

# Análise de caso: Decomposição elétrica do $TiS_3$ em condições ambientais através do método de Monte Carlo Cinético

Filipe Gonçalves Jacinto

24 de Setembro de 2021

## Contents

<b>1</b>	<b>Introdução:</b>	<b>1</b>
<b>2</b>	<b>Metodologia:</b>	<b>2</b>
2.1	Algoritmo(KMC): . . . . .	2
2.2	Estrutura do código: . . . . .	4
2.2.1	Pré-requisitos: . . . . .	4
2.2.2	Arquivo 1: . . . . .	4
2.2.3	Arquivo 2: <code>main.py</code> . . . . .	5
<b>3</b>	<b>Resultados e Discussão:</b>	<b>6</b>

## 1 Introdução:

Desde a síntese do Grafeno [1] materiais bidimensionais(2D) vem se mostrando como promissores candidatos para substituir os semicondutores convencionais baseados em Silício à medida em que se reduz a escala de dispositivos eletrônicos. Nesse sentido, são necessárias investigações mais profundas de materiais como o do  $TiS_3$  (Trissulfeto de Titânio) que possuem algumas características como um *Gap* de 1 *eV* e pode ser isolado em monocamadas e nanotubos [2].

O artigo indicado pelo professor da disciplina de Física computacional [2] discute as propriedades de decomposição elétrica do  $TiS_3$ . Os resultados que foram obtidos de forma experimental e através de simulações computacionais sugerem que este fenômeno ocorre tanto devido a efeito Joule quanto por

dessorção de moléculas de  $SO$  da superfície de  $TiS_3$  em uma atmosfera rica em  $O_2$  o que provoca a degradação do material e a consequente decomposição elétrica do  $TiS_3$ .

No contexto da disciplina de Física Computacional, este trabalho visa entender a dinâmica de dessorção de partículas no problema físico da decomposição elétrica do  $TiS_3$  em condições ambientais utilizando a linguagem Python e o Método de Monte Carlo Cinético (KMC) discutindo os resultados obtidos e comparando-os com o do artigo em questão[2]. Além disso, deve ser também um manual básico de como o código desenvolvido funciona. Para isso, pretende-se utilizar alguns resultados já disponibilizados pelo professor como as energias de formação de vacâncias calculadas utilizando o pacote Quantum Espresso[3], e a termogravimetria disponibilizada no artigo[2] para calcular a dinâmica de dessorção do  $S$  da monocamada de  $TiS_3$  rica em  $O_2$  e compara-la com o experimento.

## 2 Metodologia:

### 2.1 Algoritmo(KMC):

Para estudar a dinâmica de dessorção do  $S$  da monocamada de  $TiS_3$  foram contruídos códigos em python utilizando a estrutura de algoritmo KMC em específico o método Bortz-Kalos-Lebowitz (BKL) que é utilizado para estudar a evolução no tempo de um sistema onde os processos podem ser estudados através de uma taxa conhecida nesse caso  $R_i$  e podem ser descritas ao longo do tempo através dos seguintes passos:

1. Fixar o o tempo inicial  $t = 0$ ;
2. Criar uma lista com todas a probabilidades  $\Delta_i$  do sistema;

Este passo é extremamente importante, pois aqui esta uma parte da Física do sistema, os eventos  $\Delta_i$  são as energias associadas a cada evento de dessorção descrito nas Tabelas 1 e 2. Através de cada  $\Delta_i$  pode-se calcular as probabilidades de ocorrência de cada evento ( $\Gamma_i$ ) através da equação 1 onde  $\Gamma_i^0 = 10^{13} s^{-1}$  e é a frequência de ocorrência de cada evento.

$$\Gamma_i = \Gamma_i^0 \exp\left(\frac{-\Delta_i}{K_b T}\right) \quad (1)$$

3. Calcular a função acumulativa  $R_n$  que é a soma da probabilidade acumulativa dos eventos que pode ser feita de acordo com a Equação 2

Evento	Sistema	Defeito	Energia (eV)
1	$TiS_3O_{0.50}$	Defeito de $SO$ (primeiro)	1.23
2	$TiS_3O_{0.50}$	Defeito de $SO$ + monovacância	1.55
3	$TiS_3O_{0.50}$	Defeito de $SO$ + monovacância	1.63
4	$TiS_3O_{0.50}$	Defeito de $SO$ (divacância)	2.61

Table 1: Tabela com a respectiva energia de ativação de cada evento para uma superfície com 50% de Oxigênio.

