Київський національний університет імені Тараса Шевченка
Факультет комп’ютерних наук та кібернетики

Лабораторна робота №2
з дисципліни
“Технології чисельного моделювання”
на тему:

“ОБТІКАННЯ НЕСТАЦІОНАРНОЮ ТЕЧІЄЮ ФІКСОВАНОЇ ПЕРЕШКОДИ”

Виконала
студентка групи ПМ-1
Чернорай Владислава Олегівна

2024

ЗМІСТ

УМОВА ЛАБОРАТОРНОЇ РОБОТИ

Завдання

1. Змоделювати течію навколо перешкоди у вигляді контору W в системі координат;

2. Змоделювати нестаціонарний режим течії з періодичним вихороутворенням у вигляді “доріжки Кармана”.

ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

Програмна реалізація була виконана з використанням Python у середовищі Google Colab.

Посилання на код: Lab_2.ipynb

У роботі було використано такі основні бібліотеки:

- ***NumPy*** (для виконання чисельних операцій і роботи з масивами)
- ***Matplotlib*** (для візуалізації результатів, включно з векторними і скалярними полями)
- ***Math*** (для базових математичних операцій, таких як обчислення косинуса та синуса)
- ***Pandas*** (для обробки та аналізу табличних даних)

Програмний код складається наступних класів, що розподіляють логіку обчислень та візуалізації:

1. ***Клас Obstacle***

   Визначає характеристики перешкоди.

   Основні методи:

   1.1. Метод ***contour_length()***: обчислює загальну довжину контуру перешкоди.

   1.2. Метод ***calculate_discretization_points():*** обчислює рівномірно розподілені точки дискретизації вздовж контуру.

   1.3. Метод ***calculate_collocation_points():*** розраховує точки колокації.

   1.4. Метод ***calculate_collocation_normals():*** обчислює нормальні вектори для кожного сегмента контуру.

   1.5. Метод ***get_vertex_indices()***: визначає індекси точок дискретизації, які відповідають вершинам контуру.

   1.6. Метод ***plot_shape()***: візуалізує контур перешкоди.

2. ***Клас UnsteadyFlowModel***

Моделює несталий потік навколо перешкоди.

2.1. Метод ***_compute_regularization():*** обчислює регуляризаційну константу.

2.2. Метод ***_initialize_circulation():*** ініціалізує циркуляцію.

2.3. Метод ***_update_circulation()***: оновлює значення циркуляції.

2.4. Метод***_compute_velocity()***: обчислює швидкість потоку в певній точці.

2.5. Метод ***_compute_distance():*** обчислює відстань між точками.

2.6. Метод ***_update_free_contour()***: оновлює положення контуру.

2.7. Метод ***_compute_time_step_interval()***: обчислює інтервал часу для ітерацій.

2.8. Метод ***compute_total_velocity()***: розраховує загальну швидкість потоку.

2.9. Метод ***_advance_one_step()***: виконує один крок моделювання.

2.10. Метод ***advance_time()***: виконує моделювання потоку на задану кількість кроків.

3. ***Клас FlowModeling***

Реалізує моделювання нестаціонарного потоку і візуалізацію доріжки вихорів Кармана.

3.1. Метод ***create_unsteady_flow_model():*** створює модель нестаціонарного потоку.

3.2. Метод ***is_clockwise():*** визначає напрямок руху вихорів.

3.3. Метод ***plot_vector_field()***: створює векторне поле для візуалізації швидкостей потоку.

3.4. Метод ***plot_karman_vortex_street()***: моделює та відображає доріжку Кармана разом із векторним полем.

Ініціалізація параметрів та розрахунок характеристик потоку

Для дослідження обтікання перешкоди у вигляді літери ***W*** нестаціонарною течією було обрано п'ять контрольних точок, що формують контур перешкоди.

1. ***Визначення точок перешкоди***: Контур перешкоди представлено п’ятьма точками, які задають геометрію літери ***W*** у декартовій системі координат.

```
obstaclePoints = tuple(map(lambda x: np.array(x),
 ((-0.8, 0), (-0.35, -0.4), (0, -0.1), (0.35, -0.4), (0.8, 0))))
```

