1. Introduction

SPPARKS (Stochastic Parallel PARticle Kinetics Simulator) is a massively parallel kinetic Monte Carlo (KMC) engine designed to model time-evolving processes that are governed by stochastic reaction events. In contrast to molecular dynamics, which integrates deterministic equations of motion, SPPARKS employs a continuous-time Monte Carlo algorithm based on the residence-time (or Bortz–Kalos–Lebowitz) method to propagate system evolution through discrete reaction events. Within this framework, each possible event—such as adsorption, desorption, ligand exchange, diffusion, or surface rearrangement—is assigned a rate constant determined by the underlying reaction physics. The simulator then samples event occurrences according to their relative probabilities and advances the simulation clock by statistically exact time increments derived from the cumulative rate of all accessible events.

A key advantage of SPPARKS lies in its lattice-based architecture and domain-decomposition parallelism, which allow efficient scaling to large reaction networks and extended substrates. By encoding surface states, reaction rules, and event rates in user-defined KMC models, SPPARKS provides a flexible platform for studying rare-event phenomena in thin film growth, catalysis, and solid-state reaction kinetics. In this work, the code has been extensively customized to describe the elementary reactions of Al_2O_3 atomic layer deposition, including steric constraints, site-specific adsorption kinetics, and reaction pathways parameterized by first-principles free-energy profiles.

It should be emphasized that the present document is not intended to serve as a general introduction to SPPARKS. Readers who have never used SPPARKS before are strongly encouraged to consult the official documentation and tutorials available on the project website, which provide the necessary background on installation, lattice construction, event syntax, and basic KMC workflows. The current tutorial assumes familiarity with the standard usage of SPPARKS and focuses exclusively on the customized components developed for this work.

The source code released in this repository has been specifically engineered to model the atomic layer deposition of Al_2O_3 using trimethylaluminum and water. All reaction rules, rate expressions, lattice definitions, and steric constraints are tailored to this chemistry. Consequently, the implementation is not intended to be transferable to other precursor–substrate systems without substantial modification of the reaction catalogue and underlying algorithms.

This tutorial provides a detailed explanation of the input file *in.ald* and *data.ald*, which encapsulates the model configuration, reaction network, simulation parameters, and workflow required to reproduce the KMC simulations reported in the accompanying manuscript. For

users who wish to extend the code or modify functionalities within the custom SPPARKS source directory, a working knowledge of C++ and the SPPARKS class hierarchy is essential. The core logic of event scheduling, site-state definitions, and rate evaluation resides in the modified source files, and safe modification requires understanding the interplay between these components.

2. Explanation of the *in.ald* Input File

2.1 Initialization

The initialization block defines the global settings required for launching a kinetic Monte Carlo simulation within the customized SPPARKS framework. In this project, five keywords are used in the initial section of *in.ald*: seed, app_style, read_sites, sector, and solve_style. These options establish the stochastic behavior, simulation domain, and the calculation protocol employed by the ALD-specific KMC engine.

```
seed 12345

app_style ald/TMA

read_sites data.ald

sector no

solve_style linear
```

seed

This parameter specifies the integer random seed used by the residence-time algorithm. Because KMC simulations intrinsically rely on stochastic event sampling, the seed value determines the exact trajectory of events produced during a run. To assess statistical variability or to generate multiple realizations of the growth process, users may simply vary the seed value while keeping all other parameters fixed. Different seeds lead to different—but statistically equivalent—time evolutions.

app_style

The application style defines the KMC model to be invoked. In the present implementation, the customized module is registered as ald/TMA, which corresponds to the reaction network and lattice logic developed for TMA/H_2O ALD. This setting should not be modified unless

the underlying source code has been changed to implement a different chemistry or mechanism. For all reproductions of our simulations, the default ald/TMA must be retained.

read_sites

This command loads the preconstructed site-structure file containing the lattice geometry, site types, and initial surface states. The argument to read_sites should be replaced with the filename of the site configuration prepared in advance. This file defines the number of sites, their spatial arrangement, and their initial chemical identities, all of which are essential for initializing the KMC lattice prior to introducing reaction events.

sector

This setting determines the domain partition used internally by SPPARKS to organize lattice sites for event scheduling. Users should preserve the default value. Modifying this parameter is neither necessary nor recommended unless advanced users are restructuring the lattice or altering parallel decomposition.

solve_style

This option specifies the residence-time solution method used for selecting and executing events. The default method implemented in this work has been validated for numerical accuracy and performance; consequently, the default solve_style must be maintained. Altering this option may lead to inconsistent time advancement or incompatible behavior with the customized event classes.