Evento	Sistema	Defeito	Energia (eV)
1	$TiS_3O_{0.25}$	Defeito de $SO$ (primeiro)	1.23
2	$TiS_3O_{0.25}$	Defeito de $SO$ + monovacância	1.51
3	$TiS_3O_{0.25}$	Defeito de $SO$ + monovacância	1.78
4	$TiS_3O_{0.25}$	Defeito de $SO$ (divacância)	2.62

Table 2: Tabela com a respectiva energia de ativação de cada evento para uma superfície com 25% de Oxigênio.

onde  $R_i$  é a probabilidade de cada evento,  $N_i$  número de partículas que podem realizar o  $i$ -ésimo evento e  $n_e$  o número total de eventos considerados possíveis;

$$R_n = \sum_{i=1}^{n_e} R_i = \sum_{i=1}^{n_e} \Gamma_i N_i \quad (2)$$

4. Gerar um número aleatório em relação a uma distribuição uniforme entre  $u \in (0, 1]$ ;
5. Encontrar qual será a transição que ocorrerá, ou seja, verificar qual termo da lista de probabilidade  $R_i$  que respeita a condição da inequação 3:

$$R_{i-1} < uR_n < R_i \quad (3)$$

6. Realizar a transição. Este passo depende muito do sistema em questão, nesse caso em específico fazer a transição significa levar em conta a mudança na lista de partículas dos eventos possíveis e no caso da variação de temperatura calcular novamente os valores de  $\Gamma$ . Além disso, para calcular a massa perdida leva-se em conta que um evento para uma monovacância isto é o número de átomos diminui em um único átomo, já para o um divacância o número total de partículas será reduzido em dois.

7. Gerar um novo número aleatório em relação a uma distribuição uniforme entre  $u \in (0, 1]$ ;
8. Realizar uma translação temporal  $t \rightarrow t + \Delta t$  onde  $\Delta t$  é dada pela equação 4:
 
$$\Delta t = R^{-1} \ln(1/u) \quad (4)$$
9. Atualizar a os valores de  $R_i$  a partir das transição realizada e recalcular o valor de  $R_n$ ;
10. Voltar ao passo 2.

## 2.2 Estrutura do código:

O código é estruturado em dois arquivos o `funcoes.py` e `main.py`. O arquivo de `funcoes.py` é onde estão escritas os passos do método de Monte Carlo na forma de funções e o arquivo `main.py` é o algoritmo estruturado na ordem correta com as funções previamente escritas no arquivo `funcoes.py`.

### 2.2.1 Pré-requisitos:

Para este código foi utilizado a linguagem Python 3.9.7, e os seguintes pacotes devem ser instalados com um gerenciador de pacotes de sua escolha, neste caso foi utilizado o pip. Para instalar os pacotes basta utilizar os comandos abaixo:

- `pip install matplotlib`
- `pip install numpy`
- `pip install scipy`
- `pip install random`

### 2.2.2 Arquivo 1:

As funções necessárias para implmentar o método de KMC estão escritas no `funcoes.py` para que possam ser chamadas no arquivo principal `main.py`. Por isso, se faz necessário que este arquivo esteja sempre no mesmo diretório que o da função principal.