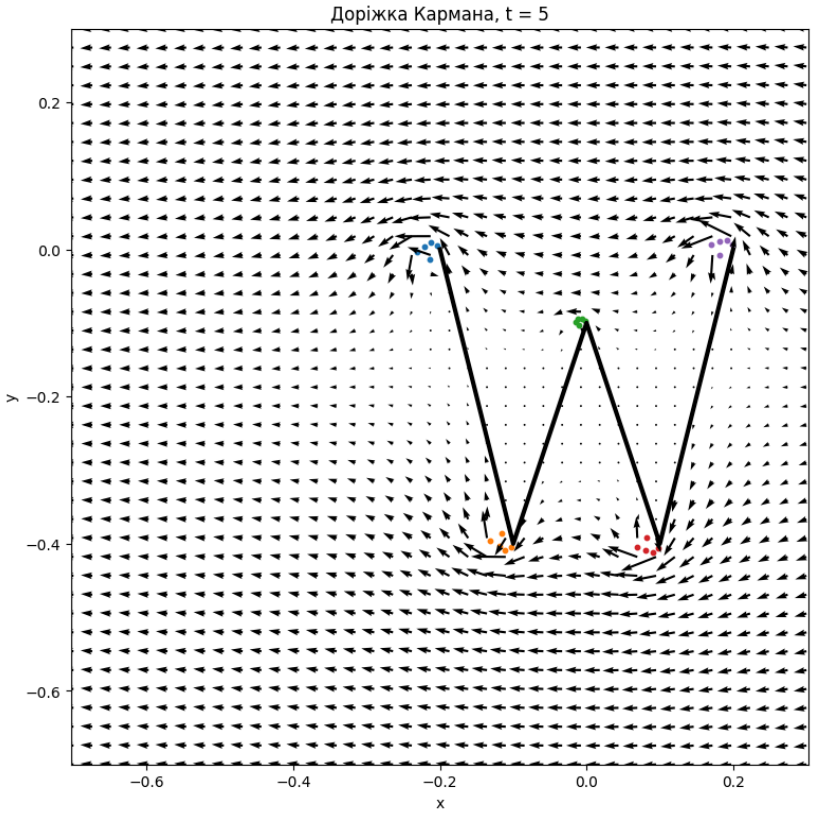
2. ***Задання кількості точок дискретизації****:* Для моделювання потоку навколо перешкоди використовувалися різні варіанти кількості точок дискретизації.
3. ***Кут потоку****:* Напрямок потоку вибрано паралельно до горизонтальної осі.

```
flow_angle = math.pi
```

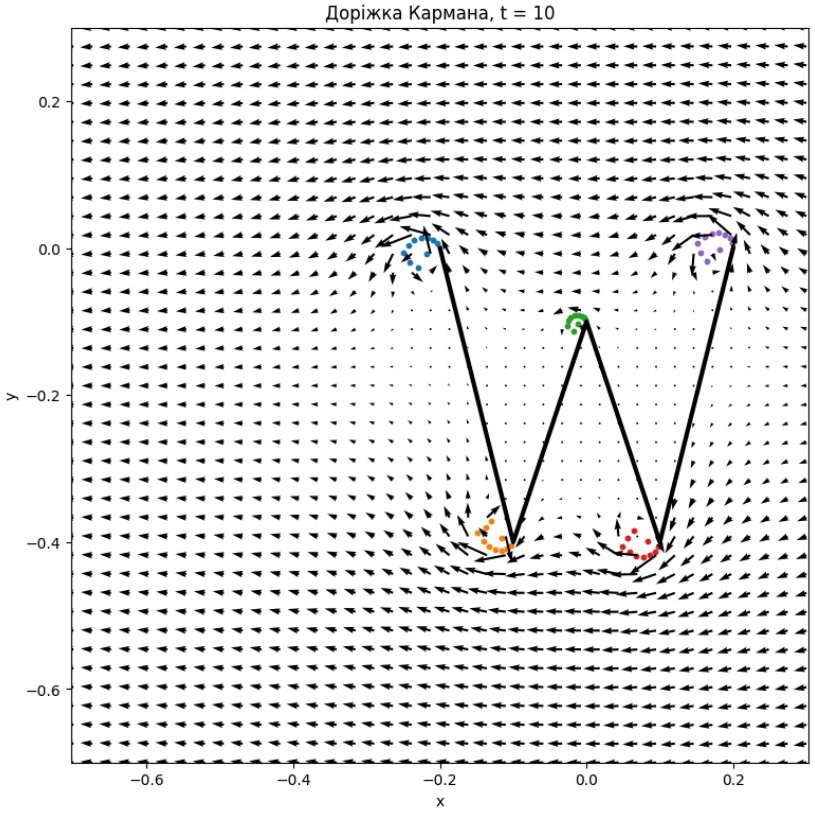
4. ***Задання часових кроків:*** Часові кроки було задано як масив `time_steps = [5, 10, 20, 30, 40, 50, 100]`, що дозволяє аналізувати динаміку потоку та вихороутворення на різних етапах розвитку доріжки Кармана.

