

Проект 2: Электрический пробой.

Этап 2. Алгоритмы решения задачи

Кадров Виктор Максимович and Туем Гислен and Туем Гислен

Содержание

1	Содержание	5
1.1	Постановка задачи	5
1.2	Методы численного решения	5
1.2.1	Сравнительный анализ	5
1.3	Алгоритм метода конечных разностей (FDM)	6
1.3.1	Шаги реализации:	6
1.4	Моделирование лавинного пробоя	6
1.5	Визуализация результатов	7
1.6	Практическая реализация	8
2	Заключение	9

Список иллюстраций

Список таблиц

1 Содержание

1. Постановка задачи
 2. Методы численного решения
 3. Алгоритм FDM
 4. Моделирование лавинного пробоя
 5. Визуализация результатов
 6. Практическая реализация
-

1.1 Постановка задачи

Цель: Расчет критического напряжения пробоя V_{br}

Уравнения:

$$\nabla^2 \phi = -\frac{\rho}{\varepsilon} \quad (\text{Уравнение Пуассона})$$

$$\alpha(E) = A \cdot e^{-B/|E|} \quad (\text{Коэффициент ионизации Таунсенда})$$

Критерий пробоя:

$$\int_0^d \alpha(E) dx \geq 1$$

1.2 Методы численного решения

1.2.1 Сравнительный анализ

Метод	Преимущества	Недостатки
FDM	Простота реализации	Ошибки на сложных сетках
FEM	Высокая точность	Вычислительно затратен
Монте-Карло	Учет стохастических эффектов	Медленная сходимость

1.3 Алгоритм метода конечных разностей (FDM)

1.3.1 Шаги реализации:

1. Дискретизация:

$$\frac{\partial^2 \phi}{\partial x^2} \approx \frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{h^2}$$

2. Итерационное решение:

```
for n in range(max_iter):
    phi[1:-1,1:-1] = 0.25*(phi[2:,1:-1] + phi[:-2,1:-1]
                        + phi[1:-1,2:] + phi[1:-1,:-2])
```

3. Расчет поля:

$$E_x = -\frac{\phi_{i+1,j} - \phi_{i-1,j}}{2h}$$

1.4 Моделирование лавинного пробоя

Алгоритм Монте-Карло:

1. Генерация начальных электронов

2. Движение в электрическом поле:

$$\Delta x = \mu E \Delta t + \mathcal{N}(0, D \Delta t)$$

3. Проверка условий ионизации

4. Учет вторичной эмиссии

graph TD

A[Старт] --> B[Генерация электронов]

B --> C[Дрейф в поле E]

C --> D{Ионизация?}

D -->|Да| E[Новые электроны]

D -->|Нет| F[Поглощение]

E --> C

1.5 Визуализация результатов

Пример кода для Python:

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

```
x = np.linspace(0, 10, 100)
```

```
E = np.sin(x) # Пример распределения поля
```

```
plt.figure(figsize=(8,4))
```

```
plt.plot(x, E, label='Напряженность поля')
```

```
plt.xlabel('Положение, мкм')
```

```
plt.ylabel('E, В/мкм')
```

```
plt.axhline(y=3, color='r', linestyle='--', label='Пробой')  
plt.legend()  
plt.show()
```

1.6 Практическая реализация

Рекомендуемый стек технологий:

- Языки: Python (NumPy, SciPy), C++
- Визуализация: Matplotlib, ParaView
- Параллельные вычисления: MPI, CUDA

Этапы проекта:

1. Реализация решателя Пуассона
 2. Валидация на аналитических решениях
 3. Моделирование пробоя
 4. Оптимизация параметров
-

2 Заключение

Ключевые результаты: 1. Разработан алгоритм FDM для расчета полей 2. Реализована модель лавинного пробоя 3. Получены зависимости V_{br} от параметров

Перспективы: - Учет тепловых эффектов - Гибридные схемы (FDM + Монте-Карло)