| Politechnika Warszawska, Wydział Elektryczny | | |
|---|---|---|
| **Matematyka Metody Numeryczne** | | |
| **Semestr: II** | **Kierunek: Elektromobilność** | |
| Temat ćwiczenia:<br>2_Obliczanie powierzchni pola metodą Montecarlo | | Prowadzący: dr inż. Tomasz Leś |
| Student: Bernard Kościewicz | | Nr albumu: 318993 |
| **Data wykonania ćwiczenia:<br>18.03.2022** | **Data oddania sprawozdania:<br>25.03.2022** | **Ocena / punkty:** |

1. Funkcja sprawdzająca przecinanie się linii

   Aby określić po której stronie prostej przechodzącej przez A, B znajduje się punkt C, sprawdzam iloczyny wektorowe S_1 = (C - A) x (B - A), S_2 = (D - A) x (B - A).

   Sprawdzenie, czy A, B przecina prostą przechodząca C, D; S_3 = (A - C) x (B - C) S_4 = (B - C) x (D - B)

   W tym celu potrzebuję dwie funkcje pomocnicze:

   a. Określenie funkcji iloczynu wektorowego

```
function [arg] = vect_mltp(A, B, C)
arg = (B(1, 1) - A(1, 1)) * (C(1, 2) - A(1, 2)) - (C(1, 1) - A(1, 1)) * (B(1, 2) - A(1, 2));
%arg = (B(2, 2) - A(1, 1)) * (C(2, 2) - A(1, 1)) - (C(2, 2) - A(1, 1)) * (B(2, 1) - A(1, 1));
end
```

   b. Określenie funkcji usytuowania punkty między dwoma innymi

```
function [arg] = among(A, B, C)
if min(A(1, 1), B(1, 1)) <= C(1, 1) && C(1, 1) <= max(A(1, 1), B(1, 1))
   arg = 1;
else
   arg = 0;
end
end
```

2. Funkcja lokalizująca punkt w stosunku do powierzchni pola

```
function [arg] = point_location(single_point, multi_points, points_nr)
result = 0;
edge = [0, 0];
for i = 1:points_nr
   point_compare = i + 1;
   if i == points_nr
       point_compare = 1;
   end
   if lines_cross(single_point, edge, multi_points(i, :), multi_points(point_compare, :)) == 1
       result = result + 1;
```

```
    end
end
if mod(result, 2) == 0
    arg = 0;
else
    arg = 1;
end
end
```

3. Funkcja licząca powierzchnie oparta na metodzie Monte Carlo

```
function [area] = func_monte_carlo(points, ~, points_rand)
x = points(:,1)';
y = points(:,2)';
%points(points_nr + 1, 1) = points(1, 1);
%points(points_nr + 1, 2) = points(1, 2);
%shape = animatedline(x, y, 'Color', 'g', 'LineWidth', 3);
outside = animatedline("Color", "b", "Marker",".","MarkerSize", 4,
"LineStyle", "-");
inside = animatedline("Color", "r", "Marker", ".", "MarkerSize", 4,
"LineStyle", "--");
%odświeżanie kształtu i wywołaj zwrotnie (typu: return)
drawnow;
min_x = min(x);
max_y = max(y);
max_x = max(x);
min_y = min(y);
%aa = [min_x, min_x, max_x, max_x, min_x];
%bb = [min_y, max_y, max_y, min_y, min_y];
%sqr = animatedline(aa, bb, "Color", "k", "LineWidth", 2);
hit_in = 0;
hit_out = 0;

for i = 1:points_rand
    addpoints(outside, i, (hit_in/(hit_in+hit_out)) * (max_y - min_y) * (max_x
- min_x));
    drawnow;
    random_x = (max_x-min_x).*rand()+min_x;
    random_y = (max_y-min_y).*rand()+min_y;
    [In_area, On_range] = inpolygon(random_x,random_y, x, y);

    X_in = numel(random_x(In_area)); %Współrzędna w polu
    Y_in = numel(random_y(In_area));
    X_on = numel(random_x(On_range)); %Współrzędne na krawędzi
    Y_on = numel(random_x(On_range));

    if  (X_in == 1 && Y_in == 1) || (X_on == 1 && Y_on == 1)
        %point_location([random_x, random_y], points, points_nr) == 1;
        hit_in = hit_in + 1;
        %addpoints(hit_in, random_x, random_y);
        %drawnow;
    else
        hit_out = hit_out + 1;
        %addpoints(hit_out, random_x, random_y);
        %drawnow;
    end
end
area = (hit_in/(hit_in+hit_out)) * (max_y - min_y) * (max_x - min_x);
```

```
addpoints(inside,[0, points_rand], [area, area]);
drawnow;
end
```

4. Wywołanie funkcji i sprawdzenie

```
points_nr= input("Proszę wprowadzić liczbe określającą kształt: ");
points_casual = input("Proszę wprowadzić liczbe punktów oscylujących przy
kształcie: ");
for i = 1:points_nr
   multi_points(i, 1) = input("Proszę podaj x: ");
   multi_points(i, 2) = input("Proszę podaj y: ");
end
display(func_monte_carlo(multi_points, points_nr, points_casual));
```

   a. Output

```
Proszę wprowadzić liczbe określającą kształt: 4
Proszę wprowadzić liczbe punktów oscylujących przy kształcie: 300
Proszę podaj x: 1
Proszę podaj y: 1
Proszę podaj x: 2
Proszę podaj y: 1
Proszę podaj x: 2
Proszę podaj y: 2
Proszę podaj x: 1
Proszę podaj y: 2
     1
```

   b. Wykres