Surface Areas for IRMOF-2 and IRMOF-3[a]

material	area (m^2/g) Langmuir[b]	area (m^2/g) BET[b]	area (m^2/g) (N_2 kinetic diam)[b]	area (m^2/g) (sigma)	area (m^2/g) (r_{min})
IRMOF-2	2544	1722	2780	2684.4 ± 1.0	2349.4 ± 0.5
IRMOF-3	3062	2446	3613	3296.4 ± 0.9	2834.0 ± 1.3

[a] Sigma and r_{min} are surface areas calculated using two different probe distances. The calculations used 2000 production cycles. [b] From ref 44.

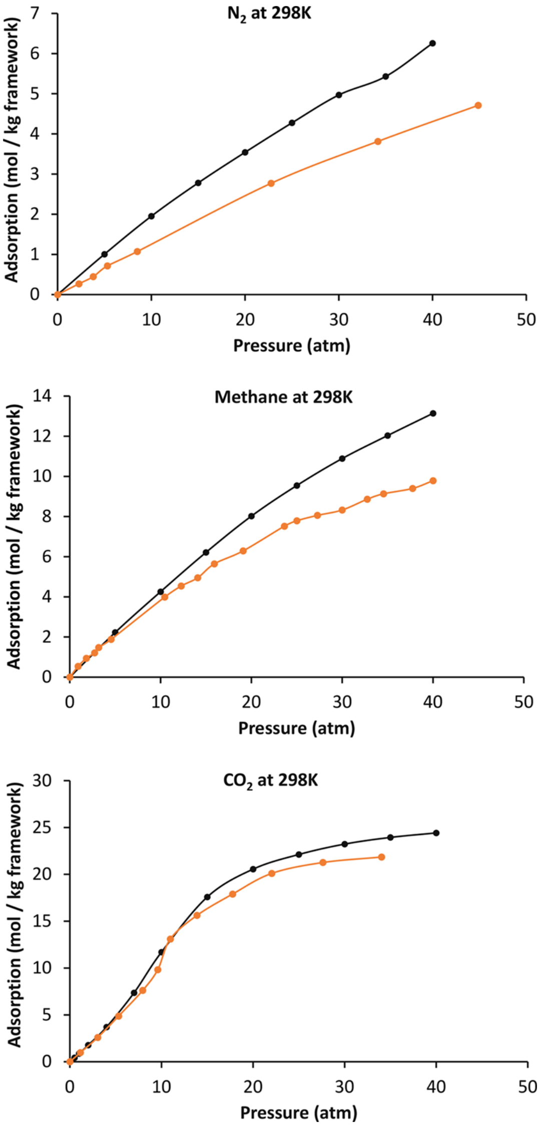


Figure 19. Comparison of calculated to experimental isotherms of N_2, methane, and CO_2 gases adsorbed in the MOF IRMOF-1. The black curves are the calculated isotherms. The orange curves are the N_2 (ref 45), methane (ref 46), and CO_2 (ref 47) experimental isotherms.

results from the gas adsorption simulation tool to previously reported experimental results. In these cases, the simulated adsorption isotherms predicted slightly more adsorption than the experimental isotherms; this could be due to imperfections in the forcefield or due to impurities (e.g., residual molecules) or defects in the MOF crystals that partially block adsorption sites.

AUTHOR INFORMATION

Corresponding Author

Thomas A. Manz − *Chemical and Materials Engineering, New Mexico State University, Las Cruces, New Mexico 88003-3805, United States*; orcid.org/0000-0002-4033-9864; Email: tmanz@nmsu.edu

Author

Julian C. Umeh − *Chemical and Materials Engineering, New Mexico State University, Las Cruces, New Mexico 88003-3805, United States*; orcid.org/0000-0002-7132-8033

Complete contact information is available at: https://pubs.acs.org/10.1021/acsomega.2c06978

Author Contributions

J.C.U. programmed the 11 nanoHUB simulation tools. T.A.M. supervised the project. Both authors designed the study, performed simulations, analyzed data, and wrote the manuscript.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

National Science Foundation (NSF) CAREER Award DMR-1555376 provided financial support. Supercomputing resources were provided by the Extreme Science and Engineering Discovery Environment (XSEDE).[48] XSEDE is funded by NSF Grant ACI-1548562. XSEDE project Grant TG-CTS100027 provided allocations on the Comet and Expanse clusters at the San Diego Supercomputing Center (SDSC). Special thanks to the nanoHUB support team for answering our many questions about setting up tools and for their technical assistance.