РЕЗУЛЬТАТИ

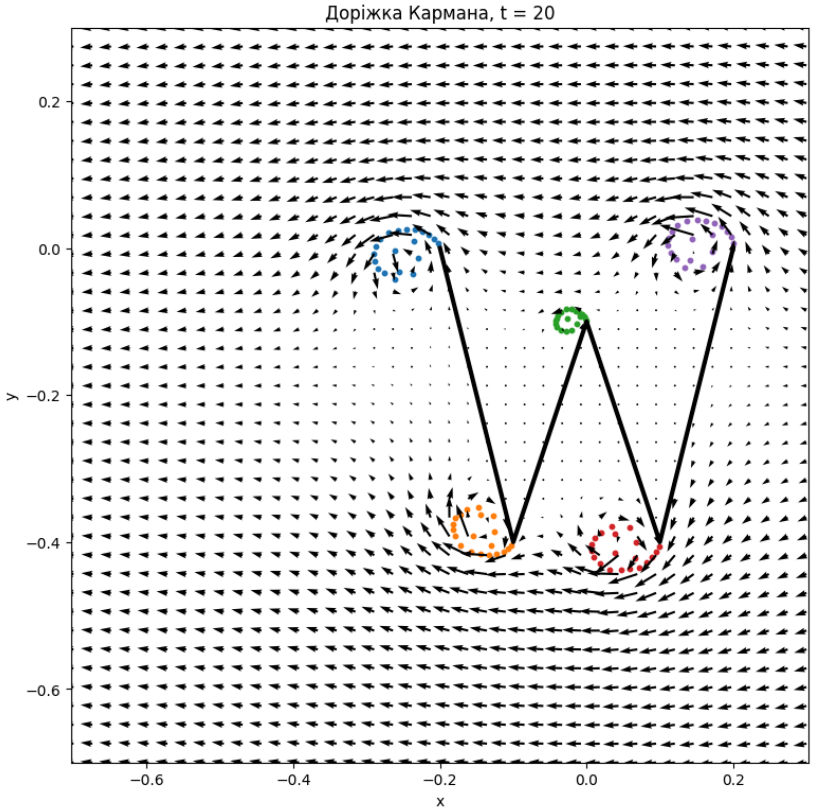
1. *t=5 n=100*



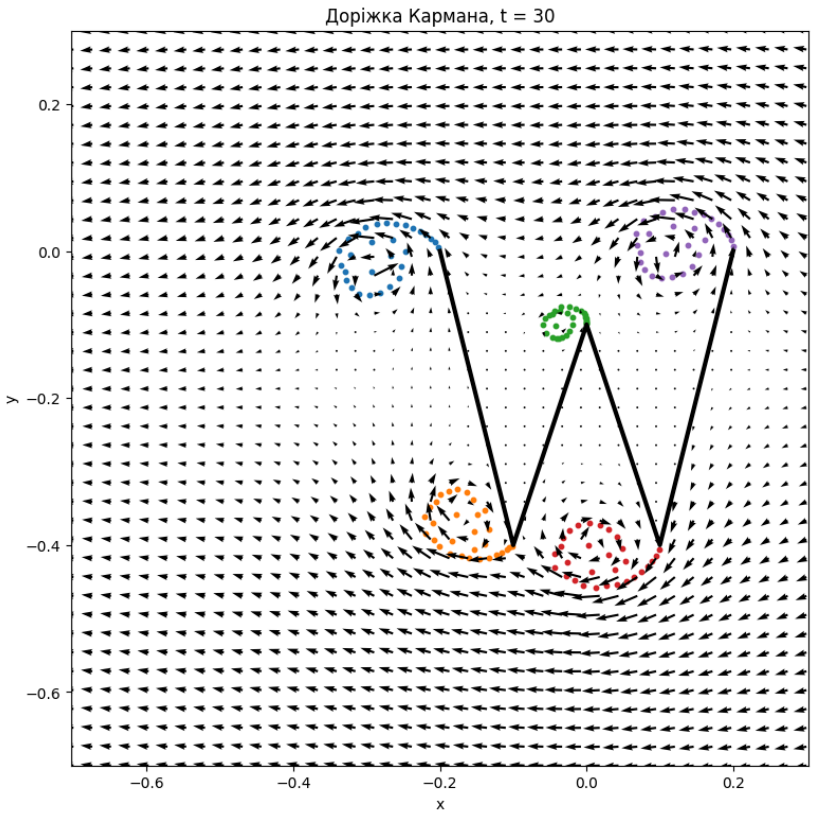
2. *t=10 n=100*