1. **Gamma:**
  - Faz o cálculo de `Gamma[i]` ;

- Soma `Gamma[i]` para obter o total `Gamma`.
2. `R_sum`:
    - Calcula `R_i[i]`;
    - Soma os termos de `R_i[i]` para obter `R_n`.
  3. `Probable_event`:
    - Define qual foi o evento escolhido;
    - Realiza a transição.
  4. `Time_foward`:
    - Calcula o `time_step` (intervalo do evento) e o adiciona ao contador de tempo(`t`).
  5. `Temp_foward`:
    - Calcula o tempo transcorrido entre duas iterações consecutivas do método de KMC;
    - Em seguida calcula o `temp_step` proporcional a diferença de tempo transcorrida entre os passos dois passos consecutivos do método de KMC;
    - Adiciona o `temp_step` ao contador de temperatura(`T`).

### 2.2.3 Arquivo 2: `main.py`

Este código é dividido em três partes sendo estas:

1. A primeira parte declara as variáveis e arrays necessários para esse algoritmo, em específico:
  - **Variáveis**
    - `gamma_zero`;
    - `temperatura`;
    - `t` (tempo).
  - **Arrays**
    - `Delta` (energias de ativação)
    - `N`(número de particulas que podem realizar o evento )
  - **Empty Arrays**

- **time** (armazena os tempos calculados em cada iteração)
  - **rate** (armazena a soma **R\_n** ao longo do tempo)
2. Esta parte é onde está estruturado a ordem do algoritmo utilizando as funções definidas em **funcoes.py**. O algoritmo segue a seguinte forma:
- (a) For in range(numero de iterações necessárias)
- Lista de eventos dentro do for:
    - i. **Gamma()** ;
    - ii. **R\_sum()** ;
    - iii. **Probable\_event()**;
    - iv. **Time Foward**;
    - v. **Temp Foward**, no caso em que se varia a temperatura;
    - vi. Adicionamos o termo no array de **time**
    - vii. Adicionamos o termo no array de **mass**
3. A parte final do algoritmo consiste em plotar/calcular quantidades que ajudem a compreender o problema. Como por exemplo o plot de gráfico utilizando o pacote matplotlib.

### 3 Resultados e Discussão:

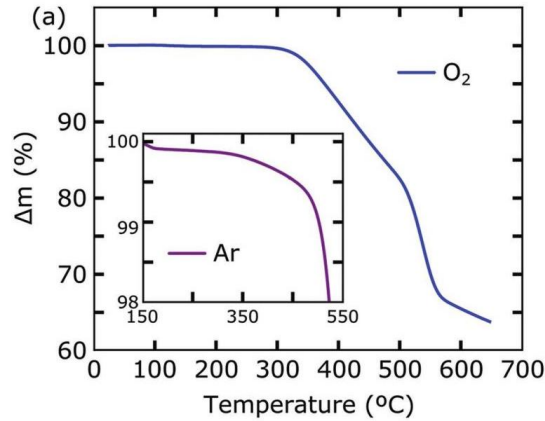


Figure 1: Curva de termogravimetria para o bulk de  $\text{TiS}_3$ .

De forma a estudar o papel da perda de massa do  $\text{TiS}_3$  no fenômeno de decomposição elétrica foram analisados resultados de alguns experimentos

sendo um desses o de termogravimetria. A Figura 1 apresenta a curva de termogravimetria sob efeito de uma atmosfera rica em  $O_2$  obtida experimentalmente[2]. Observando a Figura 1, pode-se ver uma variação de massa de 36% em um intervalo de temperatura de  $27^\circ C$  até  $650^\circ C$ . Nesse intervalo de temperatura, pode-se notar dois eventos a citar: (1) no intervalo de  $300 - 400^\circ C$  com uma variação de massa de 18% e (2) no intervalo de  $450 - 550^\circ C$  com uma variação de massa também de 18%.

