Computer Physics Communications 291 (2023) 108827

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

Computer Programs in Physics

Xsorb: A software for identifying the most stable adsorption configuration and energy of a molecule on a crystal surface ☆

Enrico Pedretti, Paolo Restuccia, M. Clelia Righi *

Dipartimento di Fisica e Astronomia, Università di Bologna, Viale Berti Pichat 6/2, Bologna, 40127, Italy

ARTICLE INFO

Article history:
Received 30 April 2023
Received in revised form 7 June 2023
Accepted 12 June 2023
Available online 19 June 2023

Keywords:
Materials science
Molecular adsorption
Density functional theory

ABSTRACT

Molecular adsorption is the first important step of many surface-mediated chemical processes, from catalysis to lubrication. This phenomenon is controlled by physical/chemical interactions, which can be accurately described by first-principles calculations. Several computational tools have been developed to study molecular adsorption based on high throughput/automatized approaches in recent years. However, these tools can sometimes be over-sophisticated for non-expert users. Here we present Xsorb, a Python-based program for identifying the accurate adsorption energy and geometry of complex molecules on crystalline (reconstructed) surfaces. The program automatically samples the potential energy surface (PES) that describes the molecule-surface interaction by generating several adsorption configurations through symmetry operations. The set of the most representative ones is automatically identified through a fast pre-optimization scheme. Finally, the PES global minimum is identified through a full structural optimization process. We show the program capabilities through an example consisting of a hydrocarbon molecule, 1-hexene, adsorbed over the (110) surface of iron and the reconstructed (001) surface of diamond. This program, despite its conceptual simplicity, is very effective in reducing the computational workload usually associated with the creation and optimization of several adsorption configurations.

Program summary
Program title: Xsorb
CPC Library link to program files: https://doi.org/10.17632/kv97tgybx8.1
Developer's repository link: https://gitlab.com/triboteam/xsorbed/
Licensing provisions: CC by 4.0
Programming language: Python (version 3.7 and above) and Quantum ESPRESSO (for the *ab initio* calculations).
Nature of problem: Identifying the most stable adsorption configuration of a molecule over a substrate and compute its adsorption energy.
Solution method: Creating a Python-based code that generates many adsorption configurations with different molecular orientations, performs a preliminary partial geometrical optimization with density functional theory calculations of all these configurations, identifies the most relevant ones for the full geometrical optimization and computes the adsorption energy.

1. Introduction

Molecular adsorption, i.e., the binding of a molecule over a surface, is a necessary prerequisite to any surface-mediated chemical process, from catalysis [1,2] to molecular electronics [3], biomedicine [4], electrochemistry [5], lubrication [6,7] and corrosion [8,9]. For example, friction is ubiquitous whenever moving components are in contact. It causes significant energy losses and undermines the functionality of devices, ultimately leading to their failure. A strategy to reduce friction and wear in engines is based on the use of lubricant additives, i.e., molecules added in base oils that first adsorb over the sliding substrate and then react with it by forming protective, lubricious films [10]. Similarly, corrosion damages many materials comprising carbon steel, which has a low intrinsic resistance to corrosive processes [11]. It is, therefore, cru-

☆ The review of this paper was arranged by Prof. Blum Volker.
☆☆ This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).
* Corresponding author.
E-mail address: clelia.righi@unibo.it (M.C. Righi).

https://doi.org/10.1016/j.cpc.2023.108827

cial to use molecular corrosion inhibitors that create a protective coating, making the surface inert, more resistant to degradation, and preventing dire consequences on major infrastructures.

Regardless of the research field of interest, molecular adsorption is controlled by the atomistic interactions between the substrate and the adsorbate. The characterization of such interactions is only partially possible by experiments, such as adsorption isotherms [12] and scanning tunneling microscopy. First-principles calculations play a relevant role in this context as they allow for an accurate description of bond-forming and breaking processes, which is essential to design novel molecular compounds for specific applications.

In recent years, several computational studies on molecular adsorption have been performed with the help of computational tools that allow automatizing the execution of *ab initio* calculations [13–20]. Most of them are based on a high-throughput approach that allows to calculate the adsorption of simple molecules and molecular fragments in an automatized way and store/retrieve all the generated data in publicly available database [14,16,21–23]. An example of such a framework is GasPy [14,21], based on Fireworks [24] and Atomate [25] as workflow manager. This approach is highly efficient in computing simultaneously thousands of potential molecules and substrates combinations, resulting very useful for identifying novel potential catalytic materials, a field in which these tools have been widely used [14,16,22,23]. On the other hand, the platform employed to generate such data is rather complex due to the advanced coding necessary to implement and apply the workflow managers. Simpler and lighter programs, such as ASAP [26] and DockOnSurf [27] have been also developed. Usually, these programs are based on Python, allowing an easier installation and more user-friendly handling of the operations [26–28]. The absence of a proper workflow manager can make it more challenging to deal with many configurations and systems. However, specific libraries like the Atomic Simulation Environment (ASE) [29] can help create, generate, and handle many systems and configurations simultaneously.

One of the most critical issues in evaluating molecular adsorption is the sampling and screening of all the possible adsorption configurations that correspond to different orientations of the molecule over different adsorption sites on the substrate. Such configurational space can become huge and almost impossible to explore in its entirety by increasing the complexity of the molecule and the substrate. Therefore, several optimization algorithms have been developed during the last years to address this analysis, like metadynamics [30], Bayesian optimization [31,32], minima hopping [33] and Monte Carlo [34]. However, linking these programs with complex DFT codes is not trivial [35,36]. A possible novel technique to predict the most stable molecular adsorption configurations combines machine learning algorithms with DFT calculations, but this work is still in its infancy [37].

Here we introduce a simple but equally efficient approach to automatically identify the adsorption configurations that have the highest statistical weight, by means of the following steps: i) automatic generation of many adsorption configurations through the molecule rotation and translation with respect to the surface; ii) identification of the most relevant ones by screening their energies through a partial optimization process; iii) full structural optimization of the configurations identified as most relevant. This approach, which is conceptually very simple and with a low level of algorithm sophistication, is anyway very effective as it allows one to easily identify the most relevant adsorption configurations of complex molecules on (reconstructed) surfaces in a relatively short time. Many possible configurations are automatically generated, thus avoiding relying exclusively on user experience and intuition, which can limit the number of configurations analyzed. The program, `Xsorb`, here released is written in Python and is based on well-established Python libraries, such as ASE [29] and Pymatgen [38], for the generation of molecular adsorbed systems and on Quantum ESPRESSO (QE) [39–41] for DFT calculations.

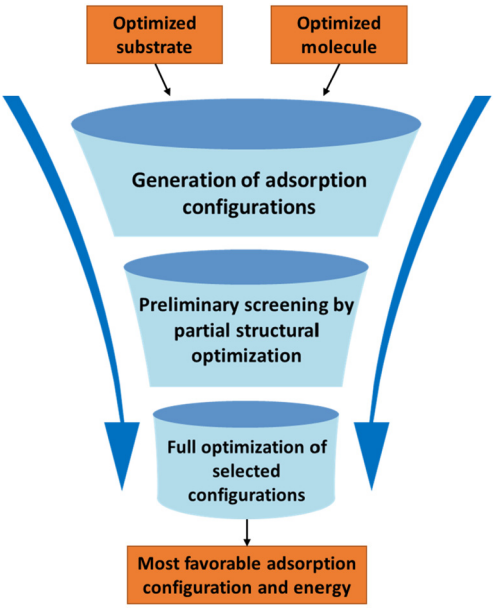


Fig. 1. Schematic flowchart of `Xsorb`.

The paper is organized as follows: Section 2 describes the workflow executed by `Xsorb` during a typical run. Section 3 explains how to install and execute `Xsorb`. Section 4 shows two examples of `Xsorb` usage for computing the adsorption of the same hydrocarbon molecule, 1-hexene (hexene from now on), over the (110) surface of a bcc crystal, iron in the specific case, and the reconstructed (001) surface of diamond.