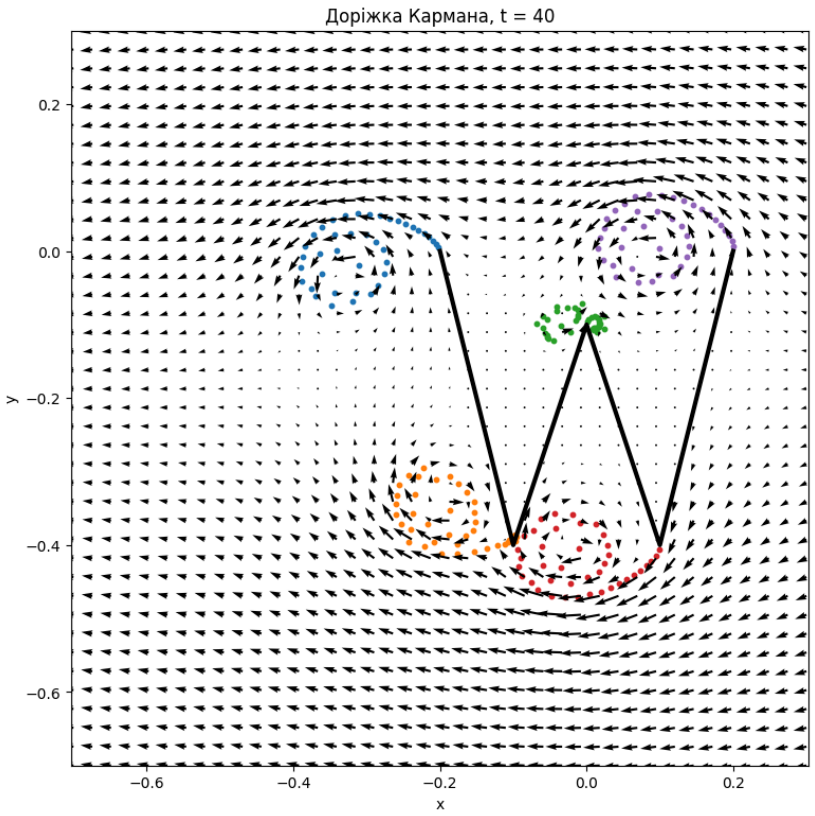
3. *t=20 n=100*



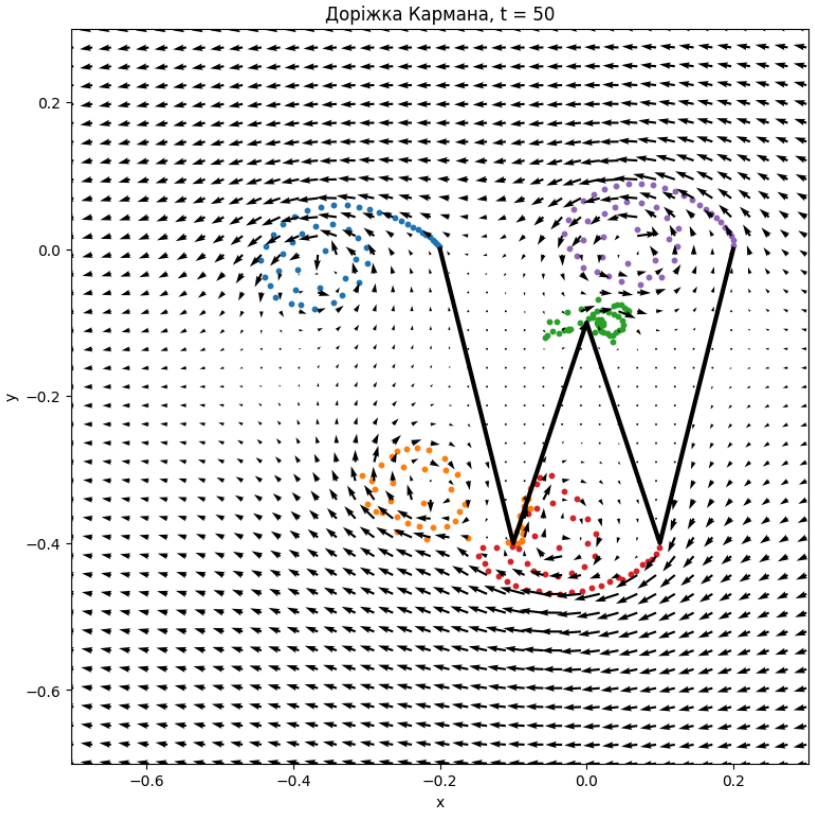
4. *t=30 n=100*



