SuSMoST: Surface Science Modeling and Simulation Toolkit

Sergey S. Akimenko, Galina D. Anisimova, Anastasiya I. Fadeeva, Vasiliy F. Fefelov, Vitaliy A. Gorbunov, Tatiana R. Kaumova, Alexander V. Myshlyavtsev, Marta D. Myshlyavtseva, and Pavel V. Stishenko*

Department of Chemical Engineering, Omsk State Technocal University, Omsk

E-mail: PavelStishenko@yandex.ru

Abstract

We offer the scientific community the Surface Science Modelling and Simulation Toolkit (SuSMoST), which includes a number of utilities and implementations of statistical physics algorithms and models. With SuSMoST one is able to predict or explain the structure and thermodynamics properties of adsorption layers. SuSMoST automatically builds formal graph and tensor-network models from atomic description of adsorption complexes. So it can be routinely used for a wide class of adsorption systems. SuSMoST aids *ab initio* calculations of interactions between adsorbed species. In particular it generates surface samples considering symmetry of adsorption complexes. Using methods of various nature SuSMoST generates representative samples of adsorption layers and computes its thermodynamics quantities such as mean energy, coverage, density, heat capacity. From these data one can plot phase diagrams of adsorption systems, assess thermal stability of self-assembled structures, simulate thermal desorption spectra, *etc.*

Introduction

Self-assembly of molecular monolayers on solid surfaces is a perspective approach for commercial scale production of nanoscale devices and new materials. Advances of experimental techniques, in particular atomic resolution microscopy, revealed a great variety of regular structures made up of adsorbed molecules.[1] The next step should be the rational design of self-assembly processes of structures with the desired properties. This requires theoretical tools - models, algorithms and computer programs that are able to *in silico* predict self-assembly processes and properties of molecular structures on surfaces. Just as molecular modeling has provided invaluable support to the drug design and pharmaceutical industry, it can help to develop new materials and nanoscale devices.

Phase diagram is the key characteristic of adsorption system. It breaks the phase space of macroscopic parameters (temperature, partial pressures, concentrations) in areas of existence of ordered structures and other phases. What structures can be obtained using these molecules? What structures will arise when replacing one surface to another? Will this structure be stable at room temperature? These questions are principal for inventing of the desired adsorption system. Answers to them can be obtained using the methods of thermodynamics and statistical physics.

Simulation of adsorption systems on molecular level is a hard task. Existence and stability of adsorption layer structures are determined by diffusion, adsorption and desorption processes. The time scale of these processes varies significantly and exceeds typical time integration step of classical molecular dynamics. Therefore it is hard to get representative samples of such systems, even with advanced sampling techniques.[2]

Moreover, up today there is no affordable way to compute energy of adsorption systems. Density functional theory is too computationally demanding for representative sampling of surface slabs with adsorbates. Creation of empirical potential for this purpose is hindered by variety of chemical elements included in interesting adsorption systems and by significant surface effect on geometry of adsorbed molecules and interactions between them.

Fortunately, quite often it is possible to distinguish a few ways of adsorption for each adsorbate molecule. IUPAC defines such a structure ”constituted by the adsorbed and the part of the adsorbent to which it is bound” as *adsorption complex* (AC).[3]

Sites on surface, where ACs may form are called *adsorption sites.* In many cases, surface geometry changes induced by adsorption layer can be neglected. Hence, we can assume that active sites constitute a lattice. ACs that can form on active sites can be considered as states of the corresponding lattice nodes. Therefore, the lattice with the set of possible site states constitute the model of the adsorption system. There are lattice models of various degrees of abstraction: Langmuir adsorption model, hard disks,[4,5] dimers,[6–9] k-mers,[10,11] binary gasses,[12–16] molecules of various symmetry,[17–20] tricarboxylic[21–24] acids, porphyrins,[25] *etc.*[26,27]

In spite of numerous successful applications of lattice models, this kind of simulations is still not a daily tool for surface science researchers. By this they are dramatically different from *ab intio* and molecular dynamics methods that are routinely used by a wide range of scientists. One of the key reasons of this popularity is a relative simplicity and universality of task formulation for these methods. Enumeration of atomic coordinates and, in some cases, velocities and covalent bonds is a straightforward and universal way to describe the system of interest. But most of computer codes that are meant for computations within lattice model framework require explicit enumeration of lattice site states and interactions between them.[28–32] For routine simulation of realistic adsorption systems this results in a lot of error-prone work.

Moreover, solving of tasks within lattice models also can be non-trivial. Even for the simplest models it could be difficult to predict stable phase or critical values of parameters.[4,8] To ensure validity and reliability of results simultaneous usage of various complicated algorithms could be necessary. There are several Kinetic Monte Carlo codes[28–32] of general purpose, but implementations of other methods, to the best of our knowledge, are tailored to specific kinds of lattice physical systems, different from adsorption layers.[33–35] Although kinetic Monte Carlo in many cases is able to predict equilibrium phase of adsorption layer, it is not the most suitable method for this kind of tasks. Probably this is the reason why contemporary computational studies of surface self-assembly do not refer to available codes, but apparently use in-house state of art codes.[10,11,18,22]

Here we offer the software toolkit SuSMoST that aids building and parameterization of lattice adsorption models and is able to conduct computations within these models with several algorithms of various nature.

Further in section **Features and algorithms** we describe in details SuSMoST features and what can be done with it. In section **Model** we describe mathematical model of adsorption that is used by SuSMoST. In section **Basic Workflow** the suggested algorithm of a computational study with SuSMoST is described with examples and recommendations.

Features and algorithms

SuSMoST is constituted by two Python packages in WHL[36] format - main and Monte Carlo, documentation and several exemplar scripts available at `susmost.com`. In addition to standard Python libraries SuSMoST requires `NumPy`, `SciPy`, `ASE` and `spglib` libraries. We recommend to install SuSMoST in isolated Python environment created by `virtualenv` command. Currently SuSMoST packages are available for Python distribution version 3.6 and 3.7. Monte Carlo package also requires OpenMPI version 2.1 library to be installed.

In further subsections SuSMoST features are grouped by tasks it automates or assists in doing:

1. Parameterization of adsorption system models by atomic description.

2. Computation of adsorption system properties through the model analysis.

SuSMoST cornerstone is the generalized lattice model of adsorption introduced in Ref. 37. It is described in details in section **Model**. The model is general enough to describe adsorption systems with arbitrary finite number of ACs of any symmetry, with any pairwise interactions, including directional and long-range ones. Manybody interactions also can be taken into account in some SuSMoST algorithms on an ad hoc basis. We created SuSMoST keeping in mind self-assembling adsorption layers, monolayers primarily. Yet the model is no less able to describe and investigate multilayer adsorption and magnetic systems.[15,16]

Both parameterization of the model and computations with its framework are complex tasks. SuSMoST provides functionality that help to solve them. Further in sections **Model parameterization** and **Model analysis** we describe SuSMoST capabilities that are useful on corresponding stages of a computational study.

Model parameterization

In principle, it is possible to manually parameterize the generalized model for any specific adsorption system, but in many cases it is laborious and error prone approach. SuSMoST automates this process significantly reducing amount of work and error risk. User's manual work could be necessary only in two steps:

1. Creation of AC sketches in Extended XYZ format..
2. Computation of energies of surface samples with one or two ACs or explicit specification of lateral interaction energies.

AC sketch is an atomic structure constituted by surface slab with an adsorbate specified as Extended XYZ file. User is expected to specify sketches for all ACs that are relevant for considered adsorption system with a sketch of empty surface slab. Examples of AC sketches and detailed requirements to them are described below in section **Basic workflow**. SuSMoST analyze symmetry of all ACs and empty surface by their sketches using library spglib.[38] Symmetry detection tolerance is an adjustable parameter. Based on symmetry data SuSMoST generates all possible unique distances and orientations of ACs relative to each other and to radius vector. For each mutual position a surface sample with a pair of ACs is generated. Additionally samples with a single AC on surface slabs of various sizes are generated. These samples are also generated as Extended XYZ files.

These samples can be used to compute ACs interaction energies either with DFT or with empirical potentials. SuSMoST includes exemplar script `test_energies.py` (also included in Supporting Information) with codes that use GPAW DFT library[39–41] and empirical potential for CO/Pt(111) system.[26] Suggested algorithm of interaction energies computing is described in details below in section **Basic workflow**.

Then using the AC symmetry data and lateral interaction energies SuSMoST parametrize the generalized lattice model in automated manner. The model can be represented in two ways described in section **Model**: as a graph[37] or as an interaction-round-a-face tensor.[42]

Model analysis

SuSMoST implements a set of statistical physics algorithm: Metropolis Monte Carlo, Replica Exchange (Parallel Tempering), Wang-Landau, transfer-matrix (TM) and tensor renormalization group (TRG). They are radically different in accuracy, performance and computational approach. Their areas of applicability and sets of output characteristics overlap, but only partially. The use of several dissimilar algorithms for the study of a single system gives a complete picture of its properties and ascertains the reliability of the results. In case of a limited computing power, an opportunity to choose the most efficient algorithm could be decisive for research project.