REFERENCES

(1) Madhavan, K.; Zentner, L.; Farnsworth, V.; Shivarajapura, S.; Zentner, M.; Denny, N.; Klimeck, G. nanoHUB.org: cloud-based services for nanoscale modeling, simulation, and education. *Nanotechnol. Rev.* **2013**, *2*, 107−117.

(2) Magana, A. J.; Brophy, S. P.; Bodner, G. M. An exploratory study of engineering and science students' perceptions of nanoHUB.org simulations. *Int. J. Eng. Educ.* **2012**, *28*, 1019−1032.

(3) Nanohub.org usage statistics. https://nanohub.org/usage, (accessed March 1, 2022).

(4) Dubbeldam, D.; Calero, S.; Ellis, D. E.; Snurr, R. Q. RASPA: molecular simulation software for adsorption and diffusion in flexible nanoporous materials. *Mol. Simul.* **2016**, *42*, 81−101.

(5) Rogge, S. M. J.; Goeminne, R.; Demuynck, R.; Gutiérrez-Sevillano, J. J.; Vandenbrande, S.; Vanduyfhuys, L.; Waroquier, M.; Verstraelen, T.; Van Speybroeck, V. Modeling gas adsorption in flexible metal-organic frameworks via hybrid Monte Carlo/molecular dynamics schemes. *Adv. Theory Simul.* **2019**, *2*, No. 1800177.

(6) Lin, L. C.; Lee, K.; Gagliardi, L.; Neaton, J. B.; Smit, B. Force-field development from electronic structure calculations with periodic boundary conditions: applications to gaseous adsorption and transport in metal-organic frameworks. *J. Chem. Theory Comput.* **2014**, *10*, 1477−1488.

(7) Dubbeldam, D.; Walton, K. S.; Vlugt, T. J. H.; Calero, S. Design, parameterization, and implementation of atomic force fields for adsorption in nanoporous materials. *Adv. Theory Simul.* **2019**, *2*, No. 1900135.

(8) Kaplan, I. G.*Intermolecular Interactions: Physical Picture, Computational Methods and Model Potentials*; John Wiley & Sons: West Sussex, U.K., 2006.

(9) Rappe, A. K.; Casewit, C. J.; Colwell, K. S.; Goddard, W. A.; Skiff, W. M. UFF, a full periodic-table force-field for molecular mechanics and molecular-dynamics simulations. *J. Am. Chem. Soc.* **1992**, *114*, 10024−10035.

(10) Walters, E. T.; Mohebifar, M.; Johnson, E. R.; Rowley, C. N. Evaluating the London dispersion coefficients of protein force fields using the exchange-hole dipole moment model. *J. Phys. Chem. B* **2018**, *122*, 6690−6701.

(11) Jones, J. E. On the determination of molecular fields - II From the equation of state of a gas. *Proc. R. Soc. London, Ser. A* **1924**, *106*, 463−477.

(12) Boda, D.; Henderson, D. The effects of deviations from Lorentz−Berthelot rules on the properties of a simple mixture. *Mol. Phys.* **2008**, *106*, 2367−2370.

(13) Berthelot, D. Sur le mélange des gaz. *Compt. Rendus* **1898**, *126*, 1703−1706.

(14) Lorentz, H. A. Ueber die anwendung des satzes vom virial in der kinetischen theorie der gase. *Ann. Phys.* **1881**, *248*, 127−136.