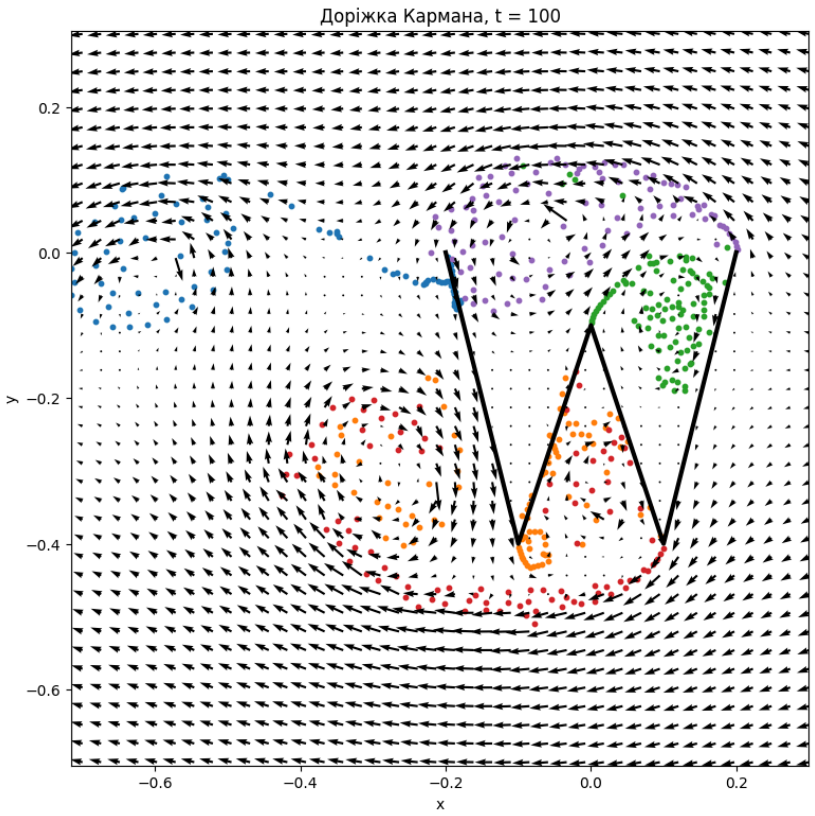
5. *t=40 n=100*



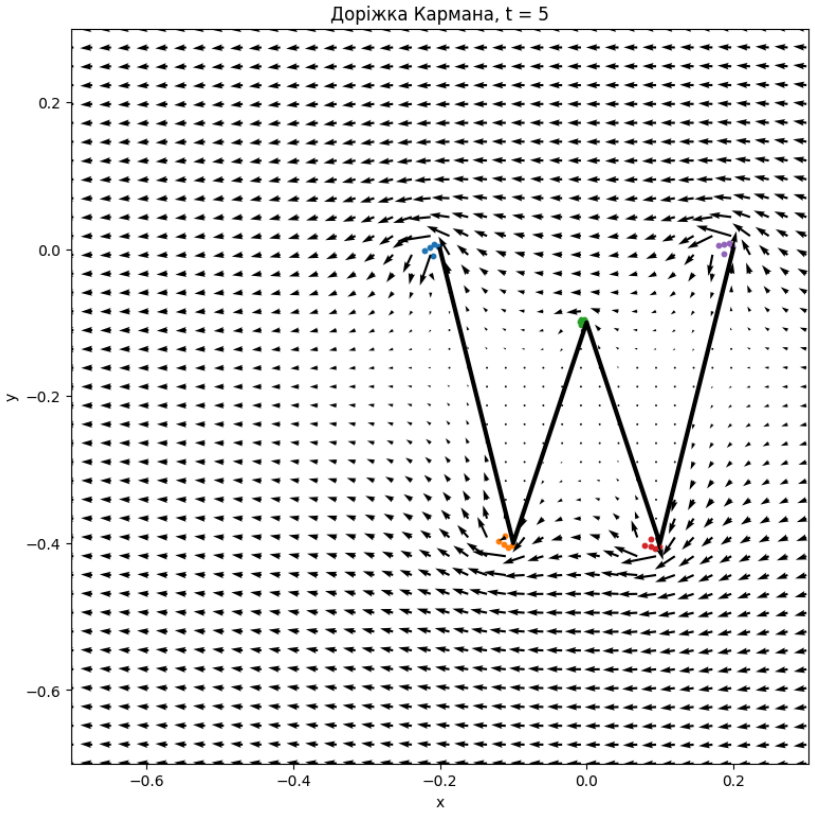
6. *t=50 n=100*