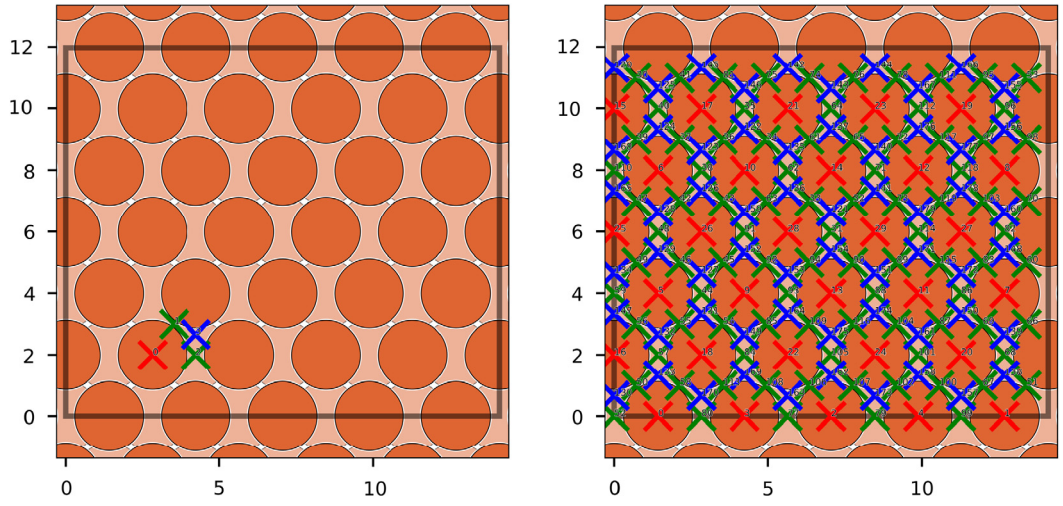
2. `Xsorb` structure and functioning

As described in the schematic flowchart of `Xsorb` shown in Fig. 1, its main feature is the automatic identification of all the substrate adsorption sites and the generation of all the desired adsorption configurations by applying several molecule rotations for each site. After this initial procedure of input generation, the code automatically launches the DFT calculations. It also provides valuable features to analyze the results, such as extracting the adsorption energies, monitoring the energy evolution during structural optimization, and generating images or animations of the output files.

2.1. Defining the setting parameters

The main settings for the calculation, such as the QE input commands and the list of molecular rotations, must be provided in a `settings.in` file, located in the folder from which the `xsorb` command is executed. A template of the `settings.in` file is provided in the main folder of the repository. The following mandatory flags should be present in the `settings.in` file to generate the adsorption configurations:

1. `slab_filename` and `molecule_filename` are the names of the two files containing the slab and molecule coordinates, respectively. These optimized structures are retrieved to generate the adsorbed structure.
2. `molecule_axis` is the molecular axis that will be placed along the x direction and considered for rotating the molecule with respect to the substrate. The axis can be defined in two ways: i) The user indicates two atoms of the molecule using



(a) Non-equivalent adsorption sites. (b) All high-symmetry sites.

Fig. 2. Adsorption sites of the Fe(110) surface. The non-equivalent adsorption sites (Panel a) and all possible sites in the supercell (Panel b) are shown. The three symmetry types (on-top, bridge, hollow) are colored in red, green and blue, respectively. A darker (lighter) color is used to indicate shallower (deeper) atoms. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

the variables `atoms [A1] [A2]`, where `Ai` are numeric indexes from `molecule_filename`, starting from 0. The axis will contain the vector $\mathbf{a} = \mathbf{r}_{A2} - \mathbf{r}_{A1}$. ii) The user explicitly provides the coordinates of the axis vector, `vector` $[v_x]$ $[v_y]$ $[v_z]$.

3. `selected_atom_index` identifies the molecule atom that will be placed in the selected adsorption sites and used as the origin of the reference frame for rotations. The numeric index must be consistent with the order in the `molecule_filename`.
4. `x_rot_angles`, `y_rot_angles` and `z_rot_angles` are the variables that provide a single angle or a list of angles for which the molecule will be rotated around the axis identified by the variable name. The rotations are executed in the following order: first around the x axis, then around the $-y$ axis and finally around the z axis of the reference frame. Following this procedure, all angle combinations are produced. Note that the rotations are always considered along a fixed Cartesian reference frame. This operation implies that, if a rotation `y_rot_angles` = 90° is applied, the `molecule_axis`, which previously was along the x axis, now coincides with the z axis. Therefore, to avoid the screening of already sampled molecular orientations around the `molecule_axis`, rotations defined by `z_rot_angles` are disregarded.

The desired distance between the reference atom of the molecule and the surface adsorption site can be set with the `screening_atom_distance` variable (the default value is 2 Å). To avoid possible overlapping between the molecule and the substrate, an additional variable `screening_min_distance` (with a default value of 1.5 Å) is implemented so that all the atoms of the molecule have a minimal vertical displacement from the surface. A final check on the Euclidean distance is done automatically for all the atoms in the molecule to avoid overlapping with the substrate. This control is essential in atomically rough surfaces such as reconstructed surfaces. Please note that the methods implemented to avoid overlapping between atoms is only performed for the molecule and the slab, and no check is performed for the overlap between the molecule and its periodic replicas, arising due to periodic boundary conditions. The user should therefore check beforehand, i.e., when constructing the atomistic model given as input to the program, that the simulation cell is large enough to host the molecule in the directions along which it will be placed with the different rotations.

2.2. Generation of adsorption configurations

`Xsorb` can identify the adsorption sites thanks to the `AdsorbateSiteFinder` class in the Pymatgen library. This class and its related commands, based on the Delaunay triangulation [42] on the topmost surface layer, are very effective in finding all the adsorption sites within the simulation cell. Moreover, for flat and clean surfaces, it also identifies the subset of non-equivalent sites with respect to surface symmetries, as shown in Fig. 2. However, for more complex substrates, like oxides and doped surfaces, a large number of sites is identified due to reduced surface symmetry. In these cases some manual tuning is necessary to select the most relevant ones, as explained below.

The command `xsorb -sites` generates a top-view image of the simulation cell where the identified adsorption sites are marked by crosses of different colors to distinguish the different site symmetries (namely, on-top, hollow, and bridge). If unexpected results are presented in this image, e.g., some sites are misplaced, it is possible to use the command `xsorb -sites-all` to obtain all the adsorption sites. In this case, the user should check if there are incorrect or missing on-top sites, adjusting the `surface_height` variable in `settings.in` until the on-top sites are identified correctly. This flag controls the Δz value (default 0.9 Å) employed by the `AdsorbateSiteFinder` class to define the topmost layer of the surface, whose atoms constitute the triangular vertices for the Delaunay triangulation and therefore also determine the positions of all the other adsorption sites.

Another issue that could arise in this automatic identification is related to the presence of too many (or too few) sites, leading to an over (or under) representation of adsorption configurations. Therefore, the user can increase (reduce) the `symm_reduce` variable (with default value of 0.01), used by Pymatgen `AdsorbateSiteFinder` class to reduce (increase) the number of identified adsorption sites.