Algorithms implemented in SuSMoST can be divided into two groups: Monte Carlo based and tensor network based methods. Methods of the first group are the classical Metropolis Monte Carlo, replica exchange (parallel tempering) and Wang-Landau algorithms. Methods of the second group are: transfer-matrix and tensor renormalization group algorithms.

Generally with this set of algorithms one can immediately get the following data:

1. Equilibrium samples of adsorption layer in Extended XYZ format.

2. Equilibrium samples of key parameters of the adsorption layer: energy, density, coverage, fractional coverage.

3. The numerically exact averages and moments of these parameters, of the logarithm of partition function, pair correlation coefficients of neighboring site states.

Analyzing these data one can get adsorption isotherms and isobars, phase diagrams, order parameters and heat capacity of the adsorption layer, thermal desorption spectra. From surface sample one can get a lot of information about adsorption layer structure and properties.

Detailed description of characteristics, required input data and limitations of implementations of algorithms in SuSMoST is provided in section **Basic workflow**.

Model

Treatment of a single system with various algorithms is possible in SuSMoST due to a solid base that is a universal lattice model of adsorption layer.[37] SuSMoST uniformly treats any crystal lattice. Specific lattice structure is automatically selected based on the symmetry of clean adsorbate surface. Lattice sites are mapped to unit cells of a crystal surface – see Figure 1. The set of lattice site states is comprised by all possible orientations of all ACs of the considered adsorption system.

The model Hamiltonian has the following form:

$$H = \sum_{i \in L} \mu(\sigma_i) + \sum_{i,j \in L} \lambda(\sigma_i, \sigma_j, \mathbf{r}_{ij}) \tag{1}$$

Here L is a set of lattice sites, σ_i is a state of i-th site, $\mu(\sigma)$ is an energy of state σ, $\mathbf{r}_{ij}$ is a radius-vector from i-th site to j-th site, λ is a function of lateral interactions that depends on the states and the radius-vector.

Physical meaning of state energy $\mu(\sigma)$ is the energy change caused by formation of the

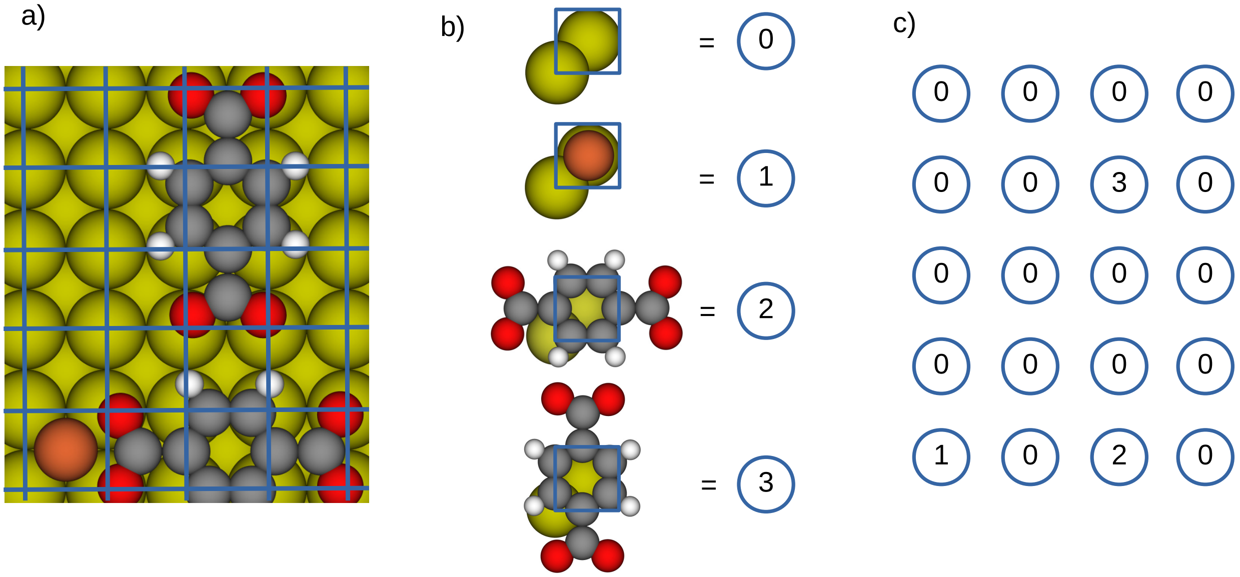


Figure 1: Lattice model of adsorption layer. a) Adsorption layer with terephthalic acid molecules and a copper atom adsorbed on Cu(100) surface. b) Mapping of adsorption complexes on lattice site states. c) Lattice model of the adsorption layer.

corresponding AC on a clean surface, for example on adsorption of a molecule from a gas phase. It can be computed as a difference between free energies of the system when the molecule is adsorbed and when it is desorbed:

$$\mu(\sigma) = G_{\sigma\text{-}ads} - (G_{empty} + G_{\sigma\text{-}des}) \qquad (2)$$

Here $G_{\sigma\text{-}ads}$ is a free energy of surface with adsorbed molecule, G_{empty} is a free energy of clean surface, $G_{\sigma\text{-}des}$ is a free energy of molecule in gas phase. So $\mu(\sigma)$ may depend on enthalpy of AC formation, gas pressure, vibrational and zero-point energies. It is important that μ does not include lateral interactions, in other words it describes an energy change in zero coverage limit. To be short further we use term *adsorption energy* to refer to μ.

In practice, in order to carry out the calculations, it is necessary to choose a formulation of λ that would be convenient for computation and parameterization. SuSMoST implements and uses two mathematical formulations of lattice models of adsorption layers. Further we will call them graph model representation and tensor network model representation. The graph representation is used by Monte Carlo based algorithms. TRG and TM algorithms rely on a tensor network representation.

Graph model representation

Graph model representation is able to describe adsorption systems with arbitrary number of ACs of any symmetry, taking into account pairwise long-range and directed lateral interactions. Mathematically rigorous description of the model in this representation is provided in Ref. 37. Here we are giving a less formal introduction sufficient for understanding of SuSMoST capabilities.

In graph model the lattice is represented as a graph with nodes matched to lattice sites. Graph edges connect only such pairs of nodes ij that there exists at least one pair of site states σ_i and σ_j, such that $\lambda(\sigma_i, \sigma_j, \mathbf{r}_{ij}) \neq 0$, *i.e.* $\exists\ \sigma_i, \sigma_j : \lambda(\sigma_i, \sigma_j, \mathbf{r}_{ij}) \neq 0$. In other words the graph edges connect only those lattice site that in principle could interact with each other with the given Hamiltonian. In fact, presence or absence of edges is analogous to usage of cut off radius in continuous models. For a surface with a crystalline lattice the set of possible radius vectors $\mathbf{r}_{ij}$ is discrete due to transnational symmetry, hence it could be enumerated. It is reasonable to assume that lateral interactions radius is finite, so the set of possible $\mathbf{r}_{ij}$ values also should be finite for any considered graph. Further we denote this set as E and use the term *edge dictionary* for it. Each graph edge corresponds to a single edge type from the dictionary E. Similarly, the set of possible lattice site states $\{\sigma_i\}$ is also finite.

Therefore the function λ can be specified as a three-dimensional array with elements corresponding to all combinations of $(\sigma_i, \sigma_j, \mathbf{r}_{ij})$. The size of this array quadratically depends on the number of lattice site states an linearly on the size of the edge dictionary E. It is important to note that the size of the edge dictionary E grows quadratically with cut off radius of lateral interactions. These dependencies should be taken into account when estimating memory requirements of simulations with SuSMoST.

Tensor model representation

Another way to specify the lattice model Hamiltonian in SuSMoST is the interaction-round-a-face (IRF) model Ref.[42,43] This approach is based on representation of the Hamiltonian as a sum of interactions of all quads of surface unit cell (plaquettes) with a common vertex:

$$H = \sum_{<ijkl>} T_{ijkl} \tag{3}$$

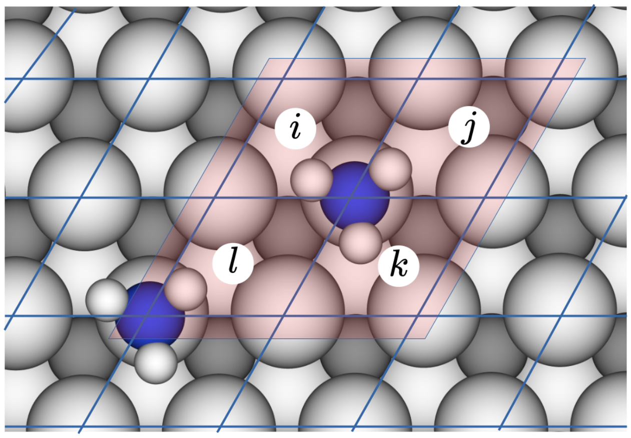


Figure 2: A quad of surface unit cell (plaquettes) with a common vertex, used in interaction-round-a-face (IRF) model.