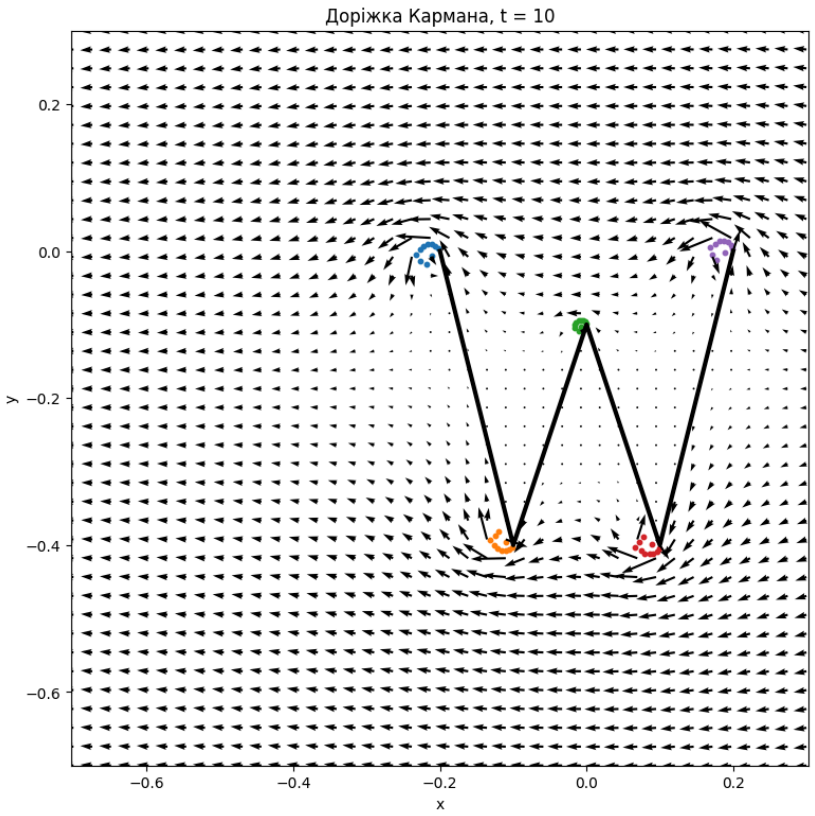
7. *t=100 n=100*



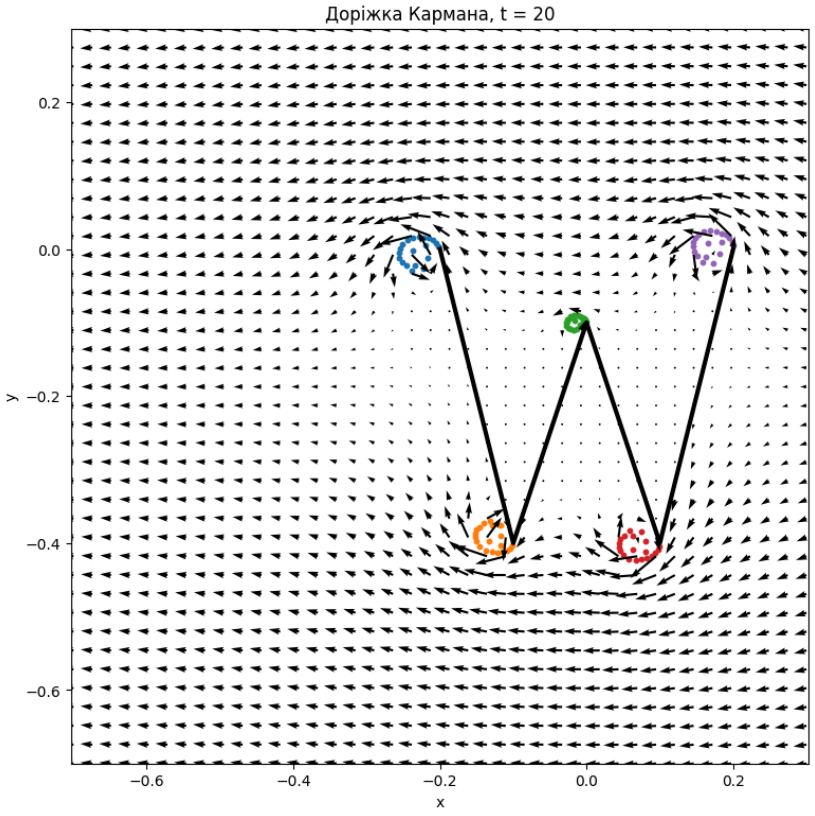
8. t=5 n=150



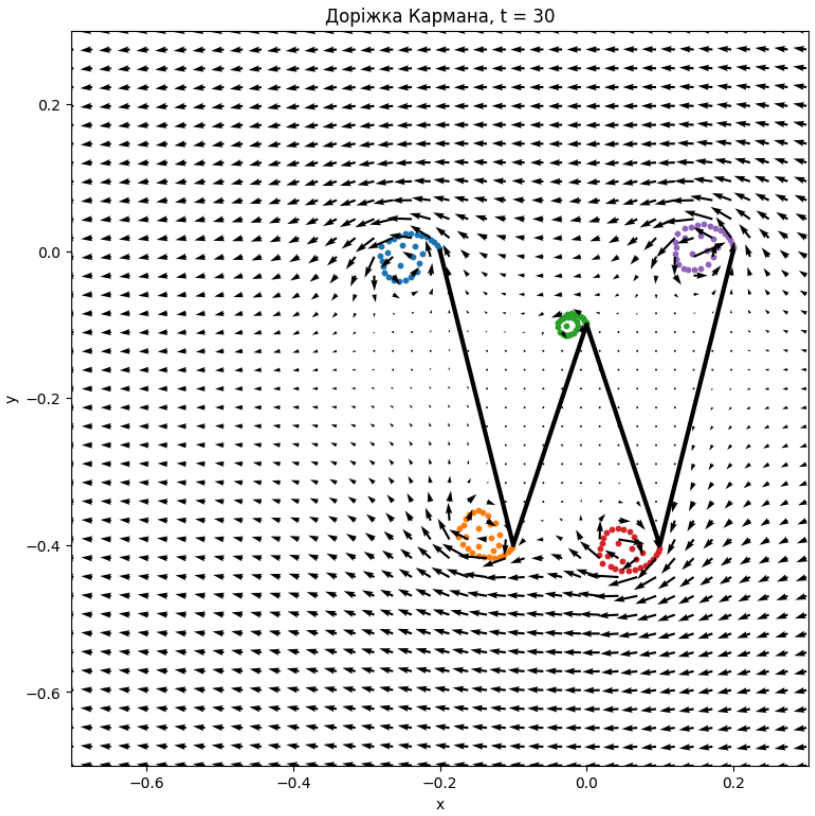