Some situations could be even more challenging for site identification. For example, a substrate with a single dopant atom can completely break the symmetry of the surface, producing an excessive number of non-equivalent adsorption sites that could lead to an unfeasible screening of adsorption configurations. Moreover,

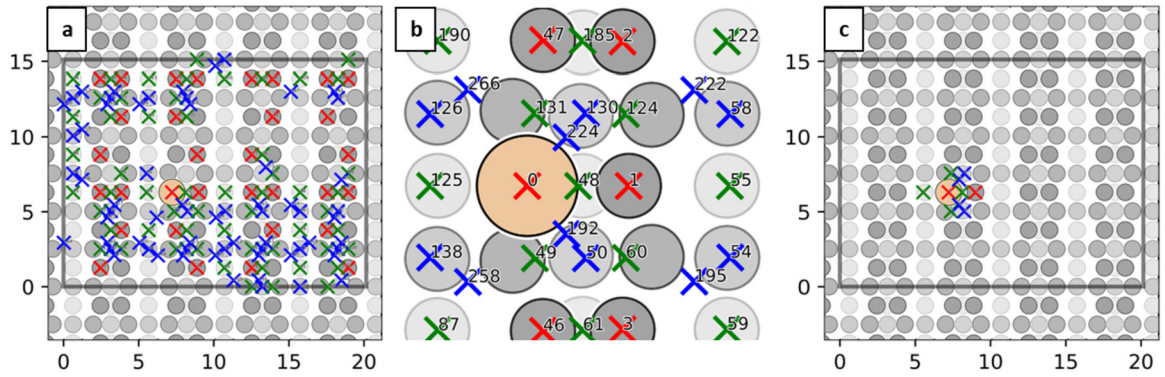


Fig. 3. Adsorption sites of the Si-doped reconstructed C(001) surface. In panel *a*, the sites identified with the default value of `symm_reduce` are reported. In panel *b*, a region around the Si dopant with all adsorption sites obtained with `xsorb -sites-all` is shown. The numeric index on each site allows the manual site selection. Panel *c* shows the result of the manual selection. The sites are colored in red, green and blue according to the three symmetry types (on-top, bridge, and hollow, respectively). A darker (lighter) color is used for shallower (deeper) C atom.

one might be interested in studying only the sites close to the dopant, neglecting the undoped regions. For these specific cases, the user can impose `symm_reduce = 0` to obtain all adsorption sites and then manually select the required ones by specifying their indexes with the flag `selected_sites`.

An example of this procedure is represented in Fig. 3, where the adsorption sites in a Si-doped diamond surface are represented. The default value of `symm_reduce` produces an unreasonable number of adsorption sites, distributed in a "random" way (see Fig. 3, panel a). In this case, the user should use the command `xsorb -sites-all`, in order to inspect all the adsorption sites, and manually choose the desired ones from their numeric index, as shown in Fig. 3, panel b, where the sites around the Si dopant are represented. The settings.in file can then be modified, specifying `symm_reduce = 0` and including in `selected_sites` the chosen sites. After this procedure, the execution of the command `xsorb -sites` now shows the subset of user-selected sites (Fig. 3, panel c), which will be used to generate all the adsorption configurations.

After the adsorption site identification, `Xsorb` generates all the adsorption configurations. Their total number is obtained by multiplying the number of selected adsorption sites by the number of provided rotations along the *x*, *y* and *z* axes.

2.3. Executing DFT calculations

In order to create the input files to perform the DFT calculations for the generated configurations, it is necessary to specify the computational parameters in the `settings.in` file using the standard syntax of QE input files (see the user guide for specific details).

As mentioned in the introduction, two types of calculations are performed: a preliminary screening to identify the most relevant configurations with the lowest energies, and a full structural optimization of a subset of these configurations, selected automatically by a pre-defined energy threshold.

2.3.1. Preliminary screening

The preliminary screening consists of a partial structural optimization executed by considering higher energy and force thresholds compared to the default values of QE. Typically, thresholds of 5×10^{-3} Ry for the energy and 5×10^{-2} Ry/bohr for the forces allow one to obtain an estimate of the most promising configurations.

This screening method, compared to the other techniques described in the introduction, is completely DFT-based, allowing for greater accuracy in capturing the electronic interactions between the molecule and the substrate than empirical or semi-empirical methods. With this technique, we can also immediately identify the presence of dissociation paths over the whole set of tested configurations, since they often occur within the first few optimization steps.

The preliminary screening can be executed by including the `jobscript` flag into the `settings.in` file. This flag specifies the path of a jobscript compatible with the job scheduler (e.g., Slurm) installed on the machine where QE is executed and the command for job submission, e.g. `sbatch` for Slurm. The calculations can be launched with the following command:

```
xsorb -s [etot_conv_thr forc_conv_thr]
```

where `etot_conv_thr` (`forc_conv_thr`) are optional, with default values of 5×10^{-3} Ry (5×10^{-2} Ry/bohr).

This command submits the jobs and labels each configuration with an integer index starting from 0. It also generates the `site_labels.csv` file, schematically represented in Table 1, which contains all the information of each configuration with a label, the applied rotations, the adsorption site coordinates, and its symmetry. The latter is expressed in the following way:

- `ontop_`*{atomic species}*, where *atomic species* indicates the species of the surface atom.
- `bridge_`*{bridge length}*, where *bridge length* indicates the distance between the two surface atoms that identify the bridge site.
- `hollow_c`*{coordination number}*, where *coordination_number* corresponds to the number of first neighboring sites.

For each site, a numeric index matching the one in the figure generated with the option `-sites` is also included.

2.3.2. Retrieving energies after the preliminary screening

Once the preliminary screening is completed, all the energies can be extracted with the command:

```
xsorb -es
```

that generates the `screening_energies.csv` file containing a label for each adsorption configuration, the symmetry of the adsorption site, and the adsorption energies E_{ads} of the screening. If the energy of the isolated molecule/surface is not available, the file reports the total energies E_{tot} instead.

The adsorption energies E_{ads} can be obtained from the E_{tot} as:

$$E_{ads} = E_{tot} - (E_{surf} + E_{mol}) \quad (1)$$

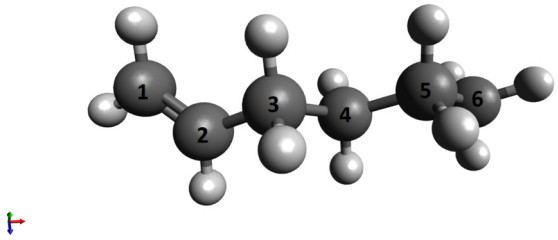


Fig. 4. 1-hexene molecule, presenting double bond between the carbon atoms labelled 1 and 2.

where E_{surf} (E_{mol}) is the energy of the system containing the isolated slab (molecule). With this notation, a negative (positive) adsorption energy means a favorable (unfavorable) adsorption configuration. The slab and molecule energies can be either manually specified by the `E_slab_mol` flag contained in the *settings.in* file or extracted from the two QE output files if provided.

2.3.3. Full geometrical optimization

After the screening, it is possible to perform the full geometrical optimization of the most relevant configurations with the following command:

```
xsorb -r [options]
```

Without any option, Xsorb will only perform the optimization for the five configurations with lowest energy obtained from the preliminary screening. This number can be changed by employing the option `--n NUMBER`. Alternatively, the configurations that will be optimized can be selected by specifying an energy threshold with the option `--t ENERGY`. In this way, all the configurations with an `ENERGY` (in eV) above the identified minimum will be considered for the full optimization. It is also possible to exclude configurations that would be otherwise included with the two aforementioned selection methods. The user can do that by using the flag `--exclude`, as in the following example:

```
xsorb -r --t 0.5 --exclude 1 3 7
```

where `Xsorb` launches the optimization of all configurations within 0.5 eV from the energy minimum of the screening, excluding the ones labelled with 1, 3 and 7. Alternatively, the user can provide a list of specific configurations to be optimized using the flag `--i`, specifying their labels.

2.3.4. Retrieving final energies after optimizations

After performing the geometrical optimizations, the user can retrieve their final energies with the command:

```
xsorb -er
```

That creates the `relax_energies.csv` file, which contains a label for each adsorption configuration, the symmetry of the adsorption site, the total energies E_{tot} of the screening and the total energy of the optimized configuration. In the same way as for the screening, `Xsorb` can print the optimized adsorption energies when the user provides the energies of the isolated slab and molecule.