Here i, j, k and l are the indices of the adjacent surface unit cells, T is an interactions tensor. An example of the quad of interacting cells is show in Figure 2. For the given mutual location of indices elements of tensor T can be computed from adsorption energies and function λ with the following formula:

$$T_{ijkl} = \frac{1}{4}E_{site}(i,j,k,l) + \frac{1}{2}E_{side}(i,j,k,l) + E_{diag}(i,j,k,l) \tag{4}$$

where

$$E_{site}(i,j,k,l) = \mu(\sigma_i) + \mu(\sigma_j) + \mu(\sigma_k) + \mu(\sigma_l) \tag{5}$$

$$E_{side}(i,j,k,l) = \lambda(\sigma_i,\sigma_j,\mathbf{r}_{ij}) + \lambda(\sigma_j,\sigma_k,\mathbf{r}_{jk}) + \lambda(\sigma_k,\sigma_l,\mathbf{r}_{kl}) + \lambda(\sigma_l,\sigma_i,\mathbf{r}_{li}) \tag{6}$$

$$E_{diag}(i,j,k,l) = \lambda(\sigma_i,\sigma_k,\mathbf{r}_{ik}) + \lambda(\sigma_j,\sigma_l,\mathbf{r}_{jl}) \tag{7}$$

Here E_{site} is an internal energy of the cells, E_{side} is an energy of interactions between cells with a common edge, E_{diag} is an energy of interactions between cell that do not have common edges. SuSMoST is able to compute the tensor T by formulae $4-7$, but users can change it in any way. In particular one can add corrections for many body interactions.

It is important that tensor based representation is able to take into account only interactions between cells that have at least one common vertex. Cells that have no common vertices do not get into any quad enumerated in formula 3. If long range interactions are exists between such cells, then it is necessary to join several cells of the original model into larger supercells, for example of size 2×2. The new model made of such supercells is able to take into account interactions that are twice further. SuSMoST can do such model reformulation automatically. But number of lattice site states grows quickly with supercell size. In case if there are no prohibitive interactions between ACs this growth is exponential. In it's turn the size of tensor T is equal to fourth power of number of lattice site states. So it is easy to hit the memory limit when dealing with systems with long range interactions with such model representation.

This model representation is used by TM and TRG algorithms

Basic workflow

We developed SuSMoST keeping in mind the workflow that formed in our group over the years of computational studies of adsorption layers with in-house SuSMoST predecessors[6–8,13–15,26,27,37,44–48] (see Scheme 1). In this section we describe practical aspects of working with SuSMoST using this workflow as an outline. But it should not be considered as the only way to do research with SuSMoST. Being modular and scriptable, with simple formats of input and output data SuSMoST and its parts can be combined with other codes and various experimental techniques.

Making sketches of adsorption complexes

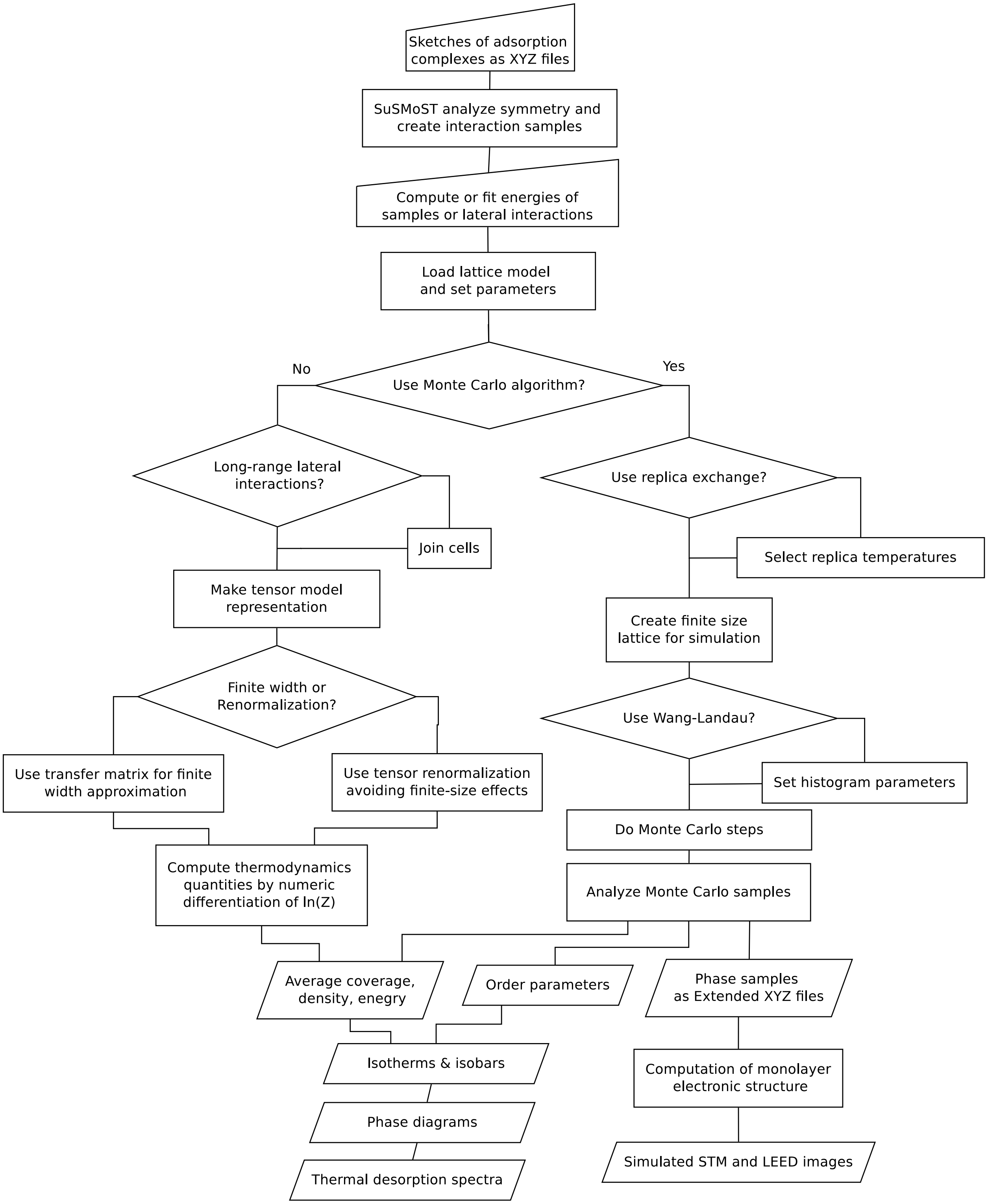
The set of adsorption complexes that exist in the considered adsorption system is the very first component of its model. One can get structure of adsorption complexes from microscopy images, from DFT calculations or by making a hypothesis from IR spectra. At first stages of modelling an accurate geometry of adsorption complexes is not necessary. Approximate descriptions of the complexes are sufficient to analyze their symmetry. Further we use the term *sketch* to underline low accuracy requirements at this stage.

Sketches of adsroption complexes must be provided to SuSMoST as files in Extended XYZ format. Extended XYZ format adds arbitrary key-value pairs to conventional XYZ data. SuSMoST expects the following keys for both adsorption complex and empty surface slab sketches:

1. `Lattice` - cell basis vectors; this parameter is usual for finite-size systems.
2. `size` - two-dimensional size of the slab in unit cells, for example `size = "3 3"` should be specified if the sketch contains 3 by 3 unit cells of slab.

In addition, extended XYZ files of adsorption complexes must provide information for distinguishing atoms of surface and adsorbed molecule (adsorbate). There are several keys that can be used for it:

1. `surface_zlevel` - value of z-coordinate that separates adsorbate atoms (above `surface_zlevel`) and adsorbent atoms (bellow `surface_zlevel`).


Sketches of adsorption complexes as XYZ files
SuSMoST analyze symmetry and create interaction samples
Compute or fit energies of samples or lateral interactions
Load lattice model and set parameters
No
Yes
Use Monte Carlo algorithm?
Long-range lateral interactions?
Join cells
Make tensor model representation
Finite width or Renormalization?
Use transfer matrix for finite width approximation
Use tensor renormalization avoiding finite-size effects
Compute thermodynamics quantities by numeric differentiation of ln(Z)
Use replica exchange?
Select replica temperatures
Create finite size lattice for simulation
Use Wang-Landau?
Set histogram parameters
Do Monte Carlo steps
Analyze Monte Carlo samples
Average coverage, density, enegry
Order parameters
Phase samples as Extended XYZ files
Isotherms & isobars
Phase diagrams
Thermal desorption spectra
Computation of monolayer electronic structure
Simulated STM and LEED images

Scheme 1: Suggested workflow for an adsorption system study with SuSMoST

2. `adsorbate_indices` - list of integers that enumerate indices of adsorbate atoms.

