Efficient implementation of Cluster Expansion models in surface Kinetic Monte Carlo simulations with lateral interactions: Subtraction Schemes, Supersites and the Supercluster Contraction

Franziska Hess*[a,b]

a Department of Nuclear Science and Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA, 02139, USA

b Institute of Physical Chemistry, RWTH Aachen, Landoltweg 2, 52074 Aachen, Germany

* E-mail address of the corresponding author: hess@pc.rwth-aachen.de

Table S 1: Comparison between site-based and rate-based VSSM schemes

step	what is calculated?	site-based	rate-based
update	update	$k_{x,y,i}$ (rate constants on site *x*, *y*, *i*)	n_k (number of occurrences of rate constant $1 \leq k \leq M$)
accounting	Γ	$\Gamma = \sum_{x,y,i} r_{x,y,i}$	$\Gamma = \sum_{k} n_k r_k$
search	ν	$\sum_{j=1}^{\nu-1} r_j \leq \rho_1 \Gamma \leq \sum_{j=1}^{\nu} r_j$ (j runs over (x,y,i))	1. $\sum_{k=1}^{\nu-1} n_k r_k \leq \rho_1 \Gamma \leq \sum_{k=1}^{\nu} n_k r_k$ 2. select an (x,y,i) from list ν at random (ρ_2).

1. Additional information about the subtractive scheme

1.1. Numerical accuracy

We have shown that updating the rate sum Γ in a subtractive scheme improves overall performance and turns the scaling of the accounting step from $\mathcal{O}(N)$ to $\mathcal{O}(1)$. However, this approach is not without drawbacks. One problem stems from the fact that floating point numbers cannot be accurately represented in binary, and numerical precision is finite (floating point error). For instance, in Fortran single and double precision, only seven and fifteen digits are stored, respectively. This makes it impossible to correctly compute small changes of large numbers (such as $10^{16} + 1$). In KMC simulations, reaction rates frequently differ by several orders of magnitude, which make Γ calculated by the subtractive accounting scheme numerically imprecise, especially in stiff systems, where the highest

and lowest rate constants differ by several orders of magnitude. In our example simulation, the final values of Γ are output together with the timing information, and one can clearly see that the final results slightly differ between the standard and subtractive accounting schemes, and increasing the number of steps in the simulation increases this deviation. This is not a bug, but a result of the floating-point error, which can cause errors in the simulation under certain circumstances. After discussing the origin of this error, we propose an efficient error tracking and handling algorithm.

The same problem does not arise in the standard accounting scheme, where Γ is reset and calculated from scratch in every iteration (Eq. 6), as the numerical error of Γ depends only on the its value (approximately $10^{-7}\Gamma$ and $10^{-15}\Gamma$ for single and double precision, respectively). In the subtractive scheme, on the other hand, Γ is reused in subsequent iterations, so that the numerical error of Γ depends on all of its previous values, i.e., the error accumulates over the course of the simulation. If Γ is stored in single precision, the error after 10^7 steps becomes greater that Γ itself. This can cause errors in the selection of reaction event in cases where $\rho_1\Gamma_{\text{subtractive}}$ (with ρ_1 a random number) happens to be larger than Γ_{exact}. Another possible side-effect is that $\Gamma_{\text{subtractive}}$ can assume negative values, which is of course unphysical. This is not a very big problem in practical application, however, as this kind of error is easy to manage as will be shown in the following.

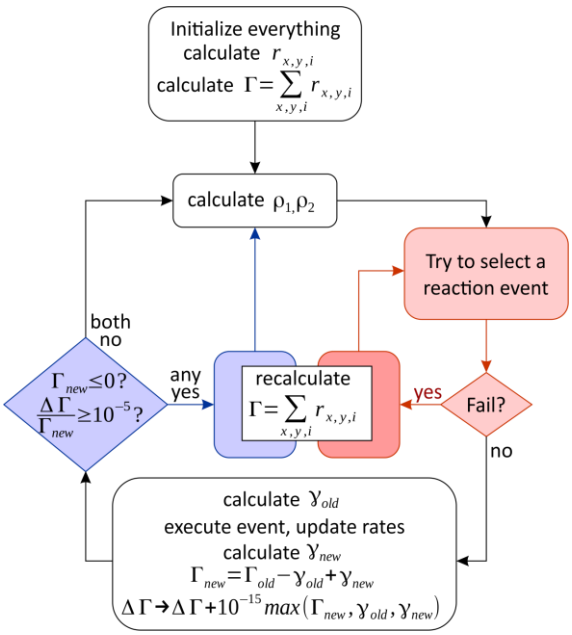

Figure S 1: Error handling algorithm in the Subtractive Γ calculation scheme. Highlighted in red and blue are the two sanity checks and subsequent error correction.

