Отчёт об этапе проекта №3

Модель электрического пробоя

Евдокимов М.М., НФИбд-01-20

Евдокимов И.А., 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

Содержание

# 1 Текст

## 1.1 Этап №3. Проект №2

Тема проекта: электрический пробой.

Задание второго этапа проектной работы: .

# 2 Написание программы

## 2.1 Краткая справка об алгоритме

Алгоритм генерации молний основан на нескольких следующих формулах:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
| Сила, оказываемая зарядом электрического поля () на движущийся заряд () | Напряжённость, создающаяся -тым зарядом в поле | Общая напряжённость электрического поля |
|  |  |  |
| - |  | - |

## 2.2 Решённые задачи

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

# 3 Код

## 3.1 Общий

using PyCall  
using Random  
@pyimport PIL.Image as img  
@pyimport matplotlib.pyplot as plt  
  
function z()  
 if rand(0:10) > 8  
 t = rand(-10:10)  
 else  
 t = 0  
 end  
 return t  
end  
  
x, y, limit = Int64(100), Int64(100), 80  
minA, maxA = 180, 360  
minB, maxB = 160, 380  
minE, maxE, count = 18.0, 21.0, 3  
losses, pL, brench\_count = 0.97, pi / 180, 3  
colorL = ["#00e0fe", "#d0e0ff"]  
condition = []  
for c1 = 0:x, c2 = 0:y  
 if rand(0:10) > 9  
 push!(condition, [c1, c2, rand(-10:10)])  
 end  
 c1 = 0  
end  
space = [[0.0 for \_ in 0:x] for \_ in 0:y]  
lightnings = [[[rand(1:x-1)], [y-1], [rand(minE:.001:maxE)]] for \_ in 0:count]  
branches = []  
image = img.new(mode="RGB", size=(y, x))  
for a in 1:y  
 println(a)  
 for b in 1:x  
 t = 0  
 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  
 space[a][b] = round(t; digits = 3)  
 if space[a][b] >= 0  
 image.putpixel((a-1, b-1), (0, 0, round(Int, space[a][b]\*2)))  
 else  
 image.putpixel((a-1, b-1), (0, round(Int, abs(space[a][b])\*2), 0))  
 end  
 end  
end  
for e in 1:count  
 for \_ in 1:limit  
 t = [-1, -1, -1]  
 for g in minA:maxA  
 temp = [abs(round(Int, (last(l[e][0]) + last(l[e][2]) \* cos(g \* pL)))),  
 abs(round(Int, (last(l[e][1]) + last(l[e][2]) \* sin(g \* pL))))]  
 if (1 < temp[1] < x) && (1 < temp[2] < y)  
 if (temp[2] < last(l[e][2])) && (1 < temp[1] < x)  
 if (last(l[e][2]) < last(l[e][3])) && (space[temp[1]][temp[2]] > 10)  
 t = [last(l[e][1]), 0, 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  
end  
  
for l in lightnings  
 ax.plot(light[1], light[2], color=color[1], linewidth=1.2, zorder=3)  
end  
fig, ax = plt.subplots()  
ax.set\_xlim(0, x)  
ax.set\_ylim(0, y)  
plt.imshow(image)  
plt.show()

## 3.2 Функция генерации заряда в точке

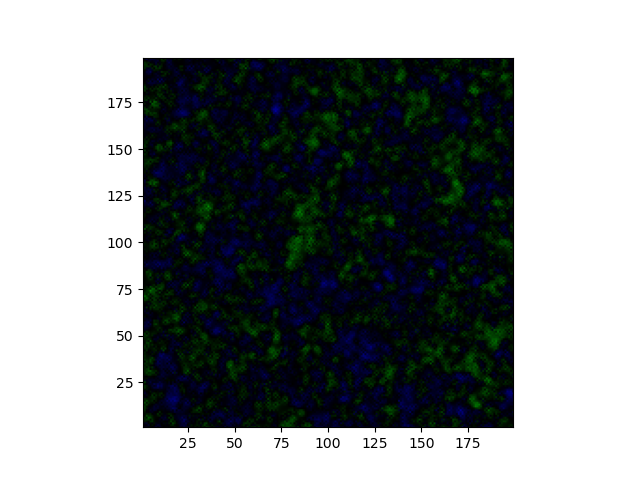
function z()  
 if rand(0:10) > 8  
 t = rand(-10:10)  
 else  
 t = 0  
 end  
 return t  
end