3. `adsorbate_elements` - list of chemical symbols of adsorbate atoms. It can be used only if sets of adsorbate and adsorbent elements do not intersect.

For example the Figure 3 shows the sketch of adsorption complex constituted by CO molecule adsorbed on a bridge site on Pt(111) surface. The figure 4 shows an excerpt from the corresponding Extended XYZ file. Slab size is specified in file header as a key-value pair `size="3 3"`. Supercell basis vectors are specified by the `Lattice` key: $x_1 = (8.315576, 0, 0)$, $x_2 = (4.157788, 7.201500, 0)$, $x_3 = (0, 0, 26.789639)$. The key-value pair `pbc="T T T"` imposes periodic boundary conditions in all three directions. Since the surface includes neither carbon nor oxygen atoms, it is possible to use element names to distinguish adsorbate and adsorbent atoms, hence the `adsorbate_elements` key is used here.

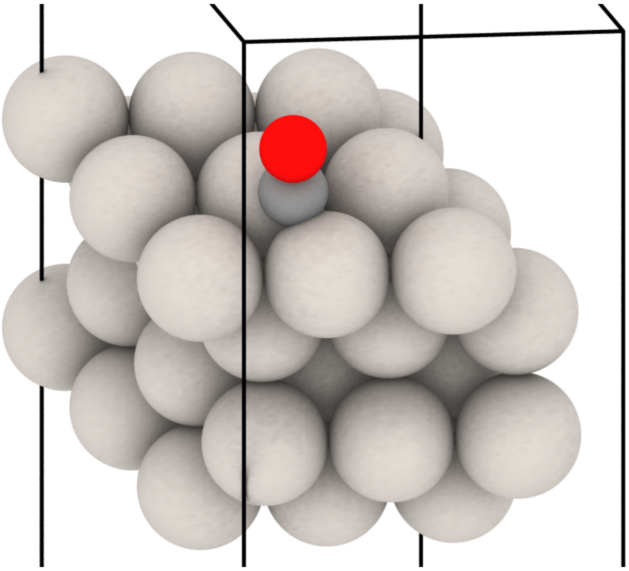
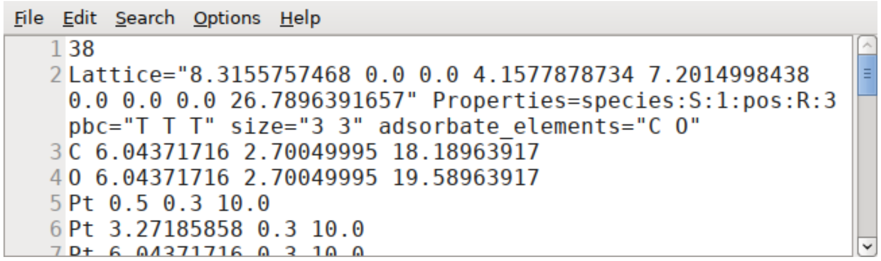


Figure 3: Sketch of the adsorption complex of CO on Pt(111) bridge site. Black lines show cell borders.

SuSMoST utilizes the AC sketches for analysis of AC symmetries and for generation of adsorption layer samples. Accurate geometrical optimization is not necessary for these purposes.



```
38
Lattice="8.3155757468 0.0 0.0 4.1577878734 7.2014998438
0.0 0.0 0.0 26.7896391657" Properties=species:S:1:pos:R:3
pbc="T T T" size="3 3" adsorbate_elements="C O"
C 6.04371716 2.70049995 18.18963917
O 6.04371716 2.70049995 19.58963917
Pt 0.5 0.3 10.0
Pt 3.27185858 0.3 10.0
Pt 6.04371716 0.3 10.0
```

Figure 4: Excerpt from the extended XYZ file with a sketch of the adsorption complex of CO on Pt(111) bridge site. The `Lattice` and `Properties` keys are common for the Extended XYZ format.[49] Key-value pairs `size="3 3"` and `adsorbate_elements="C O"` are used by SuSMoST.

Generating surface samples

For creation of surface samples SuSMoST provides `make_ac_samples()` function and helper script `make-ac-samples`. This script wraps the function call for convenient usage from command line. The script (and the function) takes as argument a list of files with sketches of empty surface and adsorption complexes. An option `--r-cut-off` specifies the distance in angstroms beyond which lateral interactions between any adsorption complexes are negligible. It affects the number of generated samples of pairwise interactions. Options `--x-border` and `--y-border` controls slab sizes of surface samples both with one and two adsorption complexes. They are meant to control or eliminate lateral interactions between periodic images if periodic boundary conditions are implied in calculations. Default value of this option is derived from `--r-cut-off` radius. Also there are several options that control symmetry analysis of surface and adsorption complexes. All options are described in details in SuSMoST documentation in API reference section.

The script `make-ac-samples` creates a directory, named `tmp` by default, with the following files:

1. `sample_empty_unit_cell.xyz` contains a unit cell of the clean surface. Using this sample one can compute an energy of empty adsorbent. For example see Figure 5a.

2. `adsorbate_state_`n`.xyz`, where n is an index of a site state in generated lattice model. Each of these files correspond to some lattice site state and contains only adsorbate

atoms of the corresponding state. For example see Figures 5b-5d. There are at least two such files, and hence at least two states of lattice sites - one for an empty site and one or more for each adsorption complex. If adsorption complex symmetry is the same as the symmetry of surface, than only one site state will be generated for it. But for adsorption complexes which poses less symmetry than the surface there will be several states - one state for each orientation of the adsorption complex. For example, CO on atop site of Pt(111) has the same symmetry as the clean Pt(111) surface. In other words, rotation of the surface around this adsorption complex do not change its orientation. Hence only one site state is necessary to represent this adsorption complex within lattice model. Contrary, CO adsorbed on a bridge site is less symmetric than the empty surface. It means that some rotations transform an empty surface in original state, but change the orientation of the adsorption complex. In particular the (100) facet of a face-centered cubic crystal is invariant to 90° rotation. But the same rotation transforms vertical bridge sites to horizontal ones and vice-versa. Hence at least two states of a lattice site are necessary to represent the adsorption complexes of CO on bridge sites of FCC(100) face. One state is required for vertical bridge complex and another for horizontal one. In the case of FCC(111) facet there are three possible orientations of bridge sites.. Therefore, SuSMoST generates three states for CO adsorbed on bridge sites of Pt(111) surface - see Figures 5b-5d for example.

SuSMoST utilizes these files for generation of surface samples during Monte Carlo runs. Also it is convenient to use these files for checking whether SuSMoST has properly detected all symmetries of adsorption complexes. If too many states have been generated for some adsorption complex then symmetry detection tolerance should be increased. There are arguments of `make-ac-samples` script and parameters of `make_ac_samples()` function that controls it.

3. `sample_ac_`*AC-name_index*`.xyz` files contain samples of surface with single adsorption complex with different sizes of surface slab. Here *AC-name* is the name of adsorption

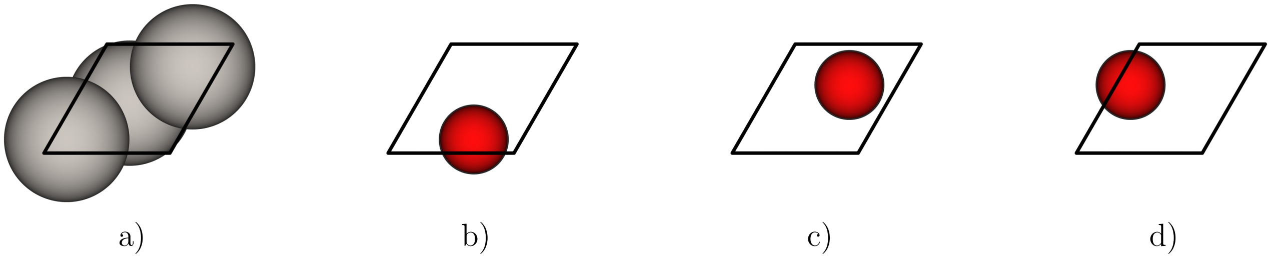


Figure 5: **a)** Empty state of Pt(111) surface unit from file named `sample_empty_unit_cell.xyz`. **b-d)** Three lattice site states generated from the CO-on-bridge adsorption complex by symmetric transformations of Pt(111) surface. These structures are stored in files named `adsorbate_state_1.xyz`, `adsorbate_state_2.xyz` and `adsorbate_state_3.xyz`.

complex and *index* is an integer. For example, a file named `sample_ac_nh3-atop-v3c2_00.xyz` contains surface slab of size (1×1) unit cells with one NH_3 molecule adsorbed on atop site. Then a file named `sample_ac_nh3-atop-v3c2_01.xyz` contains the same adsorption complex but with surface slab of size (2×2) unit cells. For example see Figure 6. Maximum slab size is limited by parameters `--r-cut-off`, `--x-border` and `--y-border`.

