

Decomposição elétrica do TiS3 em condições ambientais

Filipe Gonçalves Jacinto

September 24, 2021

Contents

1	Introdução:	1
1.1	Pontos positivos do TiS3:	1
1.2	Problema do paper: Calcular a decomposição elétrica do TiS3	1
1.2.1	Resultados:	2
2	Estrutura do código:	2
2.1	Funcoes.py	2
2.1.1	Gamma:	2
2.1.2	R _{sum} :	2
2.1.3	Probable _{event} :	2
2.1.4	Time _{foward} :	2
2.2	Main.py:	2

1 Introdução:

1.1 Pontos positivos do TiS3:

- Possível substituto do Silício
- Bandgap de 1.0 eV
- Pode ser isolado em monocamada e nanotubos

1.2 Problema do paper: Calcular a decomposição elétrica do TiS3

1.2.1 Resultados:

- A decomposição elétrica é causada tanto por efeito Joule
- A decomposição elétrica também é causada por formação de vacâncias , mesmo que a energia de ativação da reação seja alta para um único átomo de S
- O teste com atmosfera rica em Oxigênio mostra que a energia de formação de defeitos cai significativamente quando se retira uma dupla de átomos SO comparado a um átomo de S sozinho
- DFT + Monte Carlo cinético sugerem que a formação de vacâncias é devida à oxidação do material seguida da dessorção dos átomos de enxofre

2 Estrutura do código:

2.1 Funcoes.py

2.1.1 Gamma:

- Faz o cálculo de Γ_i cálculo de Gamma
- Γ

2.1.2 R_{sum} :

- Calcula $R_i[i]$
- Soma os termos de $R_i[i]$

2.1.3 Probable_{event}:

2.1.4 Time_{forward}:

- Calcula o $time_{step}$ (intervalo do evento) e o adiciona ao contador de tempo(t)

2.2 Main.py:

- Este código é dividido em três partes:
 1. A primeira parte declara as variáveis e arrays necessários nesse algoritmo em específicos:

- **Variáveis**
 - * γ_{zero}
 - * temperatura
 - * t (tempo)
 - **Arrays**
 - * Delta (energias de ativação)
 - * N (número de partículas que realizaram 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) **Gamma()** , calcula gamma
 - (b) For in range(numero de iterações necessárias)
 - * Lista de eventos dentro do for:
 - i. **R_{sum}()**
 - ii. **Probable Event()**
 - iii. **Time Foward**
 - iv. Adicionamos o termo no array de **time**
 - v. Adicionamos o termo no array de **rate**
3. A parte final do algoritmo consiste em plotar/calcular quantidades que ajudem a compreender o problema :
- (a) Plot de gráfico utilizando o pacote matplotlib