3. Installation and execution of `Xsorb`

The installation procedure for `Xsorb` is easy and straightforward since it is Python-based. It is first necessary to download the program by cloning this repository into the desired local machine:

```
git clone https://gitlab.com/triboteam/xsorbed.git
```

Once the download is completed, it is necessary to go to the `xsorbed` main folder and run:

```
bash install.sh
```

to add the executable in the `PATH` variable. With this operation, the user can launch the program from any folder. The next step is to install the required dependencies stored in the `requirements.txt` file. This step can be performed with the command:

```
pip install -r requirements.txt
```

Further details for the installation procedure, like adding the POV-Ray visualization tool, can be found in the user guide. `Xsorb` is interfaced with QE to perform the DFT calculations. It is, therefore, necessary to compile and install the `pw.x` executable before launching the calculations.

After the installation, the program works through a command-line interface (CLI) by running:

```
xsorb [command] [parameters]
```

Where `command` specifies which operation to perform and `parameters` are additional parameters available for specific commands.

4. Example of `Xsorb` use

The use of `Xsorb` is exemplified in the following sections, where the results obtained for the adsorption of a hydrocarbon molecule, hexene, on a simple transition metal surface, namely Fe(110), and an insulating substrate, namely the reconstructed C(001) surface are reported.

The atomistic structure for the hexene, represented in Fig. 4, was initially taken from the PubChem database [43], while the two surfaces were generated using Pymatgen. We performed a geometrical optimization separately for each system.

For the DFT calculations, we used an energy cutoff of 40 Ry and a k-point sampling in Γ. The choice of using only the Γ-point was justified by the large size of the cells ($14.1 \times 12.0 \times 24.0$ Å^3 for Fe, containing 120 atoms, and $20.2 \times 15.1 \times 24.0$ Å^3 for C, containing 480 C atoms), since denser k-point grids produced total energy differences lower than 1 meV/atom. van der Waals corrections were applied using the Grimme-D2 [44] scheme as implemented in QE.

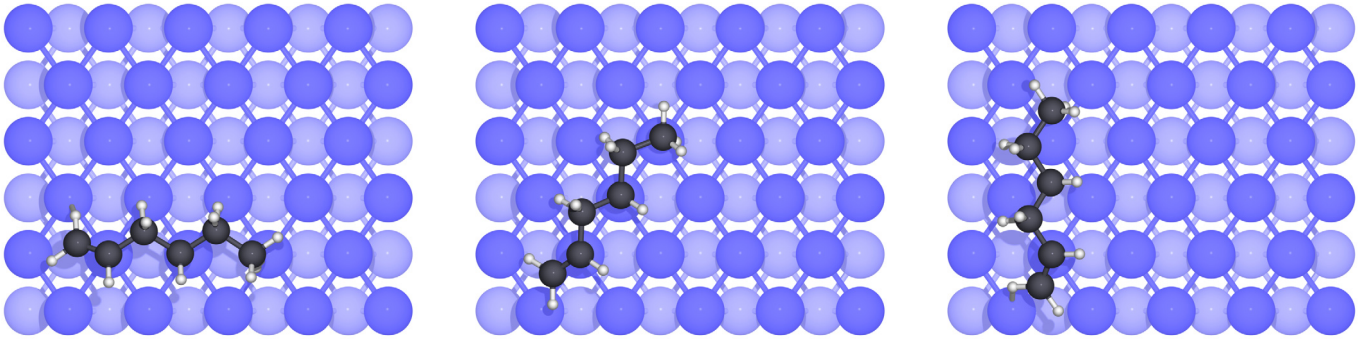
4.1. Hexene adsorption on Fe(110)

4.1.1. Generation of adsorption configurations

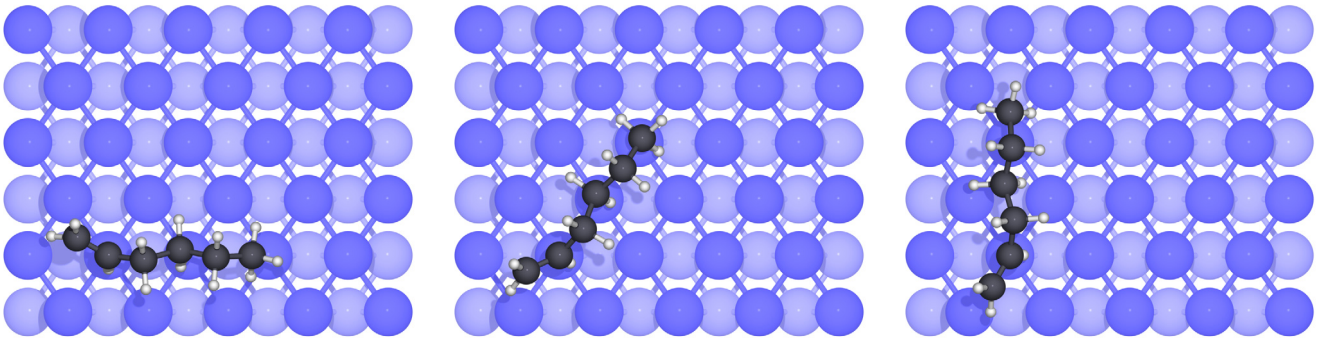
In the case of hexene, the molecule atoms that are more likely to interact with the surface are the two carbon atoms sharing the double bond. Thus, we selected the C atom with only one hydrogen, labelled 2 in Fig. 4, as the reference atom.

Even if hexene has a C_1 point group symmetry, i.e., the only symmetrical operation available is the identity, it is a quite elongated molecule. Thus, to define the `molecule_axis`, two atoms along this preferential direction were taken. In particular, we chose the C atoms labelled as 2 and 6 in Fig. 4 to identify this axis.

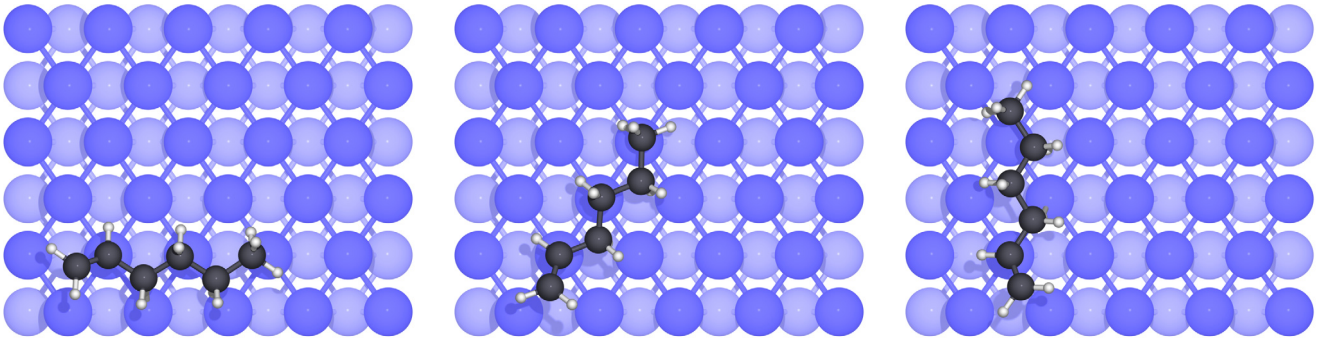
Finally, to generate all the adsorption structures, it is necessary to define the molecular rotations. This procedure partially relies on user intuition, but a sensible choice can be done by examining both the surface symmetries and the molecular structure. Since hexene has no internal rotational symmetries, we started by considering the relevant rotations around the x axis, namely 0°, 90° and 180°. Moreover, due to the molecular elongation, we selected two additional rotations around the y axis: 0°, which corresponds to the molecule parallel to the surface, and 90°, corresponding to a



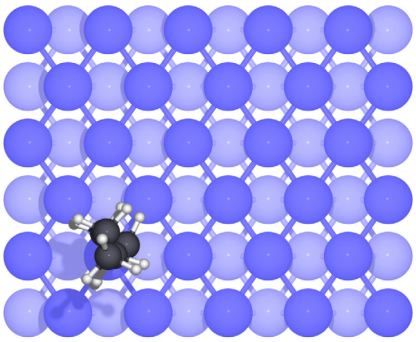
(a) Label: 0 (b) Label: 4 (c) Label: 8



(d) Label: 12 (e) Label: 16 (f) Label: 20



(g) Label: 24 (h) Label: 28 (i) Label: 32



(j) Label: 36

Fig. 5. Adsorption configuration for hexene on Fe(110) for the preliminary screening. The reference atom is positioned at the on-top site, and all the selected molecular rotations are represented. Other 30 configurations are explored by positioning the reference atom in bridge and hollow sites. The configurations are labelled according to the labels in Table 1.