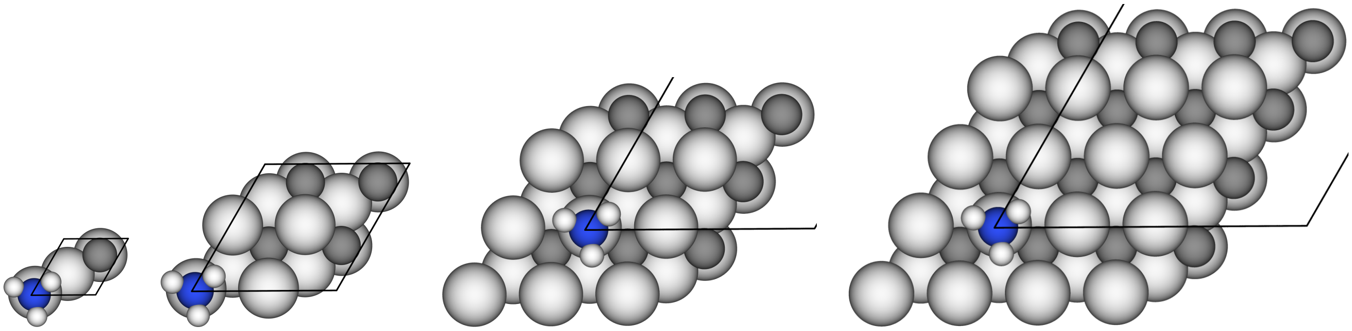


Figure 6: Samples with a single complex of NH_3 adsorbed on atop site of V_3C_2 surface. Slab sizes from left to right: 1×1, 2×2, 3×3 and 4×4. Unit cell with adsorption complex is placed in origin. Periodic boundary conditions are implied.

These files are primarily meant for computation of adsorption energies, *i.e.* μ parameters of the Hamiltonian Eq. 1. Sequence of samples can be used to check if μ converges with slab size. Lateral interactions between periodic images of an adsorption complex are supposed to be negligible if the slab is large enough.

4. sample_*index*.xyz files are the samples of couples of adsorption complexes placed on various distances from each other and with different mutual orientation. These files are meant for computation or manual specification of lateral interactions energies. If borders of the slab are large enough, then lateral interactions of adsorption complexes with periodic images is negligible and extraction of the lateral interaction energies is straightforward and can be done by SuSMoST automatically.

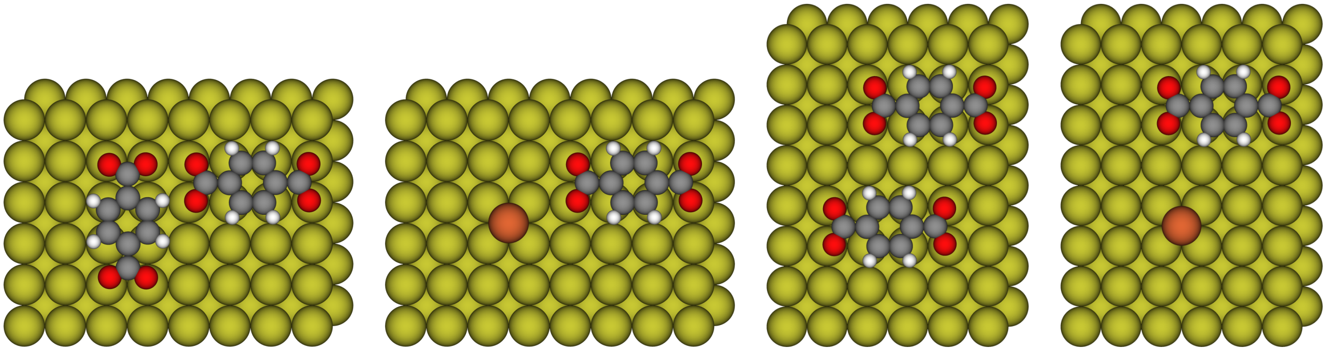


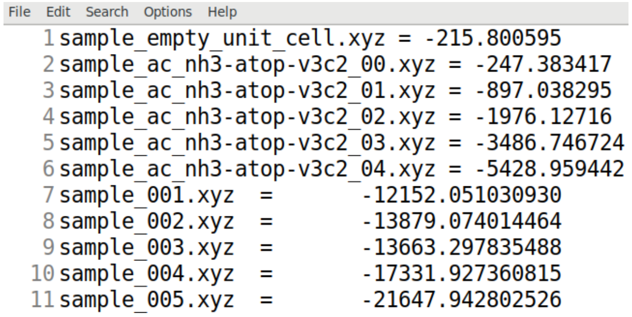
Figure 7: Samples with pairs of adsorption complexes constituted by terephthalic acid and iron adatom on Cu(100) surface.

5. interactions file contains matrices of lateral interactions between lattice sites of different states separated by different radius vectors. Elements of these matrices are not the energies, but integer indices of corresponding files sample_*index*.xyz.

Energies computation and specification

Despite of impressive progress in computational chemistry, accurate parameterization of Hamiltonian Eq. 1 is still a challenging task. Limits of computing power, long-range lateral interactions[1,26,50,51] and peculiarities of some adsorption complexes[52] impede universal approach to computation of driving forces of on-surface processes. Samples generated by SuSMoST can be used to get adsorption and lateral interaction energies using various methods, approximations and codes. But particular way to compute or fit the energies of the samples have to be chosen by a researcher.

Energies of the samples must be specified by user in the `energies` file placed into the same directory. The `energies` file must contain rows of the format `<sample-file-name> = <sample energy-value>`. The expected format of the `energies` file is illustrated in Figure 8.



```
sample_empty_unit_cell.xyz = -215.800595
sample_ac_nh3-atop-v3c2_00.xyz = -247.383417
sample_ac_nh3-atop-v3c2_01.xyz = -897.038295
sample_ac_nh3-atop-v3c2_02.xyz = -1976.12716
sample_ac_nh3-atop-v3c2_03.xyz = -3486.746724
sample_ac_nh3-atop-v3c2_04.xyz = -5428.959442
sample_001.xyz  =       -12152.051030930
sample_002.xyz  =       -13879.074014464
sample_003.xyz  =       -13663.297835488
sample_004.xyz  =       -17331.927360815
sample_005.xyz  =       -21647.942802526
```

Figure 8: Excerpt from `energies` file filled for NH_3/V_3C_2 adsorption system.

SuSMoST computes lateral interactions for Hamiltonian Eq. 1 from samples energies with the following formulae:

$$\lambda(x, y, \mathbf{r}) = J(x, y, \mathbf{r}) - J(x) - J(y) \tag{8}$$

where

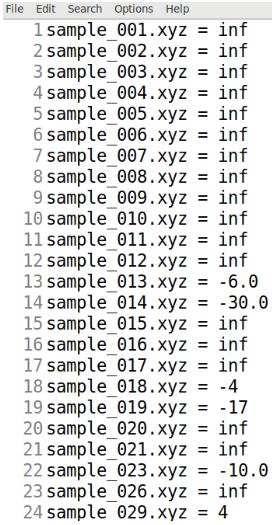
$$J(x, y, \mathbf{r}) = G(x, y, \mathbf{r}) - S(x, y, \mathbf{r})G_{empty} \tag{9}$$

and

$$J(x) = G(x) - S(x)G_{empty} \tag{10}$$

Here x, y are indices of lattice cell states; $\mathbf{r}$ is a radius vector between the cells; $G(x)$ is an energy of the sample with a single isolated adsorption complex that generates the state x (files `sample_ac_`*AC-name_index*`.xyz`); $G(x, y, \mathbf{r})$ - energy of the sample with a couple of adsorption complexes that generate states x and y, separated by vector $\mathbf{r}$ (files `sample_`*index*`.xyz`); $S(x)$ and $S(x, y, \mathbf{r})$ are numbers of unit cells in corresponding samples; G_{empty} is an energy of an empty surface unit cell(file `sample_empty_unit_cell.xyz`); $J(x)$ and $J(x, y, \mathbf{r})$ are energies of an isolated adsorption complex and of a couple of adsorption complexes excluding contributions of adsorbent slabs.

In case if you have a potential of lateral interactions in explicit form or if you are going to fit it to experimental data, it is convenient to omit entries for `sample_ac_`*AC-name_index*`.xyz` and `sample_empty_unit_cell.xyz` files. Then zero values will be used for G_{empty} and $G(x)$. Hence $J(x) = 0$ and $\lambda(x, y, \mathbf{r}) = J(x, y, \mathbf{r}) = G(x, y, \mathbf{r})$. Figure 9 shows an example of such case.



```
sample_001.xyz = inf
sample_002.xyz = inf
sample_003.xyz = inf
sample_004.xyz = inf
sample_005.xyz = inf
sample_006.xyz = inf
sample_007.xyz = inf
sample_008.xyz = inf
sample_009.xyz = inf
sample_010.xyz = inf
sample_011.xyz = inf
sample_012.xyz = inf
sample_013.xyz = -6.0
sample_014.xyz = -30.0
sample_015.xyz = inf
sample_016.xyz = inf
sample_017.xyz = inf
sample_018.xyz = -4
sample_019.xyz = -17
sample_020.xyz = inf
sample_021.xyz = inf
sample_023.xyz = -10.0
sample_026.xyz = inf
sample_029.xyz = 4
```

Figure 9: Excerpt from `energies` file with explicit energies of lateral interactions. `inf` means infinitely large energy which implies that the corresponding configuration is prohibited.