2.2 Event Definition

The core of the *in.ald* input file is the event-definition block, in which all elementary reactions available to the KMC engine are explicitly enumerated. Each event corresponds to a reversible or irreversible chemical transformation between one or two interacting lattice sites, and represents the atomistic reaction steps that constitute the TMA/H_2 O ALD cycle. SPPARKS reads these events sequentially and constructs the full reaction catalogue from which the residence-time algorithm selects possible transitions during the simulation.

In this customized ALD module, each reaction is specified through an event command using the following general syntax:

```
event type old1 new1 old2 new2 A exp energy c.n1 c.n2 time label
```

Each field in this line defines a structural or kinetic property of the reaction:

Event Type (type)

The type keyword determines the topological pattern of the reaction—whether it involves

only the central site or also one of its neighbors. Four event types are supported:

Type 1: Single-site reaction

The event modifies only the central reaction site. In this case, old2, new2, and c.n2 are not required and should be omitted.

Type 2: First-nearest-neighbor reaction

This event couples the central site with one of its first-nearest neighbors. Both old2 and new2 must be provided.

Type 3: Second-nearest-neighbor reaction

This class describes reactions that depend on the central site and one of its second-nearest neighbors.

Type 4: Same–z-coordinate nearest-neighbor reaction

Events of this type involve the reaction site and another site located at the same vertical (z-axis) level, but belonging to a different lateral layer.

Site-State Fields (old1, new1, old2, new2)

These four parameters define the change in chemical identity of the sites involved:

old1: the initial state of the central reaction site

new1: the resulting state of the central site after the reaction

old2: the initial state of the neighbor site (only for types 2–4)

new2: the resulting state of the neighbor site

A valid event is triggered only when the participating site(s) match the required old states. Upon execution, SPPARKS updates the relevant sites to the specified new states.

Kinetic Parameters (A, exp, energy)

The reaction rate is defined using an Arrhenius-type expression:

$$k = AT^{exp}e^{\left(-\frac{energy}{k_B T}\right)}$$

where A is the pre-exponential factor, exp is the temperature exponent, and energy is the activation barrier (input in eV). These parameters originate from the first-principles free-energy calculations described in the main text.

Coordination Requirements (c.n1, c.n2)

These optional constraints specify the coordination number (or local environment condition) required for the reaction to be eligible.

They are used primarily to model:

Steric blocking effects

Saturation limits

Site-counting–based reaction availability

For type 1 events, only c.n1 is relevant. For types 2–4, both c.n1 and c.n2 can be used to encode local structural rules.

Time Window (time)

The c.n fields constrain whether an event may occur based on the local coordination environment of the involved sites. These parameters are used to implement steric blocking rules, site-saturation conditions, and structural prerequisites for certain reactions.

In addition to integer values, the customized ALD module supports a special keyword:

all: no coordination restriction
When c.n = all, the event is independent of the local coordination environment and may proceed whenever the participating sites match their required old states.

For Type 1 (single-site) events, only c.n1 is relevant. For Types 2–4, both c.n1 and c.n2 may be specified.

Event Label (label)

Each event must be assigned a unique label in the number field. In this project, we adopt a prefix-based naming convention that encodes the event type directly in its identifier, facilitating debugging, post-analysis, and reaction categorization:

s — Type 1 (single-site events)

d — Type 2 (first-nearest-neighbor events)

v — Type 3 (second-nearest-neighbor events)

f — Type 4 (same-z-coordinate nearest-neighbor events)

For example:

s1 denotes the first single-site reaction,

d12 denotes the twelfth first-neighbor reaction,

v3 denotes the third second-neighbor reaction, and

f5 denotes the fifth cross-layer event.

This structured labeling system helps users locate, interpret, and validate specific reaction types when analyzing simulation outputs.

2.3 Simulation Parameter Settings

The simulation-parameter block of *in.ald* specifies the temporal and thermodynamic conditions under which the ALD cycle is executed. These parameters define the durations of precursor exposures, purge intervals, and the temperature at which all reaction rates are evaluated. Three keywords are used in this section: pulse_time, purge_time, and temperature.

pulse_time

The pulse_time command sets the durations of the precursor exposure steps. It requires two numerical inputs:

```
pulse_time     tma_pulse     h2o_pulse
```

tma_pulse: duration of the TMA exposure step (in seconds)

h2o_pulse: duration of the H_2O exposure step (in seconds)

Only events tagged with time = 1 (TMA-pulse-only) are active during the first interval, and events with time = 2 (H_2O-pulse-only) are active during the second. Events with time = 0 remain active throughout both intervals.