9. t=10 n=150



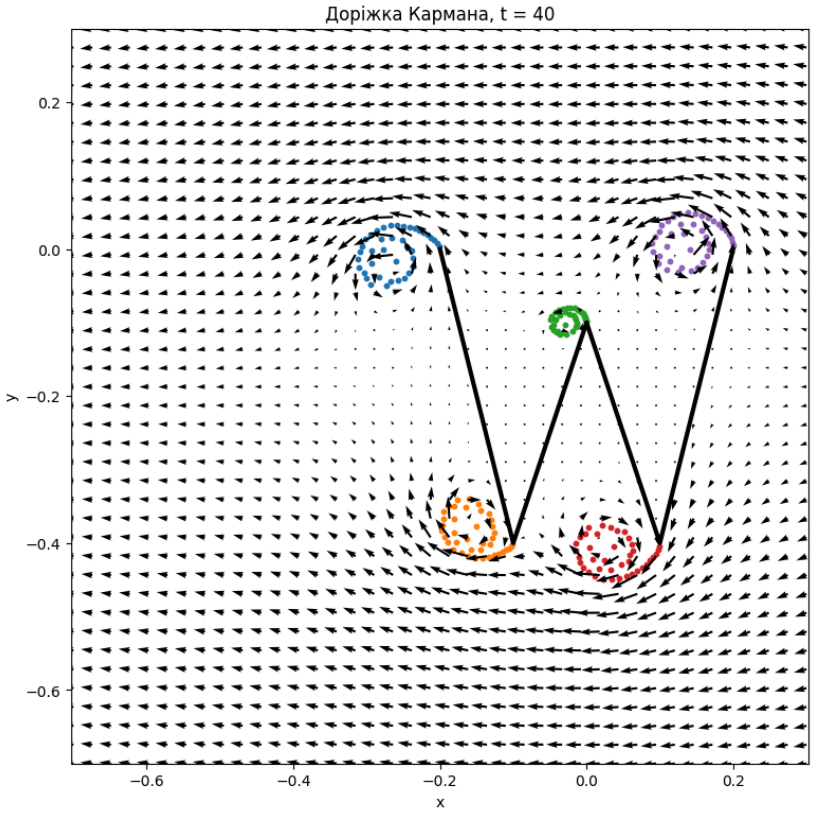
10.t=20 n=150



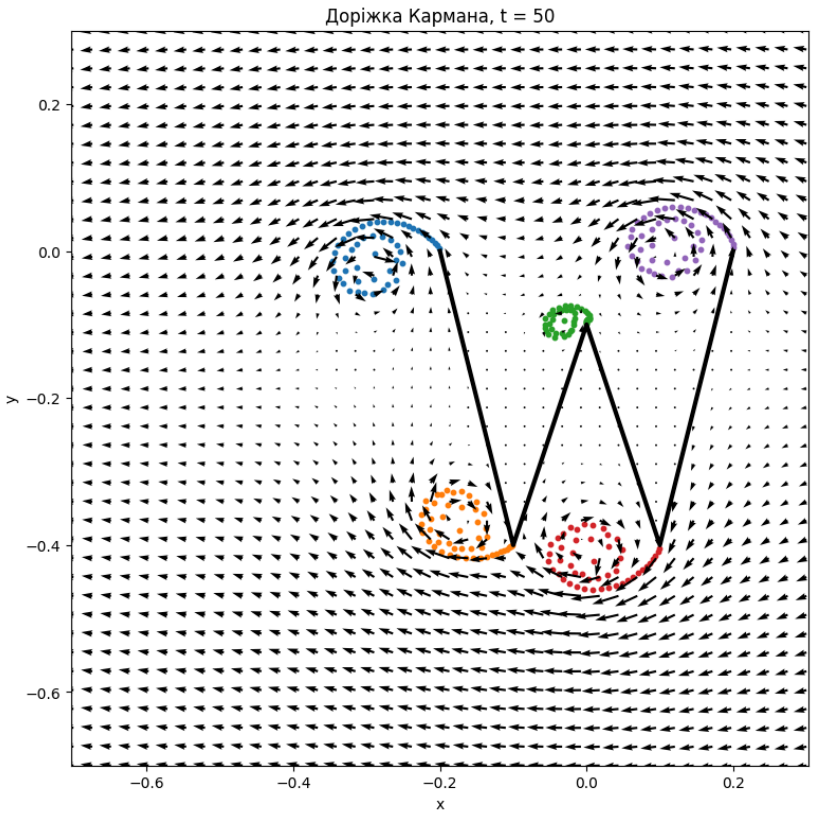