Explicit separation of adsorption energies μ from lateral interactions λ in Eqn. 1 allows us to break up the model parameterization task on clear independent subtasks. Although computation of energies of all samples with DFT may be computationally expensive, it does not require any special infrastructure for communication between computational nodes. Being embarrassingly parallelizable this problem fits well to distributed computing approach.

It is important to note that adsorption energies $\mu(x)$ are not computed automatically from samples energies. Instead the function `set_ads_energy()` must be used to explicitly set adsorption energy for each adsorption complex taking into account temperature, partial pressure and adsorption enthalpy.

Statistical simulations

After the `energies` file is prepared SuSMoST can build a lattice model and you can proceed to conducting simulations and computations with statistical physics algorithms. Further we describe algorithms implemented in SuSMoST with usage examples, underlining the most important implementation features.

Metropolis Monte Carlo

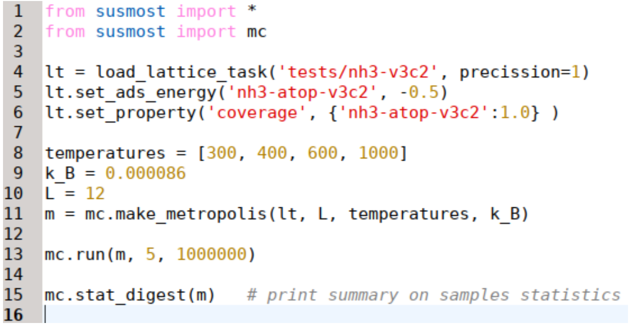
Metropolis Monte Carlo is the workhorse and reference method of statistical physics.[53] Probably it can be applied to any system. It enables one to study any characteristic of adsorption layer because it generates layer samples as XYZ files. The main shortcoming of the method the lack of reliable convergence criteria. It is difficult to estimate if your simulation have converged or not both with simulation time and with system size. This method suffers from the critical slowing down phenomenon, so it is hard to use in the vicinity of phase transitions. Due to its stochastic nature, the Metropolis Monte Carlo results have non-zero variation. So for numeric differentiation sometimes it is necessary to regularize them. It complicates simulation of thermal desorption spectra, estimation of heat capacity and susceptibility of the adsorption layer. SuSMoST includes routines for statistical analysis of Metropolis Monte Carlo simulation outcomes. In particular it computes autocorrelation times and effective sample size. With this figures one can relatively estimate the quality of convergence.

Temperature replica exchange

Temperature replica exchange also known as parallel tempering[53] is the modification of a classical Metropolis method designed to overcome critical slowing down effect and to avoid trapping in local minima. Of all Monte Carlo based methods implemented in SuSMoST this method is the most efficient for parallel computations. The best speed up is archived if replica temperatures are selected in such a way that in the vicinity of temperature driven phase transitions there are more replicas than in temperature ranges of stable phases.

An example of code that conducts simulation with parallel tempering in SuSMoST is shown in figure 10. This code is also provided in Supporting Information in file `test_mc_nh3-v3c2.py` along with `tests/nh3-v3c2` directory. This code creates a lattice model analyzing AC samples, the `energies` file and other files from `tests/nh3-v3c2` directory. Then it sets adsorption energy and the `coverage` parameter value for the AC `nh3-atop-v3c2`. Further, the function `mc.make_metropolis()` initializes data structures for parallel simulation of four surface replica of size 12 × 12 at temperatures 300 K, 400 K, 600 K and 1000 K. Parameter `k_B` specifies Boltzmann constant, defining thereby units of energies for particular simulation. Then the function `mc.run()` performs the actual simulation. $5 \cdot 10^6$ Monte Carlo attempts are executed for relaxation, then the same number of attempts is used to collect samples of adsorption layer parameters (energy, coverage, *etc*). Samples of the entire adsorption layer are saved each 10^6 attempts into files named as `traj.`*`temperature`*`.xyz`. An example of the adsorption layer sample at 300 K is shown in Figure 11. Function `mc.stat_digest` shows statistical characteristics of parameter samples for each temperature. An examples of `mc.stat_digest` output is shown in Figure 12. For each temperature and for each parameter (`’coverage’` in this example) and energy the sample size, sample mean, deviation, autocorrelation parameters, effective sample size are printed.

The code in Figure 10 must be started as four MPI tasks (one task per replica). On average modern personal computer this computation takes about 15 seconds.



```python
from susmost import *
from susmost import mc

lt = load_lattice_task('tests/nh3-v3c2', precission=1)
lt.set_ads_energy('nh3-atop-v3c2', -0.5)
lt.set_property('coverage', {'nh3-atop-v3c2':1.0} )

temperatures = [300, 400, 600, 1000]
k_B = 0.000086
L = 12
m = mc.make_metropolis(lt, L, temperatures, k_B)

mc.run(m, 5, 1000000)

mc.stat_digest(m)   # print summary on samples statistics

```

Figure 10: Code example for simulation of NH_3/V_3C_2 adsorption layer with parallel tempering Metropolis Monte Carlo method in SuSMoST.

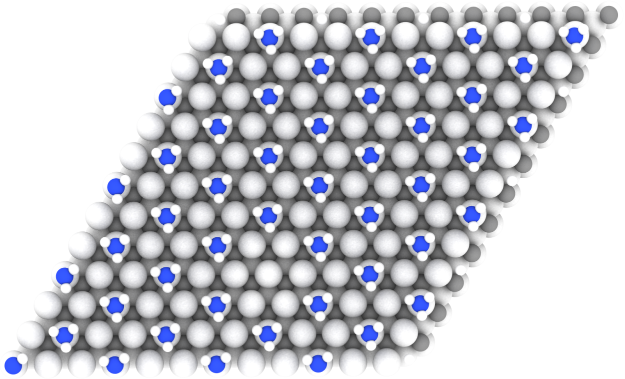
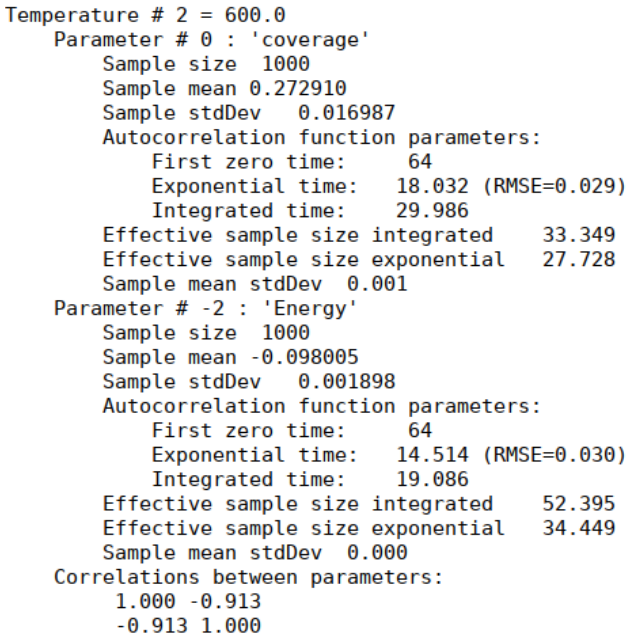


Figure 11: Sample of NH_3/V_3C_2 adsorption layer at 300 K.



```
Temperature # 2 = 600.0
    Parameter # 0 : 'coverage'
        Sample size  1000
        Sample mean 0.272910
        Sample stdDev   0.016987
        Autocorrelation function parameters:
            First zero time:     64
            Exponential time:   18.032 (RMSE=0.029)
            Integrated time:    29.986
        Effective sample size integrated    33.349
        Effective sample size exponential   27.728
        Sample mean stdDev  0.001
    Parameter # -2 : 'Energy'
        Sample size  1000
        Sample mean -0.098005
        Sample stdDev   0.001898
        Autocorrelation function parameters:
            First zero time:     64
            Exponential time:   14.514 (RMSE=0.030)
            Integrated time:    19.086
        Effective sample size integrated    52.395
        Effective sample size exponential   34.449
        Sample mean stdDev  0.000
    Correlations between parameters:
         1.000 -0.913
        -0.913 1.000
```

Figure 12: Example of `stat_digest()` output.

Wang-Landau

Wang-Landau algorithm is another Metropolis based algorithm designed to overcome critical slowing down and local minima trapping.[54–56] To get performance benefits and accurate results it is crucial to choose right energy interval and histogram step. An example of the code that use Wang-Landau method implementation from SuSMoST is provided in Supporting Information in file `test_wl_nh3-v3c2.py`. This code is similar to the parallel tempering example, so we do not discuss it in details here. It is convenient, that Wang-Landau simulations produce the density of states so one can compute thermodynamic quantities (free energy, entropy, heat capacity) for any temperature from a singe simulation.