Table 1
Data for the study of hexene adsorption on the Fe(110) surface. The data reported in each column corresponds to the numerical label of the configuration, rotations in degrees along x (x_{rot}), y (y_{rot}) and z (z_{rot}) axis, the index (as in Fig. 2) and the symmetry of the adsorption site, the distance (d) between the reference atom and the surface (in Å) and the adsorption energies (E_{ads}) of the preliminary screening and of the full optimization, in eV.

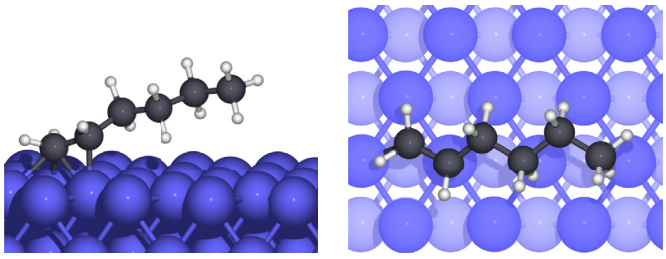
Label	x_{rot}	y_{rot}	z_{rot}	Symm. site	d	Screen. E_{ads}	Opt. E_{ads}
0	0	0	0	0 ontop_Fe	2.76	-1.45	-1.83
4	0	0	54.7	0 ontop_Fe	2.76	-1.38	-1.81
8	0	0	90	0 ontop_Fe	2.76	-1.13	-1.78
1	0	0	0	1 bridge_2.44	2.76	-0.36	-1.80
11	0	0	90	3 hollow_c3	2.76	-0.34	-1.68
6	0	0	54.7	2 bridge_2.82	2.76	-0.25	-
10	0	0	90	2 bridge_2.82	2.76	-0.23	-
5	0	0	54.7	1 bridge_2.44	2.76	-0.20	-
9	0	0	90	1 bridge_2.44	2.76	-0.19	-
27	180	0	0	3 hollow_c3	2.70	-0.19	-
7	0	0	54.7	3 hollow_c3	2.76	-0.18	-
3	0	0	0	3 hollow_c3	2.76	-0.16	-
30	180	0	54.7	2 bridge_2.82	2.70	-0.14	-
34	180	0	90	2 bridge_2.82	2.70	-0.13	-
33	180	0	90	1 bridge_2.44	2.70	-0.09	-
16	90	0	54.7	0 ontop_Fe	2.49	-0.08	-
12	90	0	0	0 ontop_Fe	2.49	-0.07	-
25	180	0	0	1 bridge_2.44	2.70	-0.07	-
2	0	0	0	2 bridge_2.82	2.76	-0.06	-
35	180	0	90	3 hollow_c3	2.70	-0.05	-
31	180	0	54.7	3 hollow_c3	2.70	-0.03	-
24	180	0	0	0 ontop_Fe	2.70	-0.03	-
29	180	0	54.7	1 bridge_2.44	2.70	-0.03	-
32	180	0	90	0 ontop_Fe	2.70	-0.02	-
26	180	0	0	2 bridge_2.82	2.70	-0.02	-
38	0	90	0	2 bridge_2.82	3.50	0.01	-
28	180	0	54.7	0 ontop_Fe	2.70	0.03	-
37	0	90	0	1 bridge_2.44	3.50	0.04	-
21	90	0	90	1 bridge_2.44	2.49	0.07	-
36	0	90	0	0 ontop_Fe	3.50	0.12	-
23	90	0	90	3 hollow_c3	2.49	0.13	-
20	90	0	90	0 ontop_Fe	2.49	0.14	-
18	90	0	54.7	2 bridge_2.82	2.49	0.14	-
39	0	90	0	3 hollow_c3	3.50	0.16	-
22	90	0	90	2 bridge_2.82	2.49	0.17	-
13	90	0	0	1 bridge_2.44	2.49	0.18	-
15	90	0	0	3 hollow_c3	2.49	0.23	-
19	90	0	54.7	3 hollow_c3	2.49	0.31	-
14	90	0	0	2 bridge_2.82	2.49	0.35	-
17	90	0	54.7	1 bridge_2.44	2.49	0.40	-

vertical orientation. We considered the surface symmetries for the rotations around the z axis and chose three orientations equal to 0°, 54.7° and 90°. For what concerns the vertical distance between the surface and the molecule, we decided to keep the `screening_atom_distance` and `screeening_min_distance` variables to their default values, i.e., 2 and 1.5 Å, respectively. We performed the automatic identification of the adsorption sites through the `xsorb -sites` command. As shown in Fig. 2, the Fe(110) is a flat and symmetric surface, and the default settings were sufficient to identify all the non-equivalent adsorption sites correctly.

A visual representation of the adsorption configurations generated for the on-top site can be seen in Fig. 5. Other 30 configurations correspond to the other two adsorptions sites.

4.1.2. Preliminary screening and full geometrical optimization

After the generation of all the adsorption structures, we performed the preliminary screening. The most relevant data obtained for each configuration are reported in Table 1. Regarding the vertical distance between the reference atom and the surface, d, it is possible to notice that the program automatically moved the molecule upwards to satisfy the minimum distance requirement and avoid any overlap. Moreover, since this operation increased the vertical distance, it avoided the dissociation of hydrogen atoms which are too close to the substrate.



(a) Lateral view. (b) Top view.

Fig. 6. Lateral (panel a) and top (panel b) representation of the adsorbed structure after the full geometrical optimization for the configuration with minimum energy (labelled 0). The different shades of colors for the Fe atoms represent the different depths of the atomic layers.

The last step in studying hexene adsorbed on Fe(110) is the full geometrical optimization of specific configurations. In particular, we selected the five configurations with lower energy from the screening. The final optimized adsorption energies are also reported in Table 1, and the fully optimized structure corresponding to the global energy minimum is represented in Fig. 6. This adsorption configuration, labelled by 0, with −1.83 eV energy, presents the hexene double bonded C-C group anchored between the 3-fold and the on-top sites, and the rest of the molecule shifted upwards from the Fe(110) surface.

The preliminary screening method consisting of a partial structural optimization was in this case very effective in predicting the configuration associated with the energy minimum after the full optimization. Indeed, the ordering of the energies in the screening is almost identically reproduced after the optimization (with a single swap between configuration 8 and configuration 1).

The computational cost of the screening was also quite contained, since for almost all configurations (with only three exceptions) the number of ionic steps in the screening was between 2 and 5, while the full optimizations of this system required between 70 and 120 ionic steps.