## 3.3 Функция подсчёта напряжённости в точке

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

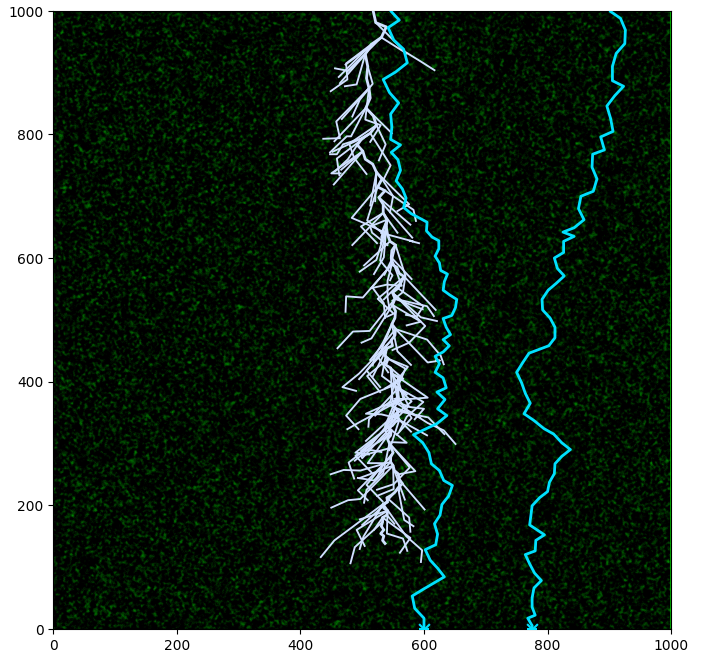
# 4 Результаты работы кода (изображения)

## 4.1 Пространство



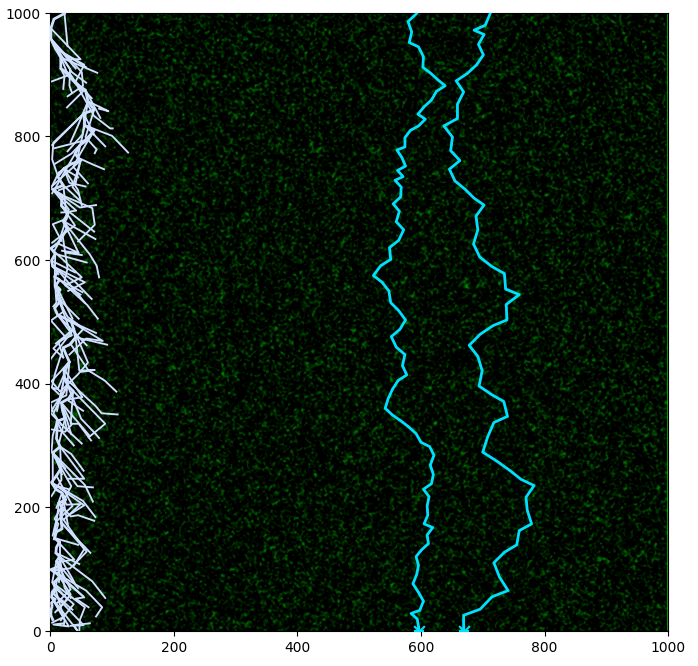
Пример генерации пространства

## 4.2 Молнии в пространстве (1)



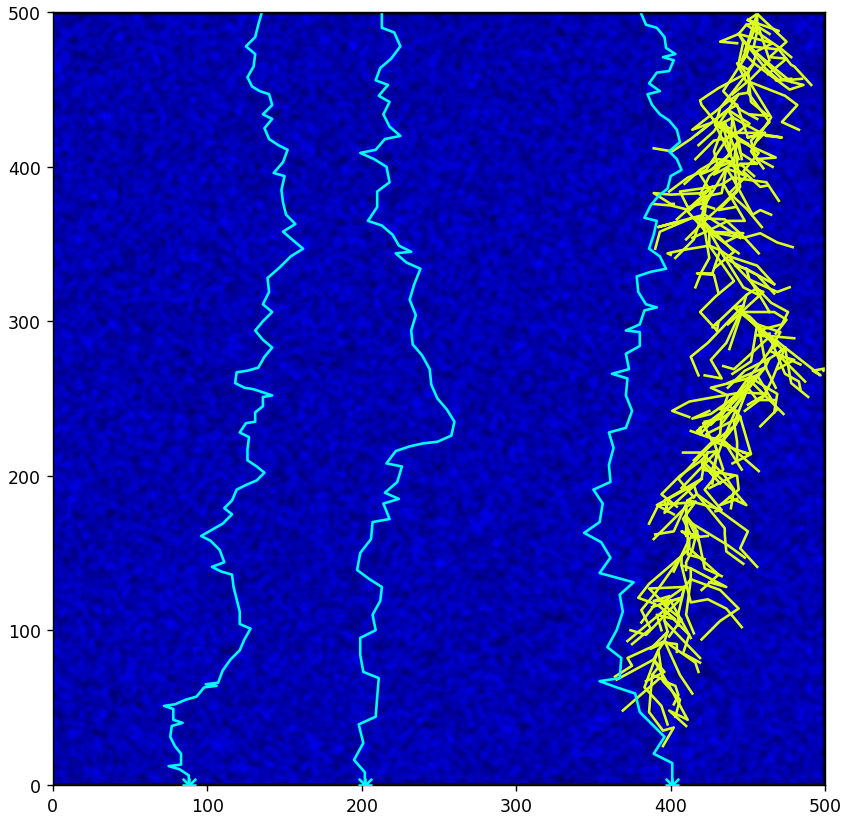
Молния, пример 1

## 4.3 Молнии в пространстве (2)



Молния, пример 2

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



Немножечко молний из промежуточных вариантов кода