(15) Dubbeldam, D.; Calero, S.; Vlugt, T. J. H.; Ellis, D. E.; Snurr, R. Q.*RASPA 2.0: Molecular Software Package for Adsorption and Diffusion in (Flexible) Nanoporous Materials*, program manual, March 2015, pp 1−145.

(16) Dubbeldam, D.; Walton, K. S.; Ellis, D. E.; Snurr, R. Q. Exceptional negative thermal expansion in isoreticular metal-organic frameworks. *Angew. Chem., Int. Ed.* **2007**, *46*, 4496−4499.

(17) Chen, B.; Siepmann, J. I. Transferable potentials for phase equilibria. 3. Explicit-hydrogen description of normal alkanes. *J. Phys. Chem. B* **1999**, *103*, 5370−5379.

(18) Eggimann, B. L.; Sunnarborg, A. J.; Stern, H. D.; Bliss, A. P.; Siepmann, J. I. An online parameter and property database for the TraPPE force field. *Mol. Simul.* **2014**, *40*, 101−105.

(19) Martin, M. G.; Siepmann, J. I. Transferable potentials for phase equilibria. 1. United-atom description of n-alkanes. *J. Phys. Chem. B* **1998**, *102*, 2569−2577.

(20) Potoff, J. J.; Siepmann, J. I. Vapor−liquid equilibria of mixtures containing alkanes, carbon dioxide, and nitrogen. *AIChE J.* **2001**, *47*, 1676−1682.

(21) Darkrim, F.; Levesque, D. Monte Carlo simulations of hydrogen adsorption in single-walled carbon nanotubes. *J. Chem. Phys.* **1998**, *109*, 4981−4984.

(22) Zhang, L.; Siepmann, J. I. Direct calculation of Henry's law constants from Gibbs ensemble Monte Carlo simulations: nitrogen, oxygen, carbon dioxide and methane in ethanol. *Theor. Chem. Acc.* **2006**, *115*, 391−397.

(23) Mayo, S. L.; Olafson, B. D.; Goddard, W. A. DREIDING - A generic force field for molecular simulations. *J. Phys. Chem.* **1990**, *94*, 8897−8909.

(24) Bai, P.; Tsapatsis, M.; Siepmann, J. I. TraPPE-zeo: Transferable potentials for phase equilibria force field for all-silica zeolites. *J. Phys. Chem. C* **2013**, *117*, 24375−24387.

(25) Dubbeldam, D.; Torres-Knoop, A.; Walton, K. S. On the inner workings of Monte Carlo codes. *Mol. Simul.* **2013**, *39*, 1253−1292.

(26) Heinen, J.; Burtch, N. C.; Walton, K. S.; Dubbeldam, D. Flexible force field parameterization through fitting on the ab initio-derived elastic tensor. *J. Chem. Theory Comput.* **2017**, *13*, 3722−3730.

(27) Heinen, J.; Dubbeldam, D. On flexible force fields for metal-organic frameworks: Recent developments and future prospects. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **2018**, *8*, No. e1363.

(28) Allen, M. P.; Tildesley, D. J.*Computer Simulation of Liquids*, 2nd ed.; Oxford University Press: Oxford, U.K., 2017.

(29) Frenkel, D.; Smit, B.*Understanding Molecular Simulation: From Algorithms to Applications*, 2nd ed.; Academic Press: San Diego, California, 2002.

(30) Hastings, W. K. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **1970**, *57*, 97−109.

(31) Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; Teller, E. Equation of state calculations by fast computing machines. *J. Chem. Phys.* **1953**, *21*, 1087−1092.

(32) Hirotani, A.; Mizukami, K.; Miura, R.; Takaba, H.; Miya, T.; Fahmi, A.; Stirling, A.; Kubo, M.; Miyamoto, A. Grand canonical Monte Carlo simulation of the adsorption of CO_2 on silicalite and NaZSM-5. *Appl. Surf. Sci.* **1997**, *120*, 81−84.