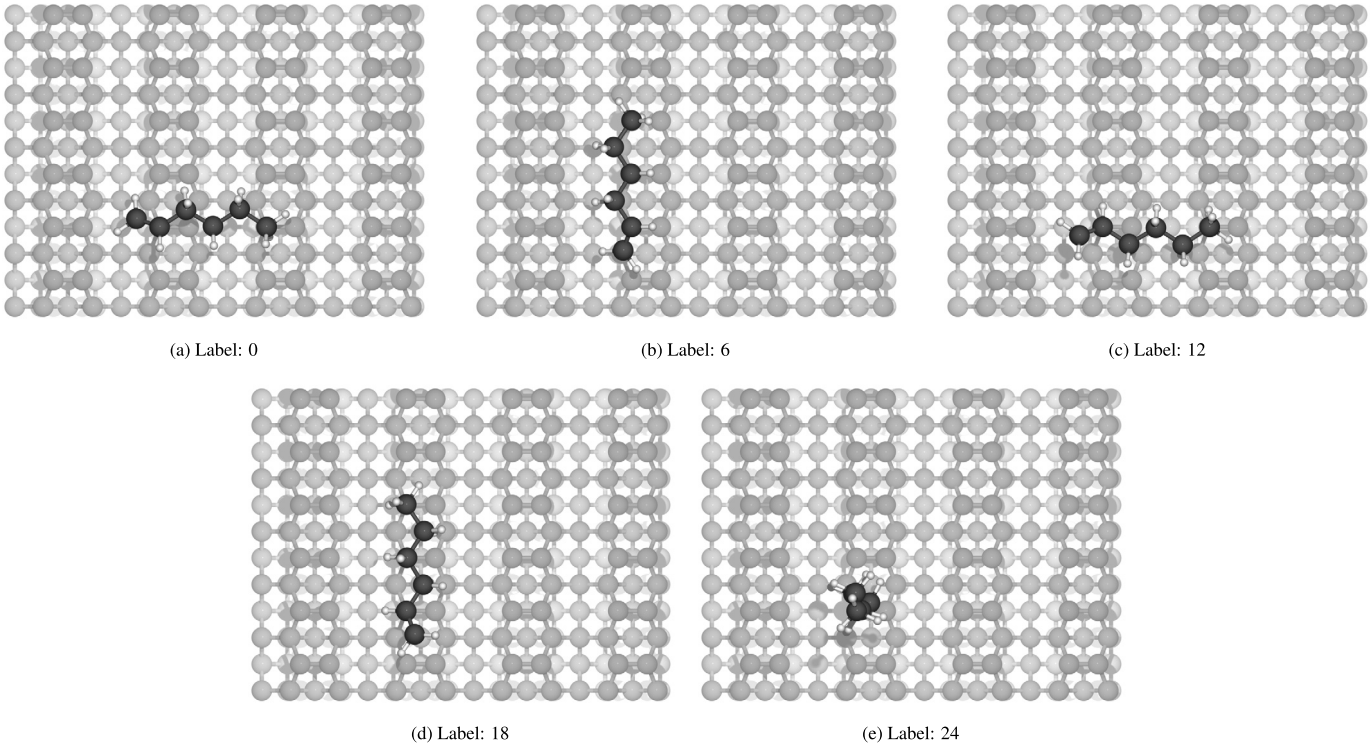
4.2. Hexene on reconstructed C(001)

We considered hexene physisorption on the reconstructed C(001) surface as an example of a more complex surface due to its rough morphology compared to the flat Fe(110).

For this case, we chose the reference atom for the molecule and the `molecule_axis` in the same way as for the Fe(110) calculations. We also considered similar molecular rotations, apart from the 54.7° rotation around the z axis, which we neglected due to the different surface symmetry of C(001) compared to Fe(110). We also excluded and the 90° rotation around the molecular axis (x axis), since all the configurations which included this rotation were the least energetically favorable in the Fe(110) screening.

Regarding the molecule-surface distance, we left the `screening_atom_distance` and `screening_min_distance` variable to the default values, which were sufficient to avoid any dissociation of hydrogen atoms.

Following the same procedure employed for the Fe(110) substrate, the adsorption sites were identified, obtaining the correct number of sites by reducing the `surface_height` variable to 0.5 Å. This change was necessary to avoid that the C atoms just below the surface dimers had been included as top sites, resulting in the wrong identification of the bridge and hollow sites. For the `symm_reduce` parameter, instead, the default value was sufficient even for a rough surface such as the reconstructed C(001). This example shows the effectiveness of Pymatgen in identifying the



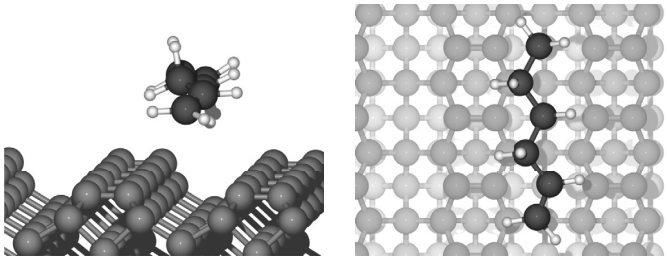
(a) Label: 0 (b) Label: 6 (c) Label: 12

(d) Label: 18 (e) Label: 24

Fig. 7. Adsorption configuration for hexene on reconstructed C(001) for the preliminary screening. The reference atom is positioned at the on-top site, and all the selected molecular rotations are represented. Other 20 configurations are explored by positioning the reference atom in bridge and hollow sites.

Table 2
Data relative to the study of hexene physisorption on the reconstructed C(001) surface. The data reported in each column correspond to the numerical label of the configuration, rotations in degrees along x (x_{rot}), y (y_{rot}) and z (z_{rot}) axis, the index and the symmetry of the adsorption site, the distance (d) between the reference atom and the surface (in Å) and the adsorption energies (E_{ads}) of the preliminary screening and of the full optimization, in eV. The systems are sorted according to their screening adsorption energy.

Label	x_{rot}	y_{rot}	z_{rot}	Symm. site	d	Screen. E_{ads}	Opt. E_{ads}
3	0	0	0	3 bridge_1.38	2.76	-0.54	-
5	0	0	0	5 bridge_3.66	2.76	-0.31	-0.51
22	180	0	90	4 hollow_c4	2.70	-0.28	-0.59
8	0	0	90	2 bridge_2.52	2.76	-0.24	-0.54
11	0	0	90	5 bridge_3.66	2.76	-0.23	-0.65
7	0	0	90	1 hollow_c4	2.76	-0.19	-0.41
20	180	0	90	2 bridge_2.52	2.70	-0.19	-
9	0	0	90	3 bridge_1.38	2.76	-0.17	-
23	180	0	90	5 bridge_3.66	2.70	-0.13	-
6	0	0	90	0 ontop_C	2.76	-0.12	-
1	0	0	0	1 hollow_c4	2.76	-0.10	-
10	0	0	90	4 hollow_c4	2.76	-0.07	-
28	0	90	0	4 hollow_c4	3.50	-0.02	-
29	0	90	0	5 bridge_3.66	3.50	-0.02	-
16	180	0	0	4 hollow_c4	2.70	0.02	-
17	180	0	0	5 bridge_3.66	2.70	0.11	-
13	180	0	0	1 hollow_c4	2.70	0.28	-
14	180	0	0	2 bridge_2.52	2.70	0.32	-
2	0	0	0	2 bridge_2.52	2.76	0.34	-
19	180	0	90	1 hollow_c4	2.70	0.36	-
27	0	90	0	3 bridge_1.38	3.50	0.44	-
0	0	0	0	0 ontop_C	2.76	0.48	-
12	180	0	0	0 ontop_C	2.70	0.52	-
4	0	0	0	4 hollow_c4	2.76	0.55	-
26	0	90	0	2 bridge_2.52	3.50	0.55	-
15	180	0	0	3 bridge_1.38	2.70	0.58	-
25	0	90	0	1 hollow_c4	3.50	0.60	-
21	180	0	90	3 bridge_1.38	2.70	0.76	-
24	0	90	0	0 ontop_C	3.50	0.79	-
18	180	0	90	0 ontop_C	2.70	0.83	-



(a) Lateral view. (b) Top view.

Fig. 8. Lateral (panel a) and top (panel b) representation of the adsorbed structure after the full geometrical optimization for the configuration with minimum energy (labelled 0). The different shades of colors for the C atoms represent the different depth of the atomic layers.

non-equivalent adsorption sites, even for more complex surfaces, as long as there are no dopants that break the surface symmetries. A representation of the generated adsorption configurations for the on-top site can be seen in Fig. 7. The results of the preliminary screening are reported in Table 2, together with the relevant information for each configuration.

Finally, we optimized a subset of configurations to identify the most stable one. In particular, we chose the first five configurations starting from the one with minimum energy, excluding the configuration labelled by 3. We avoided optimizing this configuration since it was the only one where the hexene was chemically bonded to the surface, being the goal of this example the study of physisorption configurations.