First of all, we strongly advise on storing Γ, γ and reaction rates in double precision. This alone sufficiently mitigates the error in many cases. For stiff systems and lengthy simulation runs, we have devised a reliable error tracking and handling algorithm as schematically shown in **Figure S 1**. The sample code for error handling in the Subtractive Scheme is included in the `linear_search`, `supersite_search` and `subtractive_gamma` functions of the `supersite.f90` program.

The algorithm keeps track of an error estimate $\Delta\Gamma$ by adding 10^{-15} times the largest term in Eq. 9 to the previous $\Delta\Gamma$ in every step (bottom center white box in **Figure S 1**):

$$\Delta\Gamma \rightarrow \Delta\Gamma + 10^{-15} \max(\Gamma_{\mathrm{old}}, \gamma_{\mathrm{old}}, \gamma_{\mathrm{new}}). \qquad \text{Eq. S 1}$$

The error estimate $\Delta\Gamma$ then indicates the order of magnitude of the error, and Γ should be recalculated from scratch (Eq. 6) either if $\Delta\Gamma/\Gamma_{\mathrm{new}}$ exceeds a certain threshold value, for instance 10^{-5}, or $\Gamma_{\mathrm{new}} < 0$ after evaluating Eq. 9 (blue part of **Figure S 1**). In addition, the search algorithm needs an error handler that detects if $\rho_1\Gamma_{\mathrm{subtractive}}$ points to a non-existing elementary step (red part of **Figure S 1**). This is the case when the search algorithm finishes its loop without finding an elementary step ν that fulfills the condition

$$\sum_{j=1}^{\lambda-1} r_j \leq \rho_1\Gamma \leq \sum_{j=1}^{\lambda} r_j, \qquad \text{Eq. S 2}$$

in which case Γ needs to be recalculated as well, and the search is reattempted. Note that this process is drawn as a loop in **Figure S 1**, and in fact is implemented as a recursive function in our sample program. However, the search should always succeed after re-evaluation of Γ, and if more than two search attempts are made in a row, this indicates errors unrelated to the floating-point error discussed here.

In many simulations, however, these scenarios never occur as $\Gamma_{\mathrm{subtractive}}$ is a sufficiently good approximation of Γ_{exact} in non-stiff systems with double precision. But even in a stiff system like the HCl oxidation over RuO_2(110), the recomputation of Γ occurs only about once in 10^6 steps, which is a small price to pay for overall vastly superior computation speed and scaling (cf. **Section 3.4**). In some cases, however, frequent recalculation of Γ may noticeably impact scaling and run time, and the error handling proposed here may be undesirable. But even in such cases one does not need to give up on the Subtractive Scheme, as the error can also be efficiently mitigated by maintaining several variables lists for storing and evaluating Γ, γ_{old} and γ_{new}, where contributions on different orders of magnitude are handled separately, each retaining its respective accuracy until they are finally added together. This approach would entirely remove the need to recompute Γ from scratch, instead adding a larger effort of $\mathcal{O}(1)$. We have yet to come across a system where this option would be a good choice and do not provide sample code for it, but propose this method as a alternative strategy to handle the floating-point error in the Subtractive Scheme.

1.2. Other applications of the Subtraction Scheme

As indicated at the start of **Section 3.2**, while Γ is definitely the most important quantity to address with the Subtractive Scheme, it is not the only that can be treated in this fashion. In principle, it can be applied to any quantity whose evaluation involves a single sum over all (or many) sites of the lattice that undergoes only a small, localized change between successive steps. Possible applications include, but

are not limited to, surface coverages and total surface energies. Beside these trivial options, however, the Subtractive Scheme offers a way to treat long-range interactions in the update step efficiently.

