

Моделирование электрического пробоя

Проектная работа. Этап №3

Евдокимов И.А., НФИбд-01-20 Евдокимов М.М., НФИбд-01-20 Манаева В.Е., НФИбд-01-20 Покрас
И.М., НФИбд-02-20 Сулицкий Б.Р., НФИбд-02-20 Новосельцев Д.С., НФИбд-02-20

17 марта 2023

Российский университет дружбы народов, Москва, Россия

Описание программной реализации проекта.

Задачи третьего этапа проекта:

- Написать программу, моделирующую электрический пробой по модели;
- Разобрать применение заявленных в предыдущих этапах формул;
- Продемонстрировать результаты реализации;

Написание программы

Алгоритм генерации молний основан на нескольких следующих формулах:

$$\vec{E}_i = k \frac{q_i}{\epsilon r_i^2}$$

Напряжённость, создаваемая i -тым зарядом
в точке

$$\vec{E} = \sum_0^n \vec{E}_i$$

Общая напряжённость электрического поля
в точке

- Генерация зарядов в воздушном пространстве;
- Подсчёт напряжённости поля в конкретных точках;
- Создание поля;
- Генерация молнии в поле.

Код

```
function z()  
    if rand(0:10) > 6  
        t = rand(-10:10)  
    else  
        t = 0  
    end  
    return t  
end
```

Цикл подсчёта напряжённости в конкретной точке пространства

```
for c in condition
  if (c[1] != b) || (c[2] != a)
    t += c[3] / ((c[1]-b)*(c[1]-b) + (c[2]-a)*(c[2]-a))
  end
```


Цикл создания молний в пространстве

```
# молнии
for e in 1:count
  for _ in 1:limit
    t:Array{Any} = [0, 0, -1]
    for g in minA:maxA
      temp = [abs(round(Int, (last(lightnings[e][1]) + last(lightnings[e][3]) * cos(g * pi))))),
              abs(round(Int, (last(lightnings[e][2]) + last(lightnings[e][3]) * sin(g * pi))))]
      if (1 < temp[1] < x) && (1 < temp[2] < y)
        if (temp[2] < last(lightnings[e][2]))
          if (last(lightnings[e][2]) < last(lightnings[e][3])) && (space[temp[1]][temp[2]] > 10)
            t = [last(lightnings[e][1]), 1, space[temp[1]][temp[2]]]
            break
          elseif (t[3] < space[temp[1]][temp[2]]) || (space[temp[1]][temp[2]] == 100)
            t = [temp[1], temp[2], space[temp[1]][temp[2]]]
          end
        end
      end
    end
  end
  if (last(lightnings[e][3]) >= last(lightnings[e][2]))
    push!(lightnings[e][1], last(lightnings[e][1]))
    push!(lightnings[e][2], 1)
    push!(lightnings[e][3], round(last(lightnings[e][3]) * losses, digits=3))
    break
  elseif t[3] != -1
    push!(lightnings[e][1], t[1])
    push!(lightnings[e][2], t[2])
    push!(lightnings[e][3], round(last(lightnings[e][3]) * losses, digits=3))
  else
    push!(lightnings[e][1], last(lightnings[e][1])+rand(-1:1))
    push!(lightnings[e][2], last(lightnings[e][2])-1)
    push!(lightnings[e][3], round(last(lightnings[e][3]) * losses, digits=3))
  end
end
end
```

Рис. 1: Код создания молний

Результаты работы кода
(изображения)

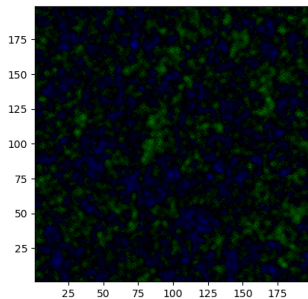


Рис. 2: Пример генерации пространства

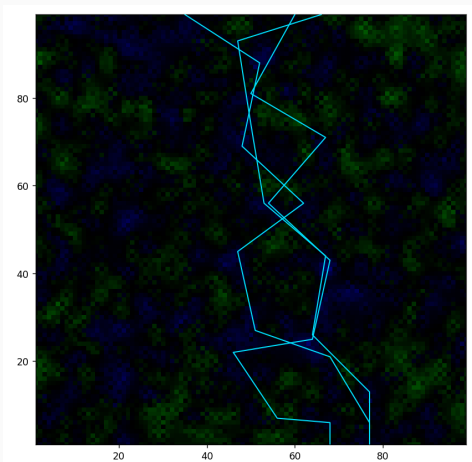


Рис. 3: Молния, пример 1

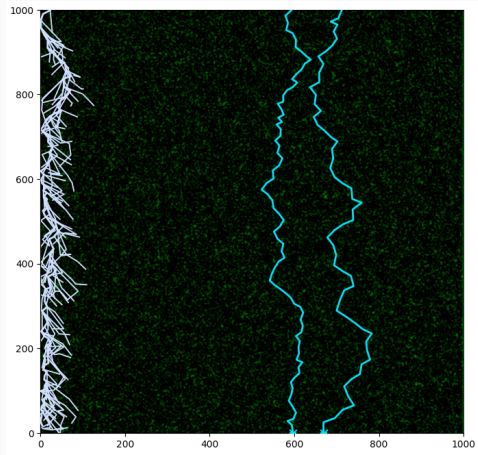


Рис. 4: Молния, пример 2

Молнии не бьют в одно место дважды. По нашему коду.

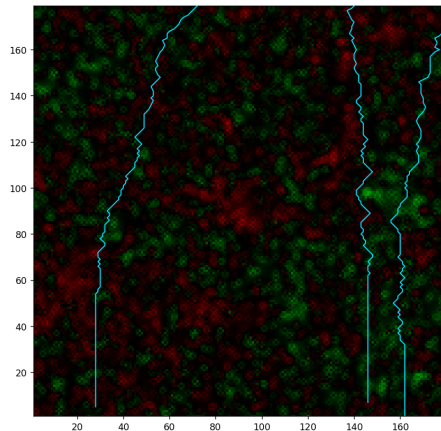


Рис. 5: Немножечко молний из промежуточных вариантов кода