Transfer-matrix

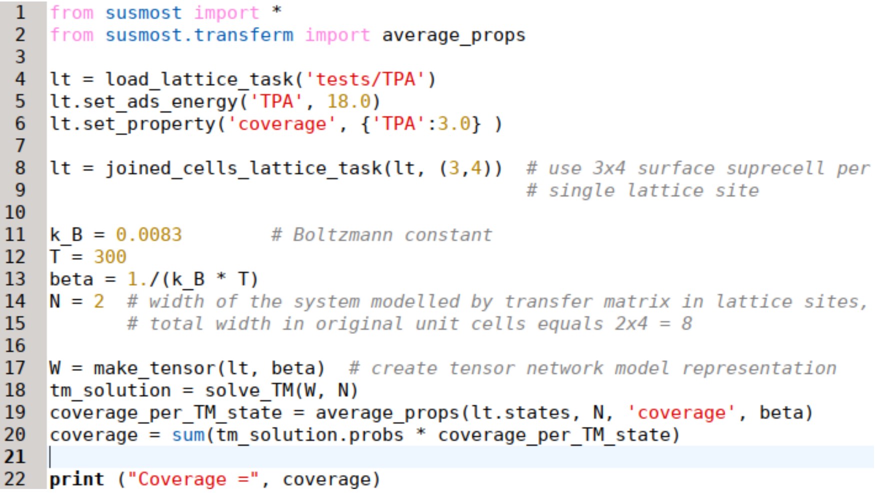
Transfer-matrix method is the numerically exact method with explicit convergence criteria.[42] It weakly suffers from critical slowing down - although in the vicinity of a phase transition computations takes longer time, it is always possible to get desired accuracy. Unlike Monte Carlo methods, this method considers the simulated system as semi-infinite, with one finite dimension and another infinite one. In fact the ribbon of finite width and infinite length is simulated. As the periodic boundary conditions are implied along the finite dimension, one can also image the simulated system as a tube of infinite length or a stack of rings of finite circumference. Therefore this method is exact for naturally one-dimensional systems, such as surface terraces or nanotubes. In case of a two-dimensional system it is necessary to reach convergence with system width. Due to high accuracy of computations for particular width it suits well for finite-size scaling. But for scaling it is necessary to use only those widths that are multiple of unit cell sizes of expected phases. Hence, a preliminary analysis of the ground state is necessary for efficient use of computing resources. The main limitation of the TM method is an amount of memory required. It grows exponentially with system width. Therefore, it is hard to use in case of presence of phases with large unit cells or in case of long-range lateral interactions. To some extent the problem can be solved by joining of lattice sites. SuSMoST provides a special function to do it - `join_cells_lattice_task()`.

The Figure 13 shows an example of the simplest program that use the TM method to calculate the mean coverage of copper surface by terephthalic acid (TPA) molecules.

Initially, the lattice model is created by the function `load_lattice_task()` according to the data from the `tests/TPA` directory. Functions `set_ads_energy()` and `set_property()` set the TPA AC adsorption energy equal to 18 kJ/mol and the surface coverage to 3.0. Then, with the `join_cells_lattice_task()` function, the model is reformulated in such a way that one lattice site corresponds to a 3×4 surface supercell. This is necessary to take into account long-range interactions and to make the system width multiple of the unit cell size. In this case the width of the expected phase unit cell is 4. The overall width of the system simulated by this program is equal to two lattice sites that corresponds to eight unit cells of the real surface. The `make_tensor()` function creates a tensor representation of the model using the Eq. 4. The function `solve_TM()` finds the eigenvalue, as well as the left and right eigenvectors of the transfer matrix and computes the probabilities of ring states from them.[42] The surface coverage for all possible rings is calculated by the `average_props()` function. Then the entire surface coverage is computed through element-wise multiplication and summation of the probabilities of the rings and their coverage. The program runs in 30 seconds on a modern PC and requires about 1 GB of RAM.

Tensor renormalization group

Tensor renormalization group (TRG) is also numerically exact method. It is based on reducing singluar value decomposition (SVD) of tensor T (Eq. 4) and on renormalization approach.[57] Contrary to TM method, TRG considers the simulated system as infinite in both dimensions. So one do not need to bother about convergence by system size. But due to usage of the reducing SVD the hardly estimatable approximation error is inherent in the TRG method. The greater the number of singular values kept in the tensor by reducing SVD, the smaller the approximation error. Therefore, like the TM method, the TRG method is



```
from susmost import *
from susmost.transferm import average_props

lt = load_lattice_task('tests/TPA')
lt.set_ads_energy('TPA', 18.0)
lt.set_property('coverage', {'TPA':3.0} )

lt = joined_cells_lattice_task(lt, (3,4))  # use 3x4 surface suprecell per
                                           # single lattice site

k_B = 0.0083        # Boltzmann constant
T = 300
beta = 1./(k_B * T)
N = 2  # width of the system modelled by transfer matrix in lattice sites,
       # total width in original unit cells equals 2x4 = 8

W = make_tensor(lt, beta)  # create tensor network model representation
tm_solution = solve_TM(W, N)
coverage_per_TM_state = average_props(lt.states, N, 'coverage', beta)
coverage = sum(tm_solution.probs * coverage_per_TM_state)

print ("Coverage =", coverage)
```

Figure 13: Code example for computation of equilibrium coverage of Cu(100) surface by terephthalic acid using transfer-matrix method in SuSMoST. Coverage is computed for temperature 300 K, adsorption energy 18 kJ/mol, assuming that one terephthalic acid molecule covers three adsorption sites.

demanding of the amount of memory. In case of long-range lateral interaction it is necessary to join lattice sites and to keep more singular values. This leads to increase in the amount of memory required.

An example of a program using the TRG method is given in the Supporting Information in file `test_trg_TPA.py`. The use of TRG in SuSMoST is similar to the TM method, since they both use the tensor representation of the model. Therefore, here we do not consider this code in detail. We only note that for the TRG method, in contrast to the TM, it is not required to math the size of the supercell with the period of expected phases. Also, since the TRG method returns only the logarithm of partition function, numerical differentiation is required to obtain heat capacity or layer density. This is usually not problematic, since the dependencies generated by the TRG are fairly smooth.

Conclusion

The software toolkit presented here enables one to study adsorption layers with a set of efficient methods of statistical physics. The toolkit can be used in a black-box manner virtually for any adsorption layer. Input data can be specified in an intuitive form based on atom-wise geometry of adsorption complexes. Model parameterization is assisted by automation scripts. Several algorithm of different nature can be uniformly used for a single system due to the unified model base. Thus shortcomings of some algorithms can be compensated and obtained results becomes more reliable. SuSMoST computes thermodynamic characteristics (free energy, adsorption layer coverage and density, heat capacity, entropy) and generates representative samples of adsorption layer structure as XYZ-files. Obtained thermodynamic characteristics can be used to plot adsorption isotherms, phase diagrams, temperature programmed desorption spectra. Samples of adsorption phases can be used for comparison with experimental images and diffraction spectra and for analysis of electronic structure of the phases. As SuSMoST is implemented as a Python library it is easy to script computations with it. One can explore phase space of parameters and visualize results. It is not necessary to follow the suggested workflow exactly. It is possible to use generated model representations for computations with another algorithm, that are not implemented in SuSMoST. On the other hand one can create or adjust models in any convenient way and still use SuSMoST algorithms for computations within these models. We hope that SuSMoST will become a daily tool both in experimental and computational surface science.

Acknowledgement

This study was supported by the Russian Science Foundation under Grant No. 17-71-20053.

Supporting Information Available

Examples of code with SuSMoST usage and sketches of adsorption complexes.

This material is available free of charge via the Internet at `http://pubs.acs.org/`.

References

(1) Goronzy, D. P.; Ebrahimi, M.; Rosei, F.; Arramel,; Fang, Y.; De Feyter, S.; Tait, S. L.; Wang, C.; Beton, P. H.; Wee, A. T. S.; Weiss, P. S.; Perepichka, D. F. *ACS Nano* **2018**, *12*, 7445–7481, PMID: 30010321.

(2) Bonomi, M.; Branduardi, D.; Bussi, G.; Camilloni, C.; Provasi, D.; Raiteri, P.; Donadio, D.; Marinelli, F.; Pietrucci, F.; Broglia, R. A.; Parrinello, M. *Computer Physics Communications* **2009**, *180*, 1961 – 1972.

(3) Burwell, R. *Manual of symbols and terminology for physicochemical quantities and units – appendix 2 : definitions, terminology and symbols in colloid and surface chemistry*; Pergamon: Oxford New York, 1976.

(4) Nath, T.; Rajesh, R. *Phys. Rev. E* **2014**, *90*, 012120.

(5) Zhang, W.; Deng, Y. *Phys. Rev. E* **2008**, *78*, 031103.