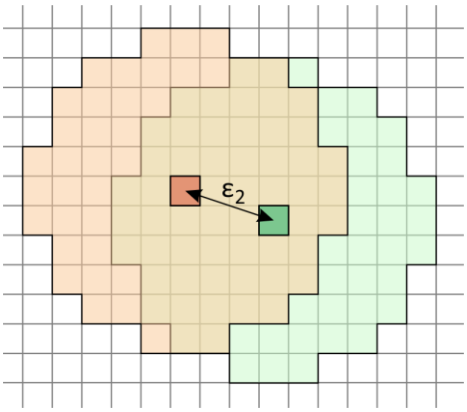

Figure S 2: Illustration of the subtractive energy update scheme for adsorbates with long-range lateral interactions. Here a reaction event occurs on the orange site, and the orange area indicates the updated zone of interacting sites. If the rate on the green site is to be updated, rather than re-evaluating the whole cluster expansion (for the whole green area), only the terms that involve both the orange and the green site need to be updated, for instance the pairwise interaction, ε_2.

Consider a surface with charged adsorbates and long-range interactions. Assuming a cutoff radius of 15 Å, the interaction radius would contain 4-5 sites on the RuO_2(110) surface, as shown in **Figure S 2**. A cluster expansion that contains only pairwise interactions at R = 5 would have 89 terms (including the zero-coverage adsorption energy). After a reaction event on the orange site, the rates on all 89 interacting sites (indicated in light orange) need to be updated, and each of these 89 surrounding sites (for instance, the green site) again has 89 sites within its interaction radius (drawn in light green), resulting in 89·89 interaction terms evaluated. However, considering that a cluster expansion considers only interaction clusters for specific sites, we realize that for the green site, only one pairwise ε_2 interaction term changes due to the reaction event (indicated by arrows). It is therefore unnecessary to re-evaluate the whole cluster expansion. Instead, one can store the energy of each adsorbate in each step, and then the energy update reads, similar to Eq. 9:

$$E_{\mathrm{new,x,y}} \rightarrow E_{\mathrm{old,x,y}} - \varepsilon_{2,old} + \varepsilon_{2,new} \qquad \text{Eq. S 3}$$

Here, $E_{\mathrm{new,x,y}}$ and $E_{\mathrm{old,x,y}}$ stand for the new and old adsorption energies on site (x, y), and $\varepsilon_{2,old}$ and $\varepsilon_{2,new}$ stand for the old and new interaction energies of the pairwise interaction subject to the update. This reduces the energy update to three terms to be evaluated (opposed to 89) if only pairwise interactions are considered, regardless of the range of lateral interactions (89·3 terms in total). Of course, this approach can be combined with the Supercluster Contraction (in which case the respective superclusters take the positions of $\varepsilon_{2,old}$ and $\varepsilon_{2,new}$ in Eq. S 3). This is also applicable to shorter-range interactions, but the expected boost in performance for our test system will be marginal at best because our supercluster contraction has already been reduced to just a few terms at short interaction range.

Although surface adsorption systems with long-range interactions have so far not been studied with KMC methods, there already are possible applications for the Subtractive Scheme for updating the

energy. For instance, on metal surfaces, where the density of undercoordinated sites is much higher than on oxide surfaces, an interaction model can easily contain eight nearest-neighbor (8NN) and higher interactions within a relatively "short" interaction radius as shown by Bajpai et al. for the NO-O-vacancy system on Pt(111) [1]. Or, if different site types are considered (on top, bridge, hcp, fcc sites) on metal surfaces, even Cluster Expansion models with short-range interactions can already include an impressive number of terms, for instance the O-vac cluster expansion for the Pt(321) surface by Bray et al. [2, 3]. This approach may also be useful for the study of defect in the bulk, where interactions can have quite a long range and the interacting sites form a sphere around the event center. Even for short-range interaction, many sites need to be updated in 3D systems, which results in a potentially high computational effort in KMC simulations employing such models.

