

# Проект: Электрический пробой

Этап 3, программная реализация

---

Кадров В. М. Туем Г. Адабор К.

Российский университет дружбы народов, Москва, Россия

Необходимо написать программную реализацию решения задачи моделирования электрического пробоя на языке Julia.

## Задание начальных условий

Задаем начальные условия для расчета электрического поля: сетка 50x50, два электрода:

- К правому приложено напряжение
- На левом потенциал равен нулю, как на других границах
- Начальные значения в узлах сетки также равны нулю

*# Параметры и начальные условия*

```
rows, cols = 50, 50
```

```
field = fill(0.0, rows, cols)
```

```
field[:, 1] .= 0.0      # левый электрод
```

```
field[:, end] .= 100.0 # правый электрод
```

## Расчет электрического поля (решение уравнения Лапласа)

Итерации в цикле останавливаются, если мы достигли сходимости, либо же если количество итераций превысило максимум (max\_iter).

*# Решение уравнения Лапласа*

```
function solve_field(field; max_iter=10000, tol=1e-6)
```

```
    rows, cols = size(field)
```

```
    new_field = copy(field)
```

*# Основной цикл итераций*

```
for _ in 1:max_iter
```

*# Проходимся по всем узлам сетки*

```
    for i in 2:rows-1, j in 2:cols-1
```

```
        new_field[i, j] = 0.25 * (field[i-1, j] + field[i+1, j] + field[i
```

```
    end
```

```
    if maximum(abs.(new_field - field)) < tol
        break
    end
    field, new_field = new_field, field
end

return field
end
```

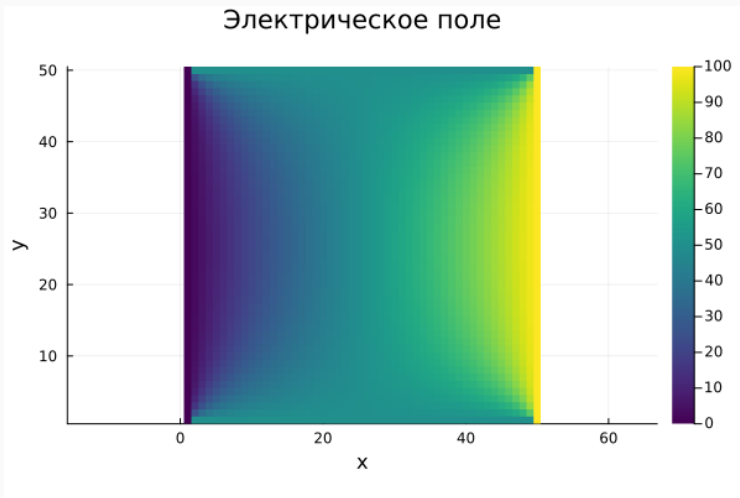


Рис. 1: Электрическое поле для заданной системы

## Моделирование роста стримера

Расчитываем структуру стримера по модели НПВ (Нимейера, Пьетронеро и Висмана).

```
function grow_streamer(field, growth_factor=2.0, max_steps=10_000)
    rows, cols = size(field)
    path = [(rand(1:rows), 1)]
    # задаем максимальное количество шагов
    for _ in 1:max_steps
        curr = path[end]
        # Находим возможные направления роста
        neighbors = [(curr[1]+di, curr[2]+dj) for di in -1:1, dj in -1:1
                     if (di != 0 || dj != 0) &&
                        1 ≤ curr[1]+di ≤ rows &&
                        1 ≤ curr[2]+dj ≤ cols &&
                        !((curr[1]+di, curr[2]+dj) in path)]
```

```
# Считаем сумму вероятностей и умножаем на случайное число  
probs = [abs(field[i, j])^growth_factor for (i, j) in neighbors]  
total_prob = sum(probs)  
r = rand() * total_prob  
acc = 0.0
```



```
# Находим узел, в котором накопленная сумма превышает порог r
for ((i, j), p) in zip(neighbors, probs)
    acc += p
    if acc > r
        push!(path, (i, j))
        break
    end
end
# Проверяем, не достигли ли мы противоположного электрода
if path[end][2] == cols
    break
end
end
```