(33) Gao, G.; Wang, W. Gibbs Ensemble Monte Carlo simulation of binary vapor-liquid equilibria for CFC alternatives. *Fluid Phase Equilib.* **1997**, *130*, 157−166.

(34) Slepoy, A.; Thompson, A. P.; Plimpton, S. J. A constant-time kinetic Monte Carlo algorithm for simulation of large biochemical reaction networks. *J. Chem. Phys.* **2008**, *20*, No. 205101.

(35) Dove, M. T. An introduction to atomistic simulation methods. *Seminarios de la SEM* **2008**, *4*, 7−37.

(36) Braun, E.; Gilmer, J.; Mayes, H. B.; Mobley, D. L.; Monroe, J. I.; Prasad, S.; Zuckerman, D. M. Best practices for foundations in molecular simulations. *Living J. Comput. Mol. Sci.* **2019**, *1*, No. 5957.

(37) Abouelnasr, M. K. F.; Smit, B. Diffusion in confinement: kinetic simulations of self- and collective diffusion behavior of adsorbed gases. *Phys. Chem. Chem. Phys.* **2012**, *14*, 11600−11609.

(38) Skoulidas, A. I.; Sholl, D. S. Self-diffusion and transport diffusion of light gases in metal-organic framework materials assessed using molecular dynamics simulations. *J. Phys. Chem. B* **2005**, *109*, 15760−15768.

(39) Werder, T.; Walther, J. H.; Jaffe, R. L.; Halicioglu, T.; Koumoutsakos, P. On the water-carbon interaction for use in molecular dynamics simulations of graphite and carbon nanotubes. *J. Phys. Chem. B* **2003**, *107*, 1345−1352.

(40) Henry, A. S.; Chen, G. Spectral phonon transport properties of silicon based on molecular dynamics simulations and lattice dynamics. *J. Comput. Theor. Nanosci.* **2008**, *5*, 141−152.

(41) Karplus, M.; McCammon, J. A. Molecular dynamics simulations of biomolecules. *Nat. Struct. Biol.* **2002**, *9*, 646−652.

(42) Umeh, J. C.Development of Simulation Tools on Nanuhub as Learning Modules. Master's thesis; New Mexico State University: Las Cruces, NM, 2019.

(43) Humbert, M. T.; Zhang, Y.; Maginn, E. J. PyLAT: Python LAMMPS analysis tools. *J. Chem. Inf. Model.* **2019**, *59*, 1301−1305.

(44) Rowsell, J. L. C.; Yaghi, O. M. Effects of functionalization, catenation, and variation of the metal oxide and organic linking units on the low-pressure hydrogen adsorption properties of metal-organic frameworks. *J. Am. Chem. Soc.* **2006**, *128*, 1304−1315.

(45) Siberio-Perez, D. Y.; Wong-Foy, A. G.; Yaghi, O. M.; Matzger, A. J. Raman spectroscopic investigation of CH_4 and N_2 adsorption in metal-organic frameworks. *Chem. Mater.* **2007**, *19*, 3681−3685.

(46) Duren, T.; Sarkisov, L.; Yaghi, O. M.; Snurr, R. Q. Design of new materials for methane storage. *Langmuir* **2004**, *20*, 2683−2689.

(47) Walton, K. S.; Millward, A. R.; Dubbeldam, D.; Frost, H.; Low, J. J.; Yaghi, O. M.; Snurr, R. Q. Understanding inflection and steps in carbon dioxide adsorption isotherms in metal-organic frameworks. *J. Am. Chem. Soc.* **2007**, *130*, 406−407.

(48) Towns, J.; Cockerill, T.; Dahan, M.; Foster, I.; Gaither, K.; Grimshaw, A.; Hazlewood, V.; Lathrop, S.; Lifka, D.; Peterson, G. D.; Roskies, R.; Scott, J. R.; Wilkins-Diehr, N. XSEDE: accelerating scientific discovery. *Comput. Sci. Eng.* **2014**, *16*, 62−74.