These parameters allow the simulation to match the precursor residence times used in experimental ALD processes or to explore pulse-length effects systematically.

purge_time

The purge_time command defines the durations of the inert purge steps separating the TMA and H_2O pulses. It also requires two numerical arguments:

```
purge_time     purge_after_tma     purge_after_h2o
```

purge_after_tma: duration of the purge following the TMA pulse

purge_after_h2o: duration of the purge following the H_2O pulse

During purge windows, only events labeled with time = 0 are active; both TMA- and

H_2 O-specific reactions are disabled. These intervals ensure that no cross-reactions between residual precursors occur, mimicking the ideal ALD self-limiting behavior.

temperature

The temperature command sets the simulation temperature in Kelvin:

```
temperature     T
```

All reaction rates are evaluated using this temperature through the Arrhenius expression defined in the event catalogue. The temperature affects both the magnitude of the rate constants and the competition among parallel reaction pathways. Users should ensure that the temperature matches the conditions under which the free-energy barriers were evaluated.

2.4 Output Setting

The output block in *in.ald* controls how diagnostic information, event statistics, and lattice configurations are recorded during the KMC simulation. Four commands are relevant in this section: diag_style, stats, dump, and run. Together, they define what information is collected, how frequently it is written, and the total duration of the simulation.

```
diag_style ald/TMA stats yes list events QCM species

stats .001

dump 1 text .01 dump.ald id i1 i2 x y z

run 320
```

diag_style

The diag_style command specifies the diagnostic output format used throughout the simulation. In addition to reporting the standard SPPARKS KMC quantities—such as the number of accepted events, rejected events, simulation time, and sweep count—the customized ald/TMA diagnostic module allows users to track the populations of individual surface species and the occurrence frequency of each reaction.

Users may modify the list of tracked quantities by editing the source file *diag_ald_TMA.cpp.*

Any additional species counts, reaction counters, or derived quantities of interest can be inserted there. After recompilation, these new diagnostics will appear automatically in the output generated by diag_style.

stats

The stats command defines the time interval at which statistical information is written to the log file.

dump

The dump command outputs a full snapshot of the lattice, including the state of every surface site and its spatial coordinates. Such snapshots are useful for post-analysis, visualization, or debugging of the surface evolution.

run

The run command specifies the total duration of the KMC simulation.

3. Explanation of the *data.ald* File

The *data.ald* file provides the complete geometric and chemical specification of the lattice on which the KMC simulation operates. It is read by the read_sites command in *in.ald* and defines the spatial arrangement of surface sites, their neighbor relationships, and their initial chemical identities. The file is divided into three major sections: the header block, the site coordinates block, and the neighbor and state blocks.

3.1 Header: Global Lattice Information

The first section of *data.ald* contains global descriptors that define the size and topology of the simulation domain. These include:

dimension
Specifies the dimensionality of the lattice. For the dual-surface ALD model, this is typically set to 3.

sites
The total number of lattice sites in the system. Each site represents a surface reaction location and is assigned a unique index.

max neighbors
The maximum number of neighbors that any site can have. This value determines the memory allocation and neighbor-list format used by SPPARKS.

xlo xhi, ylo yhi, zlo zhi
The spatial extent of the simulation box in each direction. These bounds define the lattice vectors and are used to interpret the absolute coordinates of the surface sites.

This header block establishes the geometric framework used throughout the remainder of the KMC simulation.

3.2 Site Coordinates

The second section lists the positions of all lattice sites. Each line contains:

```
site_ID    x    y    z
```

where:

site_ID is the unique index of the site (from 1 to the total number of sites),

x, y, z are the coordinates of the site in the simulation box.

These coordinates determine the spatial organization of the adsorption sites and allow the KMC engine to identify first- or second-nearest neighbors based on pre-defined neighbor lists. They also support visualization of the evolving surface structure when used together with dump files.

3.3 Neighbor Lists

The third section specifies the neighbor relationships for each site. Each line follows the format:

```
site_ID    neighbor_1    neighbor_2    ...    neighbor_N
```

where the entries following site_ID correspond to the indices of all neighboring sites associated with that site. The number of neighbors must not exceed the max neighbors value declared in the header.

These neighbor lists are essential for enabling Type 2, Type 3, and Type 4 reactions, which require interaction between a central site and one of its neighbors.

3.4 Initial Site Values

The final section assigns a chemical state and coordination number to each site. Each line has the form:

```
site_ID    species_ID    coordination_number
```

species_ID is an integer code that denotes the chemical identity of the site.
These species identifiers correspond directly to the enumerations defined in *app_ald_TMA.cpp*, where all surface species are assigned unique numerical labels.

coordination_number is the initial coordination value (c.n) of the site, which may influence the eligibility of certain events in the KMC model, especially those governed by steric constraints.

This section provides the initial chemical configuration from which the ALD cycle begins.