We report the results of the adsorption energy after the full optimization in Table 2, and the fully optimized structure corresponding to the global energy minimum in Fig. 8. All the optimized structures exhibit a similar planar disposition, with the molecule parallel to the surface. We identified that the most stable physisorption configuration is the one with label 11, with physisorption energy equal to -0.65 eV. In this configuration, hexene

is placed between two dimer rows, with the first C atom of the molecule (labelled by 1 in Fig. 4) turned downwards. Between the five fully optimized structures, the two with lower energy were both placed above the trenches between two dimer rows. The other three configurations were instead above a dimer row, either oriented parallel or perpendicular to it. These five configurations are thus quite representative of all the possible orientations of hexene over C(001) the surface. It is, therefore, reasonable to conclude that the physisorption in the trench between two dimer rows is the most favorable one and that it is very unlikely that other not fully optimized configurations would reach lower physisorption energies.

As for the previous surface, the computational cost of the preliminary screening was not prohibitive. The number of optimization steps ranged between 0 and 15 for all the configurations, being below 6 steps for one-third of them. The number of steps for the full relaxation ranged instead between 50 and 90. For both the studied surfaces, the configurations that reached a larger number of ionic steps in the screening were often those with lower energy. Therefore, in many cases, computational resources are not wasted, since the final positions after the partial optimization are used as starting positions for the full relaxations.

5. Summary

In this work, we present `Xsorb`, a Python-based code that automatically generates molecular adsorption configurations, guides the user in the identification of the most relevant ones, which are then fully optimized. The code relies on well-established Python libraries, and on an open-source package for density functional theory calculations. As an example of application, we presented the study 1-hexene adsorption on both the Fe(110) and the reconstructed C(001) surfaces. This approach can significantly reduce the workload usually associated with the study of molecular adsorption and can help the users, even non-expert, to quickly identify the most favorable adsorption configuration, thanks to a user-friendly Python interface and automatized imaging.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

These results are part of the SLIDE project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Grant Agreement No. 865633).

References

[1] T. Bligaard, J. Nørskov, S. Dahl, J. Matthiesen, C. Christensen, J. Sehested, J. Catal. 224 (1) (2004) 206–217, https://doi.org/10.1016/j.jcat.2004.02.034, https://www.sciencedirect.com/science/article/pii/S0021951704001010.

[2] M.P. Andersson, T. Bligaard, A. Kustov, K.E. Larsen, J. Greeley, T. Johannessen, C.H. Christensen, J.K. Nørskov, J. Catal. 239 (2) (2006) 501–506, https://doi.org/10.1016/j.jcat.2006.02.016, https://www.sciencedirect.com/science/article/pii/S0021951706000674.

[3] C. Joachim, M.A. Ratner, Proc. Natl. Acad. Sci. 102 (25) (2005) 8801–8808, https://doi.org/10.1073/pnas.0500075102, https://www.pnas.org/content/102/25/8801.

[4] B. Kasemo, Surf. Sci. 500 (1) (2002) 656–677, https://doi.org/10.1016/S0039-6028(01)01809-X, https://www.sciencedirect.com/science/article/pii/S003960280101809X.

[5] B. Hinnemann, P.G. Moses, J. Bonde, K.P. Jørgensen, J.H. Nielsen, S. Horch, I. Chorkendorff, J.K. Nørskov, J. Am. Chem. Soc. 127 (15) (2005) 5308–5309, https://doi.org/10.1021/ja0504690, https://pubs.acs.org/doi/10.1021/ja0504690.

[6] A. Neville, A. Morina, T. Haque, M. Voong, Tribol. Int. 40 (10) (2007) 1680–1695, https://doi.org/10.1016/j.triboint.2007.01.019, https://www.sciencedirect.com/science/article/pii/S0301679X07000230.

[7] S. Peeters, P. Restuccia, S. Loehlé, B. Thiebaut, M.C. Righi, J. Phys. Chem. A 123 (32) (2019) 7007–7015, https://doi.org/10.1021/acs.jpca.9b03930.

[8] M. Finšgar, J. Jackson, Corros. Sci. 86 (2014) 17–41, https://doi.org/10.1016/j.corsci.2014.04.044, https://www.sciencedirect.com/science/article/pii/S0010938X14002157.

[9] K. Kousar, M. Walczak, T. Ljungdahl, A. Wetzel, H. Oskarsson, P. Restuccia, E. Ahmad, N. Harrison, R. Lindsay, Corros. Sci. 180 (2021) 109195, https://doi.org/10.1016/j.corsci.2020.109195, https://www.sciencedirect.com/science/article/pii/S0010938X20324768.

[10] I. Minami, Appl. Sci. 7 (5) (2017), https://doi.org/10.3390/app7050445, https://www.mdpi.com/2076-3417/7/5/445.

[11] M. Kermani, A. Morshed, Corrosion 59 (08) (2003) 659–683, https://onepetro.org/corrosion/article/116393/Carbon-Dioxide-Corrosion-in-Oil-and-Gas.

[12] R. Kecili, C.M. Hussain, in: C.M. Hussain (Ed.), Nanomaterials in Chromatography, Elsevier, 2018, pp. 89–115, https://www.sciencedirect.com/science/article/pii/B9780128127926000042.

[13] C. Di Valentin, F. Wang, G. Pacchioni, Top. Catal. 56 (15) (2013) 1404–1419, https://doi.org/10.1007/s11244-013-0147-6, https://link.springer.com/article/10.1007/s11244-013-0147-6.

[14] K. Tran, Z.W. Ulissi, Nat. Catal. 1 (9) (2018) 696–703, https://doi.org/10.1038/s41929-018-0142-1, https://www.nature.com/articles/s41929-018-0142-1.

[15] G. Hong, H. Heinz, R.R. Naik, B.L. Farmer, R. Pachter, ACS Appl. Mater. Interfaces 1 (2) (2009) 388–392, https://doi.org/10.1021/am800099z, https://pubs.acs.org/doi/full/10.1021/am800099z.

[16] J.H. Montoya, K.A. Persson, npj Comput. Mater. 3 (1) (2017) 14, https://doi.org/10.1038/s41524-017-0017-z, https://www.nature.com/articles/s41524-017-0017-z.

[17] K. Mathew, A.K. Singh, J.J. Gabriel, K. Choudhary, S.B. Sinnott, A.V. Davydov, F. Tavazza, R.G. Hennig, Comput. Mater. Sci. 122 (2016) 183–190, https://doi.org/10.1016/j.commatsci.2016.05.020, http://www.sciencedirect.com/science/Article/pii/S0927025616302440.

[18] O. Borodin, M. Olguin, C.E. Spear, K.W. Leiter, J. Knap, Nanotechnology 26 (35) (2015) 354003, https://doi.org/10.1088/0957-4484/26/35/354003, https://iopscience.iop.org/article/10.1088/0957-4484/26/35/354003.

[19] A. Jain, S.P. Ong, G. Hautier, W. Chen, W.D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, K.A. Persson, APL Mater. 1 (1) (2013) 011002, https://doi.org/10.1063/1.4812323, https://aip.scitation.org/doi/10.1063/1.4812323.

[20] R. Tran, Z. Xu, B. Radhakrishnan, D. Winston, W. Sun, K.A. Persson, S.P. Ong, Sci. Data 3 (1) (2016) 160080, https://doi.org/10.1038/sdata.2016.80, https://www.nature.com/articles/sdata201680.

[21] K. Tran, A. Palizhati, S. Back, Z.W. Ulissi, J. Chem. Inf. Model. 58 (12) (2018) 2392–2400, https://doi.org/10.1021/acs.jcim.8b00386.

[22] J.R. Boes, O. Mamun, K. Winther, T. Bligaard, J. Phys. Chem. A 123 (11) (2019) 2281–2285, https://doi.org/10.1021/acs.jpca.9b00311.

[23] S. Pablo-García, M. Álvarez Moreno, N. López, Int. J. Quant. Chem. 121 (1) (2021) e26382, https://doi.org/10.1002/qua.26382, https://onlinelibrary.wiley.com/doi/abs/10.1002/qua.26382.