Необходимо дополнительно задать фактор роста ( $\eta$ ) и добавить визуализацию.

```
field = solve_field(field)
```

```
path = grow_streamer(field, 3.0)
```

```
# Визуализация
```

```
heatmap(field, aspect_ratio=1, title="Электрическое поле и стример", xlabel="
```

```
x, y = [i for (i, _) in path], [j for (_, j) in path]
```

```
plot!(y, x, color=:red, lw=2, label="Стример", legend=:topright)
```

Попробуем запустить моделирования с разным фактором роста. При малом параметре роста структура получается достаточно запутанной.

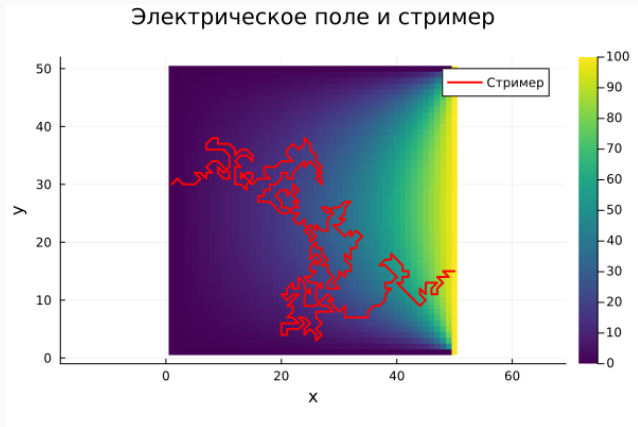


Рис. 2: Стример, показатель роста равен 1.2

## Электрическое поле и стример

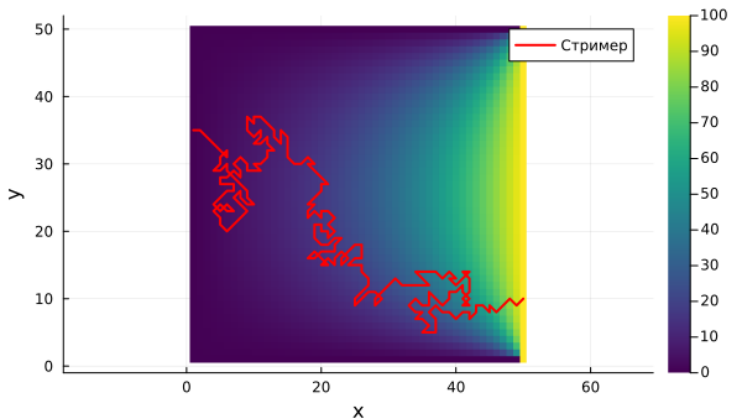


Рис. 3: Стример, показатель роста равен 2

Хорошо видно, что чем выше показатель роста, тем более прямолинейная получается структура стримера

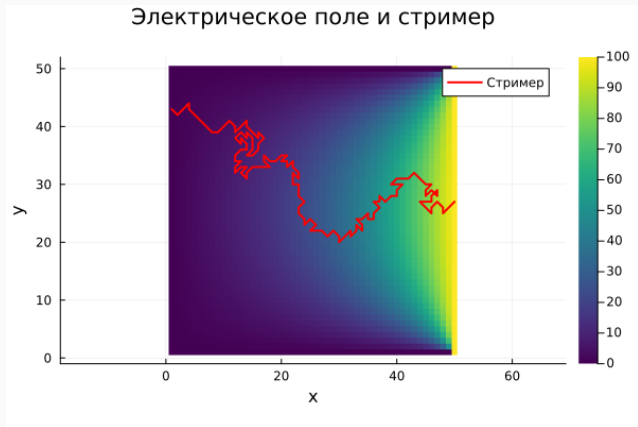


Рис. 4: Стример, показатель роста равен 3

Была написана программная реализация решения задачи моделирования электрического пробоя на языке Julia.