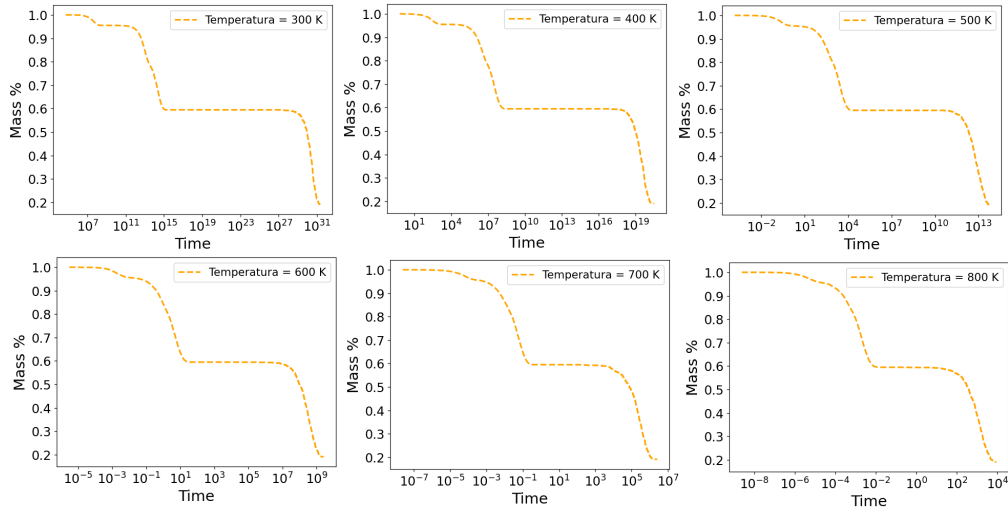


Figure 2: Curva da variação de massa em relação ao tempo obtida através do método de monte carlo cinético para uma superfície com 50% de  $O_2$ .

Em seguida, para investigar a dinâmica do sistema de dessorção de átomos de  $S$  do  $TiS_3$  foi aplicado o método de Monte Carlo cinético calculando primeiro a variação de massa em porcentagem ao longo do tempo mantida uma temperatura constante. Este resultado é apresentado nas Figuras 2 e 3 onde o modelo considera que o sistema é uma monocamda de  $TiS_3$  exposta a uma Atmosfera com 50% e 25% de concentração de  $O_2$  respectivamente. Pode-se ver pelas Figuras 2 e 3 que a perda de massa é mais rápida quanto maior for a temperatura considerada para os cálculos.

Além disso, as energias de ativação de cada evento tem participação importante na dinâmica do sistema já que energias maiores terão uma probabilidade menor de acontecer e portanto em geral ocorrerão com maior frequência depois que os eventos de menor energia se tornarem menos prováveis devido por exemplo a diminuição do número de partículas disponíveis para que os eventos de menor energia ocorram. Este comportamento se torna claro

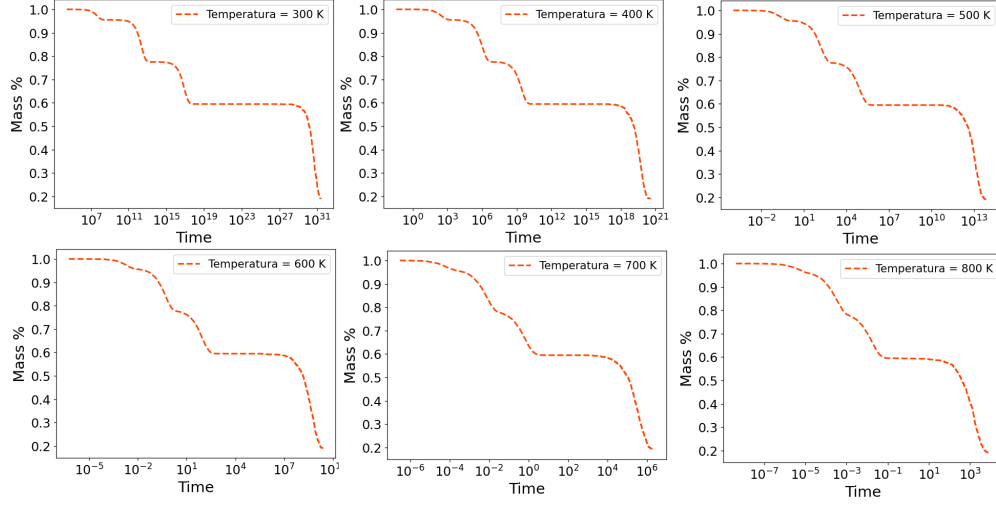


Figure 3: Curva da variação de massa em relação ao tempo obtida através do método de monte carlo cinético para uma superfície com 25% de  $O_2$ .