[24] A. Jain, S.P. Ong, W. Chen, B. Medasani, X. Qu, M. Kocher, M. Brafman, G. Petretto, G.-M. Rignanese, G. Hautier, D. Gunter, K.A. Persson, Concurr. Comput., Pract. Exp. 27 (17) (2015) 5037–5059, https://doi.org/10.1002/cpe.3505, https://onlinelibrary.wiley.com/doi/10.1002/cpe.3505.

[25] K. Mathew, J.H. Montoya, A. Faghaninia, S. Dwarakanath, M. Aykol, H. Tang, I. heng Chu, T. Smidt, B. Bocklund, M. Horton, J. Dagdelen, B. Wood, Z.-K. Liu, J. Neaton, S.P. Ong, K. Persson, A. Jain, Comput. Mater. Sci. 139 (2017) 140–152, https://doi.org/10.1016/j.commatsci.2017.07.030, https://www.sciencedirect.com/science/Article/pii/S0927025617303919.

[26] S.A. Wilson, C.L. Muhich, Comput. Theor. Chem. 1216 (2022) 113830, https://doi.org/10.1016/j.comptc.2022.113830, https://www.sciencedirect.com/science/article/pii/S2210271X22002432.

[27] C. Martí, S. Blanck, R. Staub, S. Loehlé, C. Michel, S.N. Steinmann, J. Chem. Inf. Model. 61 (7) (2021) 3386–3396, https://doi.org/10.1021/acs.jcim.1c00256.

[28] S.N. Steinmann, A. Hermawan, M. Bin Jassar, Z.W. Seh, Chem Catal. 2 (5) (2022) 940–956, https://doi.org/10.1016/j.checat.2022.02.009, https://www.sciencedirect.com/science/article/pii/S266710932200104X.

[29] A.H. Larsen, J.J. Mortensen, J. Blomqvist, I.E. Castelli, R. Christensen, M. Dułak, J. Friis, M.N. Groves, B. Hammer, C. Hargus, E.D. Hermes, P.C. Jennings, P.B. Jensen, J. Kermode, J.R. Kitchin, E.L. Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J.B. Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. Schütt, M. Strange, K.S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, K.W. Jacobsen, J. Phys. Condens. Matter 29 (27) (2017) 273002,

https://doi.org/10.1088/1361-648X/aa680e, https://iopscience.iop.org/article/10.1088/1361-648X/aa680e.
[30] A. Laio, M. Parrinello, Proc. Natl. Acad. Sci. 99 (20) (2002) 12562–12566, https://doi.org/10.1073/pnas.202427399, https://www.pnas.org/doi/abs/10.1073/pnas.202427399.
[31] B. Shahriari, K. Swersky, Z. Wang, R.P. Adams, N. de Freitas, Proc. IEEE 104 (1) (2016) 148–175, https://doi.org/10.1109/JPROC.2015.2494218, https://ieeexplore.ieee.org/document/7352306.
[32] A. Deshwal, C.M. Simon, J.R. Doppa, Mol. Syst. Des. Eng. 6 (2021) 1066–1086, https://doi.org/10.1039/D1ME00093D, https://pubs.rsc.org/en/content/articlelanding/2021/me/d1me00093d/.
[33] S. Goedecker, J. Chem. Phys. 120 (21) (2004) 9911–9917, https://doi.org/10.1063/1.1724816, https://aip.scitation.org/doi/10.1063/1.1724816.
[34] E. Vignola, S.N. Steinmann, K. Le Mapihan, B.D. Vandegehuchte, D. Curulla, P. Sautet, J. Phys. Chem. C 122 (27) (2018) 15456–15463, https://doi.org/10.1021/acs.jpcc.8b04108, https://pubs.acs.org/doi/10.1021/acs.jpcc.8b04108.
[35] M. Todorović, M.U. Gutmann, J. Corander, P. Rinke, npj Comput. Mater. 5 (1) (2019) 35, https://doi.org/10.1038/s41524-019-0175-2, https://www.nature.com/articles/s41524-019-0175-2.
[36] J. Rey, P. Clabaut, R. Réocreux, S.N. Steinmann, C. Michel, J. Phys. Chem. C 126 (17) (2022) 7446–7455, https://doi.org/10.1021/acs.jpcc.2c00998, https://pubs.acs.org/doi/full/10.1021/acs.jpcc.2c00998.
[37] H. Jung, L. Sauerland, S. Stocker, K. Reuter, J.T. Margraf, Machine-learning driven global optimization of surface adsorbate geometries, https://chemrxiv.org/engage/chemrxiv/article-details/632dac64114b7e384b1aa1d2, 2022.
[38] S.P. Ong, W.D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V.L. Chevrier, K.A. Persson, G. Ceder, Comput. Mater. Sci. 68 (2013) 314–319, https://doi.org/10.1016/j.commatsci.2012.10.028, https://www.sciencedirect.com/science/Article/pii/S0927025612006295.
[39] P. Giannozzi, S. Baroni, N. Bonini, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, G.L. Chiarotti, M. Cococcioni, I. Dabo, A.D. Corso, S. de Gironcoli, S. Fabris, G. Fratesi, R. Gebauer, U. Gerstmann, C. Gougoussis, A. Kokalj, M. Lazzeri, L. Martin-Samos, N. Marzari, F. Mauri, R. Mazzarello, S. Paolini, A. Pasquarello, L. Paulatto, C. Sbraccia, S. Scandolo, G. Sclauzero, A.P. Seitsonen, A. Smogunov, P. Umari, R.M. Wentzcovitch, J. Phys. Condens. Matter 21 (2009) 395502, https://doi.org/10.1088/0953-8984/21/39/395502, https://iopscience.iop.org/article/10.1088/0953-8984/21/39/395502.
[40] P. Giannozzi, O. Andreussi, T. Brumme, O. Bunau, M.B. Nardelli, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, M. Cococcioni, N. Colonna, I. Carnimeo, A.D. Corso, S. de Gironcoli, P. Delugas, R.A. DiStasio, A. Ferretti, A. Floris, G. Fratesi, G. Fugallo, R. Gebauer, U. Gerstmann, F. Giustino, T. Gorni, J. Jia, M. Kawamura, H.-Y. Ko, A. Kokalj, E. Küçükbenli, M. Lazzeri, M. Marsili, N. Marzari, F. Mauri, N.L. Nguyen, H.-V. Nguyen, A.O. de-la Roza, L. Paulatto, S. Poncé, D. Rocca, R. Sabatini, B. Santra, M. Schlipf, A.P. Seitsonen, A. Smogunov, I. Timrov, T. Thonhauser, P. Umari, N. Vast, X. Wu, S. Baroni, J. Phys. Condens. Matter 29 (46) (2017) 465901, https://doi.org/10.1088/1361-648X/aa8f79, https://iopscience.iop.org/article/10.1088/1361-648X/aa8f79.
[41] P. Giannozzi, O. Baseggio, P. Bonfà, D. Brunato, R. Car, I. Carnimeo, C. Cavazzoni, S. de Gironcoli, P. Delugas, F. Ferrari Ruffino, A. Ferretti, N. Marzari, I. Timrov, A. Urru, S. Baroni, J. Chem. Phys. 152 (15) (2020) 154105, https://doi.org/10.1063/5.0005082, https://aip.scitation.org/doi/10.1063/5.0005082.
[42] B. Delaunay, Izv. Akad. Nauk SSSR, Otd. Mat. Estestv. Nauk 6 (1934) 793–800, https://www.mathnet.ru/eng/im4937.
[43] National Center for Biotechnology Information, PubChem Compound Summary for CID 11597, 1-Hexene, retrieved April 18, 2023, https://pubchem.ncbi.nlm.nih.gov/compound/1-Hexene, 2023.
[44] S. Grimme, J. Comput. Chem. 27 (15) (2006) 1787–1799, https://doi.org/10.1002/jcc.20495, https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.20495.