2. Additional information about the supersite algorithm

2.1. Choice of s_x and s_y

The previous algorithm contains the size of the supersite, s_x and s_y, as parameters, and the performance of the algorithm is sensitive to the choice of these parameters. The attentive reader may have noticed that the cases where $s_x = s_y = 1$ or $s_x = n_x$ and $s_y = n_y$, i.e., where a supersite contains either one or all the sites on the surface are similar to the linear search algorithm, and give similar performance as the number of operations and comparisons is the same. The run time of the Supersite Search algorithm for three different lattice sizes (small: 36×36, medium: 144×144 and large: 576×576) was measured using the sample program `supersite.f90` for the same cluster expansion as before (evaluated by Eq. 1) and is shown in **Figure S 3a**.

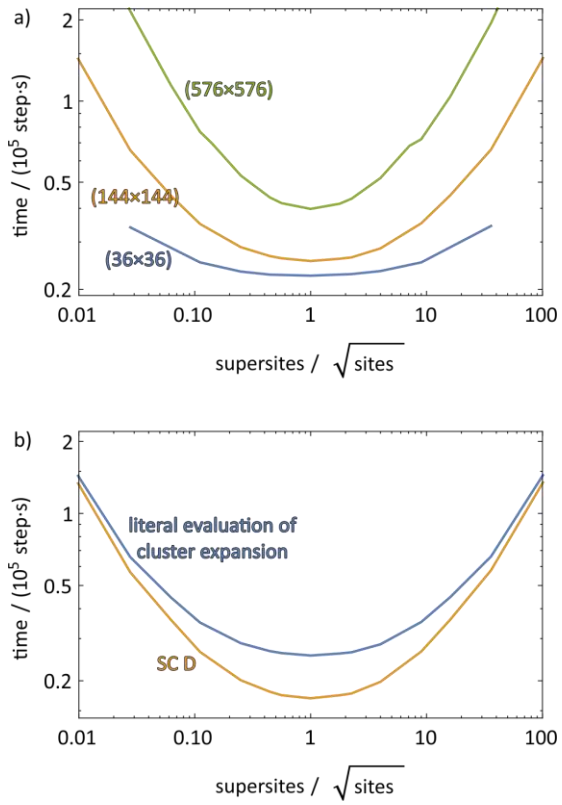

Figure S 3: a) Total run time of our sample program with Supersite Search, local update and the Subtractive Scheme from **Section 3.2** for different lattice sizes (small: 36×36, medium: 144×144 and large: 576×576) for different choices of the number of supersites, employing literal evaluation of the cluster expansion (Eq. 1). b) Run time comparison between literal cluster expansion evaluation and SC D for the medium (144×144) lattice.

The total time for 10^5 is plotted over the number of supersites, normalized by the square root the total number of sites. For all three lattice sizes, a curve with a pronounced minimum is obtained, which appears as a parabola in the log-log plot, indicating that both very large (left end) and very small (right end) supersizes are not advantageous. The minima of each curve are located close to one, i.e., where the number of supersites is equal to the square root of the total number of sites. This behavior is as theoretically expected for the two-level scheme and identical to the observation by Maksym [4].

The minimum appears quite shallow for the small lattice (**Figure S 3a**, blue curve), which is due to the fact that the Supersite Search, which scales as $\mathcal{O}(\sqrt{N})$ is coupled sequentially to several $\mathcal{O}(1)$ steps, namely the local update, which includes the energy evaluation, and the Subtractive Scheme for the evaluation of Γ. For small lattice sizes these $\mathcal{O}(1)$ parts are actually what determines the overall performance. Replacing, for instance, the literal evaluation of the cluster expansion (**Figure S 3b**, blue curve) by the Supercluster Contraction SC D (**Figure S 3b**, orange curve) in the medium lattice simulation significantly decreases the minimum run time, but does not otherwise alter the shape of the curve.

References

(1) Bajpai, A.; Frey, K.; Schneider, W. F. *The Journal of Physical Chemistry C* **2017**, *121*, 7344–7354.

(2) Bray, J. M.; Smith, J. L.; Schneider, W. F. *Topics in Catalysis* **2014**, *57*, 89–105.

(3) Bray, J. M.; Schneider, W. F. *ACS Catalysis* **2015**, *5*, 1087–1099.

(4) Maksym, P. *Semicond. Sci. Technol.* **1988**, *3*, 594.