(6) Fefelov, V. F.; Gorbunov, V. A.; Myshlyavtsev, A. V.; Myshlyavtseva, M. D. *Phys. Rev. E* **2010**, *82*, 041602.

(7) Fefelov, V. F.; Gorbunov, V. A.; Myshlyavtsev, A. V.; Myshlyavtseva, M. D.; Akimenko, S. S. *Adsorption* **2013**, *19*, 495–499.

(8) Akimenko, S. S.; Fefelov, V. F.; Myshlyavtsev, A. V.; Stishenko, P. V. *Phys. Rev. B* **2018**, *97*, 085408.

(9) Centres, P. M.; Ramirez-Pastor, A. J.; Gimenez, M. C. *Physical Review E* **2018**, *98*.

(10) Slutskii, M. G.; Barash, L. Y.; Tarasevich, Y. Y. *Phys. Rev. E* **2018**, *98*, 062130.

(11) Ramirez-Pastor, A. J.; Centres, P. M.; Vogel, E. E.; Valdés, J. F. *Phys. Rev. E* **2019**, *99*, 042131.

(12) Sanchez-Varretti, F. O.; Pasinetti, P. M.; Bulnes, F. M.; Ramirez-Pastor, A. J. *Adsorption* **2017**, *23*, 651–662.

(13) Fefelov, V. F.; Stishenko, P. V.; Kutanov, V. M.; Myshlyavtsev, A. V.; Myshlyavtseva, M. D. *Adsorption* **2016**, *22*, 673–680.

(14) Fefelov, V. F.; Myshlyavtsev, A. V.; Myshlyavtseva, M. D. *Phys. Chem. Chem. Phys.* **2018**, *20*, 10359–10368.

(15) Fefelov, V. F.; Myshlyavtsev, A. V.; Myshlyavtseva, M. D. *Adsorption* **2019**,

(16) Silva, D.; Rikvold, P. A. *Phys. Chem. Chem. Phys.* **2019**, –.

(17) Rysko, W.; Nieckarz, D.; Szabelski, P. *Adsorption* **2019**, *25*, 75–85.

(18) Szabelski, P.; Nieckarz, D.; Rysko, W. *The Journal of Physical Chemistry C* **2017**, *121*, 25104–25117, 00001.

(19) Nieckarz, D.; Rysko, W.; Szabelski, P. *Physical Chemistry Chemical Physics* **2018**, *20*, 23363–23377.

(20) Nieckarz, D.; Szabelski, P. *Chemical Communications* **2014**, *50*, 6843.

(21) imnas, M.; Ibenskas, A.; Tornau, E. E. *The Journal of Physical Chemistry C* **2015**, *119*, 20524–20534.

(22) Ibenskas, A.; imnas, M.; Tornau, E. E. *The Journal of Physical Chemistry C* **2018**, 00000.

(23) Ibenskas, A.; imnas, M.; Kizlaitis, K. J.; Tornau, E. E. *The Journal of Physical Chemistry C* **2019**, *123*, 3552–3559.

(24) Ibenskas, A.; imnas, M.; Tornau, E. E. *The Journal of Physical Chemistry C* **2016**, *120*, 6669–6680.

(25) Akimenko, S.; Gorbunov, V.; Myshlyavtsev, A. *Procedia Engineering* **2015**, *113*, 108 – 112, Oil and gas engineering (OGE-2015) Omsk State Technical University, Omsk, Russian Federation, 25-30 April 2015.

(26) Myshlyavtsev, A. V.; Stishenko, P. V. *Surface Science* **2015**, *642*, 51 – 57.

(27) Gorbunov, V.; Myshlyavtsev, A.; Myshlyavtseva, M.; Fefelov, V. *Surface Science* **2015**, *636*, 1 – 7.

(28) Hoffmann, M. J.; Matera, S.; Reuter, K. *Computer Physics Communications* **2014**, *185*, 2138 – 2150.

(29) Andersen, M.; Panosetti, C.; Reuter, K. *Frontiers in Chemistry* **2019**, *7*, 202.

(30) CARLOS/Kinetix. `http://carlos.win.tue.nl/`.

(31) Stamatakis, M.; Vlachos, D. G. *The Journal of Chemical Physics* **2011**, *134*, 214115.

(32) Zacros. `http://zacros.org/`.

(33) Alzate-Cardona, J. D.; Sabogal-Su'arez, D.; Arbel'aez-Echeverri, O. D.; Restrepo-Parra, E. Vegas: Software package for the atomistic simulation of magnetic materials. 2018.

(34) Plimpton, S.; Battaile, C.; Chandross, M.; Holm, L.; Thompson, A.; Tikare, V.; Wagner, G.; Webb, E.; Zhou, X.; Garcia Cardona, A., C. Slepoy *Crossing the Mesoscale No-Man's Land via Parallel Kinetic Monte Carlo*; Sandia National Laboratories: Sandia, USA, 2009.

(35) Chang, J. H.; Kleiven, D.; Melander, M.; Akola, J.; Lastra, J. M. G.; Vegge, T. *Journal of Physics: Condensed Matter* **2019**,

(36) Python Wheels. `https://pythonwheels.com/`.

(37) Akimenko, S. S.; Gorbunov, V. A.; Myshlyavtsev, A. V.; Stishenko, P. V. *Phys. Rev. E* **2016**, *93*, 062804.

(38) Togo, A.; Tanaka, I. *arXiv e-prints* **2018**, arXiv:1808.01590.

(39) Mortensen, J. J.; Hansen, L. B.; Jacobsen, K. W. *Phys. Rev. B* **2005**, *71*, 035109.

(40) Larsen, A. H. et al. *Journal of Physics: Condensed Matter* **2017**, *29*, 273002.

(41) Larsen, A. H. et al. *Journal of Physics: Condensed Matter* **2017**, *29*, 273002.

(42) Baxter, R. *Exactly solved models in statistical mechanics*; Academic Press: London New York, 1982.

(43) Nishino, T. *Journal of the Physical Society of Japan* **1995**, *64*, 3598–3601.

(44) Akimenko, S.; Gorbunov, V.; Myshlyavtsev, A.; Fefelov, V. *Surface Science* **2015**, *639*, 89 – 95.

(45) Gorbunov, V. A.; Myshlyavtsev, A. V.; Myshlyavtseva, M. D.; Fefelov, V. F. *Protection of Metals and Physical Chemistry of Surfaces* **2013**, *49*, 379–385.

(46) Gorbunov, V. A.; Akimenko, S. S.; Myshlyavtsev, A. V.; Fefelov, V. F.; Myshlyavtseva, M. D. *Adsorption* **2013**, *19*, 571–580.

(47) Fefelov, V.; Gorbunov, V.; Myshlyavtsev, A.; Myshlyavtseva, M. In *Thermodynamics*; Morales-Rodriguez, R., Ed.; IntechOpen: Rijeka, 2012; Chapter 15.

(48) Fefelov, V.; Gorbunov, V.; Myshlyavtsev, A.; Myshlyavtseva, M.; Evseeva, S. *Applied Surface Science* **2010**, *256*, 5298 – 5304, Seventh International Symposium Effects of Surface Heterogeneity in Adsorption and Catalysis on Solids - ISSHAC-7.

(49) Exteneded XYZ format in `quippy` library documentation. `https://libatoms.github.io/QUIP/io.html#module-ase.io.extxyz`.

(50) Mazilova, T. I.; Sadanov, E. V.; Mikhailovskij, I. M. *Philosophical Magazine Letters* **2018**, *98*, 310–318.

(51) Repp, J.; Moresco, F.; Meyer, G.; Rieder, K.-H.; Hyldgaard, P.; Persson, M. *Phys. Rev. Lett.* **2000**, *85*, 2981–2984.

(52) Schimka, L.; Harl, J.; Stroppa, A.; Grüneis, A.; Marsman, M.; Mittendorfer, F.; Kresse, G. *Nature materials* **2010**, *9*, 741.

(53) Frenkel, D.; Smit, B. *Understanding molecular simulation: from algorithms to applications*; Elsevier, 2001; Vol. 1.

(54) Wang, F.; Landau, D. P. *Phys. Rev. Lett.* **2001**, *86*, 2050–2053.

(55) Belardinelli, R. E.; Pereyra, V. D. *The Journal of Chemical Physics* **2007**, *127*, 184105.

(56) Shchur, L. *arXiv preprint arXiv:1808.09251* **2018**,

(57) Levin, M.; Nave, C. P. *Phys. Rev. Lett.* **2007**, *99*, 120601.

Graphical TOC Entry

We offer the scientific community the Surface Science Modelling and Simulation Toolkit (SuSMoST), which includes a number of utilities and implementations of statistical physics algorithms and models. With SuSMoST one is able to predict or explain the structure and thermodynamic properties of adsorption layers. SuSMoST generates formal graph and tensor-network model from atomic description of adsorption complexes, assists *ab initio* computation of interactions between adsorbed species, produces representative samples of adsorption layers, computes thermodynamic quantities enabling one to plot phase diagrams and thermal desorption spectra.