MMonCa: A flexible and powerful new Kinetic Monte Carlo Simulator

I. Martin-Bragado¹, A. Rivera², G. Valles², MJ Caturla³

¹ Institute IMDEA Materials, Madrid, Spain
² Instituto de Fusión Nuclear (UPM), Madrid, Spain
³ Universidad de Alicante, Alicante, Spain

www.materials.imdea.org
Outline

- Why new code?
- The KMC algorithm & Object KMC: Rates
- MMonCa:
  - Ideas
  - Implementation
  - Physical mechanisms
  - Flexibility
- Examples
  - Isochronal annealing of $\alpha$-Fe
  - Annealing of implanted defects in Si
  - Evolution of damage in W
- Conclusions

ignacio.martin@imdea.org

COSIRES 2012. June 29th
Why developing new code?

- Kinetic models must reach high irradiation doses for very complex systems
  - Point defects and impurities
  - Extended defects, dislocations
  - Clusters (with helium, carbon...)
- So far there are no easy available “standard” KMC simulators.
- There is a need for **efficiency** and **flexibility**.
The KMC algorithm

• determine all possible processes for a given configuration and build a list of their rates $r_i$ (e.g. I jump, V jump, I emission from defect)

calculate the total rate of events $R = \Sigma r_i$

generate a random number $n \in [0,1)$

use $n$ to select an event from the list (e.g. “This Interstitial will jump ”)

Perform the event. Manage interactions between particles.

Advance time. New time = old time + $1/R$
Object KMC: Rates

- All different rates are assume to be produced by a prefactor and an activation energy.
- The activation energy is computed as the difference in formation energies between final and initial states plus a barrier (typically a migration energy).

\[ E_{ij} = E_{j}^{f} - E_{i}^{f} + E_{ij}^{b} \]

\[ r_{ij} = P_{ij} \times \exp\left(-\frac{E_{ij}}{k_B T}\right) \]
Development ideas

- **Modular**: Possibility for Object KMC, Lattice KMC and maybe others.
- **Versatile and flexible**: Build on top of Tcl scripting language.
- **Efficient and modern**: Written using Object Oriented Methodologies in C++
- **Professional development**, debugging, profiling and testing
- **User oriented**: Customizable through input files rather than code changes.

ignacio.martin@imdea.org 6 COSIRES 2012. June 29th
MMonCa: Implementation

- Four main areas: space, time, user interfaces and defects
- Defects belong to 5 categories: Extended Defects (ED), Mobile Particles (MP), Multi Clusters (MC), IV Damage (IV) and Interfaces (Int).

User input script

- Space
- Time
- Objects (Defects)
- ED
- MC
- IV
- MP
- Int

TCL library

Operative System (O.S.)

COSIRES 2012. June 29th
OKMC mechanisms

- **Diffusion**
  - $I, V, C, He, \ldots$
  - $He_n V_m, \ldots$
  - $I_n, V_m, \ldots$

- **Cluster emission**
  - $He_n V_m \leftrightarrow He_{n-1} V_{m-1} + He V$
  - $He_n V_m \leftrightarrow He_{n-1} V_{m-1} + He$
  - $He_n V_m \leftrightarrow He_{n-1} V_{m-1} + V$
  - $He_n V_m \leftrightarrow He_{n-1} V_{m+1} + I$

- **Very complex clusters**
  - $He_n C_m V_0 \leftrightarrow He_{n-1} C_{m-1} V_0 + C$

- **Extended defects**
  - $I_n \leftrightarrow I_{n-1} + I$
  - $I_n (A) \leftrightarrow I_n (B)$
  - $I_n + \text{Trap (mobile)} \leftrightarrow I_n \text{Trapped (immobile)}$

- **Damage**
  - $I_n V_m \rightarrow I_{n-1} V_{m-1}$

- **Interfaces**
  - $MP^A + A/B \leftrightarrow MP(A/B) \leftrightarrow MP^B + A/B$
  - $MP + A/B \leftrightarrow A/B$
  - $ED + A/B \leftrightarrow A/B$
  - $MC + A/B \leftrightarrow A/B$
  - $Int \rightarrow Int + I$
Flexibility in MMonCa

- **IDEA**: The **tcl** language is used to allow **flexibility**.

- User defined:
  - The material **structure** (full 3D)
  - All the clusters and extended defect **rates**
  - Reactions, using **wildcards**
  - Default parameters can be overwritten in the **input** file: **self-contained**

```tcl
array<string,string> interactions {
    He*+Hei true
    He*+V true
    He*+Gas true
    C*I*+I true
    C*V*+V true
}

proc migrate { size } {
    if { $size == 2 } { return "8.2e-3 0.42" }
    if { $size == 3 } { return "8.2e-3 0.43" }
    if { $size == 4 } { return "8.2e-3 0.43" }
    set pref [expr 3.5e-4+1.7e-3/pow($size,1.7)]
    set ener [expr 0.06+ 0.11/pow($size,1.6)]
    return "$pref $ener"
}

proc material { x y z } {
    if { $x < 0 } { return "Copper" }
    return "Niobium"
}
```
Isochrononal annealing of $\alpha$-Fe
Defects and resistivity recovery during α-Fe annealing

- Simulated isochronal annealing of 2e-4 dpa irradiated Fe.
- Excellent agreement with experiments\(^1\) and simulations.\(^2,3\)

\(^1\) Takaki et al. Radiation Effects 79, 87 (1983)
Defect population evolution during annealing of α-Fe

a) 107 K

b) 146 K

c) 186 K

d) 277 K

ignacio.martin@imdea.org
Annealing of implanted defects in silicon
Interstitial evolution in damaged Si

- Interstitial supersaturation $[I]/[I^*]$ after a 40 keV, $2 \times 10^{13}$ cm$^{-2}$ Si into Si implant annealed at different temperatures.
- Excellent agreement with experimental$^1$ results and previous simulations.$^2$

Evolution of damage in tungsten
MMonCa parametrization

<table>
<thead>
<tr>
<th>Particles</th>
<th>Clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobile</td>
<td>He, V, I</td>
</tr>
<tr>
<td>Immobile</td>
<td>C (as traps for V, I)</td>
</tr>
</tbody>
</table>

- Clusters defined by:
  - Potential energy
  - Emission prefactor
  - Migration prefactor
  - Migration energy

The particle-cluster binding energy is calculated by MmonCa

<table>
<thead>
<tr>
<th>He3</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Epot</td>
<td>-2.39 eV</td>
</tr>
<tr>
<td>Emission pref. (He3 → He2+He)</td>
<td>9.9856e-4 cm² s⁻¹</td>
</tr>
<tr>
<td>Emig</td>
<td>0.05 eV</td>
</tr>
<tr>
<td>Migration pref.</td>
<td>9.9856e-6 cm² s⁻¹</td>
</tr>
</tbody>
</table>

Two implantations:
- Sub-threshold at 400 eV (only He implantation)
- Above-threshold at 3 KeV (He and FP implantation)
- Irradiated at 5K
- We followed the Marlowe implantation carried out by Becquart et al.

Isochronal annealing for 60 s every 2K

Simulation box: 317x126x126 nm$^3$

MMonCa simulations
Under-threshold

Number of clusters vs Number of He in the clusters for different simulations.
In the simulations:

- Interstitial/interstitial clusters move in 3D
- Capture radii not depends on the object
Conclusions
Many thanks for your attention

Want to use MMonCa? Write to ignacio.martin@imdea.org