quando se analisa a Figura 3 onde a diferença de energia entre cada evento é maior e observa-se um número maior de degraus no gráfico. Estes degraus denotam quando cada evento ocorre com maior frequência sendo que os de menor energia ocorrem sempre primeiro.

Finalmente, de maneira a comparar os resultados obtidos na curva de termogravimetria apresentada na Figura 1, o método de KMC foi utilizado variando a temperatura em escalas de tempo diferentes. Este resultado é apresentado na Figura 4. Pode-se ver na Figura 4 assim como na Figura 1, a ocorrência de dois eventos, embora em diferentes intervalos de temperatura. O evento (1) ocorre no intervalo de  $150 - 250^\circ C$  com uma variação de massa de 17.5% que se deve principalmente por formações de monovacâncias e o evento (2) ocorre no intervalo de  $500 - 600^\circ C$  com uma variação de massa de 21.5% que por sua vez se deve principalmente a formação de divacâncias.

Portanto, o método de KMC concorda razoavelmente bem com a curva de termogravimetria. Este resultado permite concluir que a formação de monovacância e divacâncias podem provocar a degradação do material e consequentemente ocasionar no fenômeno de decomposição elétrica do  $TiS_3$ .



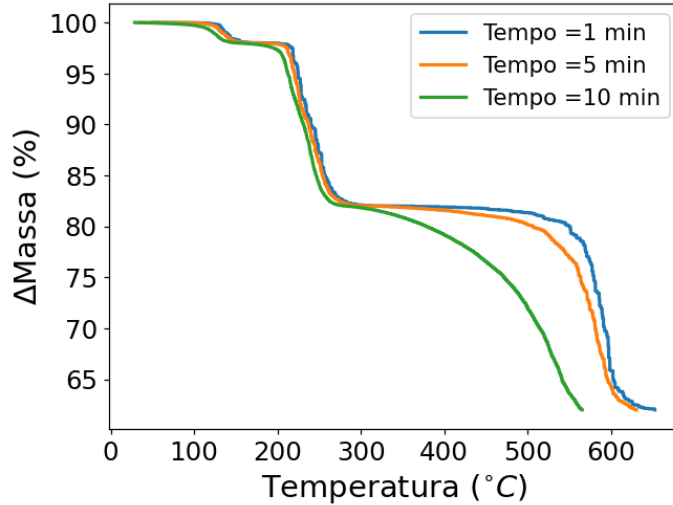


Figure 4: Curva da variação de massa em relação a temperatura para diferentes intervalos de tempo descritos na legenda, em uma superfície com 50% de  $O_2$ .

## References

- [1] A. K. Geim and K. S. Novoselov, “The rise of graphene,” in *Nanoscience and technology: a collection of reviews from nature journals*, pp. 11–19, World Scientific, 2010.
- [2] A. J. Molina-Mendoza, J. O. Island, W. S. Paz, J. M. Clamagirand, J. R. Ares, E. Flores, F. Leardini, C. Sánchez, N. Agraït, G. Rubio-Bollinger, *et al.*, “High current density electrical breakdown of *tis3* nanoribbon-based field-effect transistors,” *Advanced Functional Materials*, vol. 27, no. 13, p. 1605647, 2017.
- [3] P. Giannozzi, S. Baroni, N. Bonini, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, G. L. Chiarotti, M. Cococcioni, I. Dabo, *et al.*, “Quantum espresso: a modular and open-source software project for quantum simulations of materials,” *Journal of physics: Condensed matter*, vol. 21, no. 39, p. 395502, 2009.