11. t=30 n=150



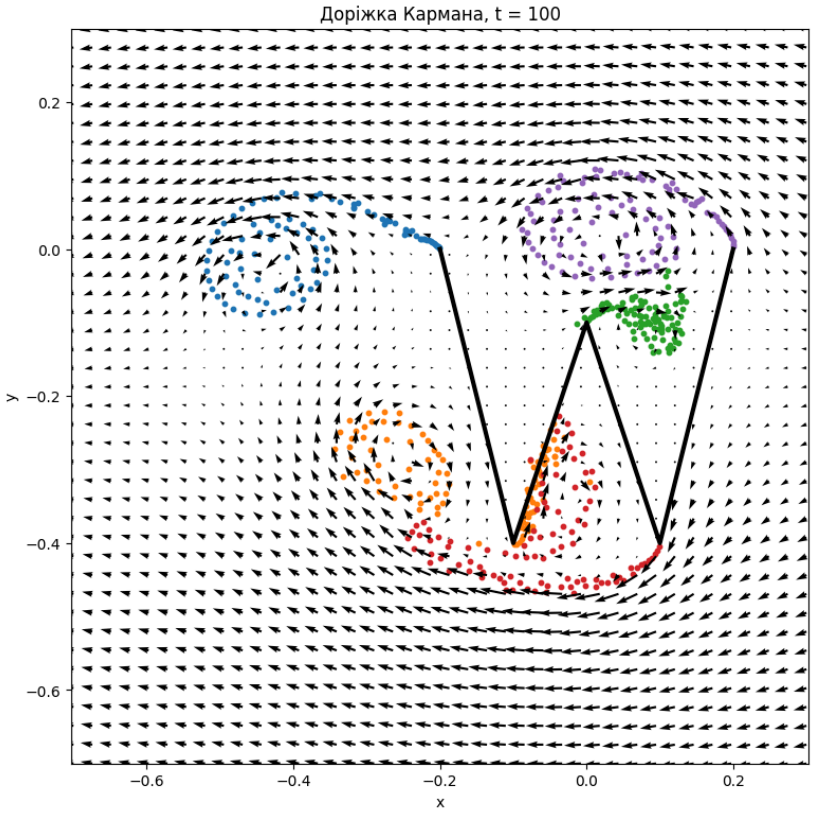
12. t=40 n=150



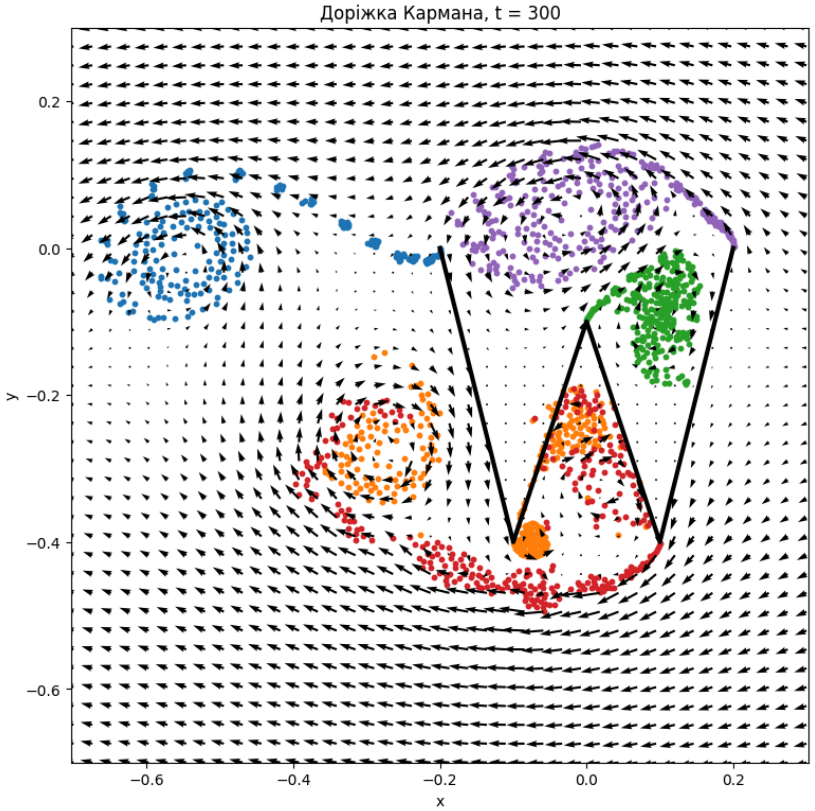
13. t=50 n=150



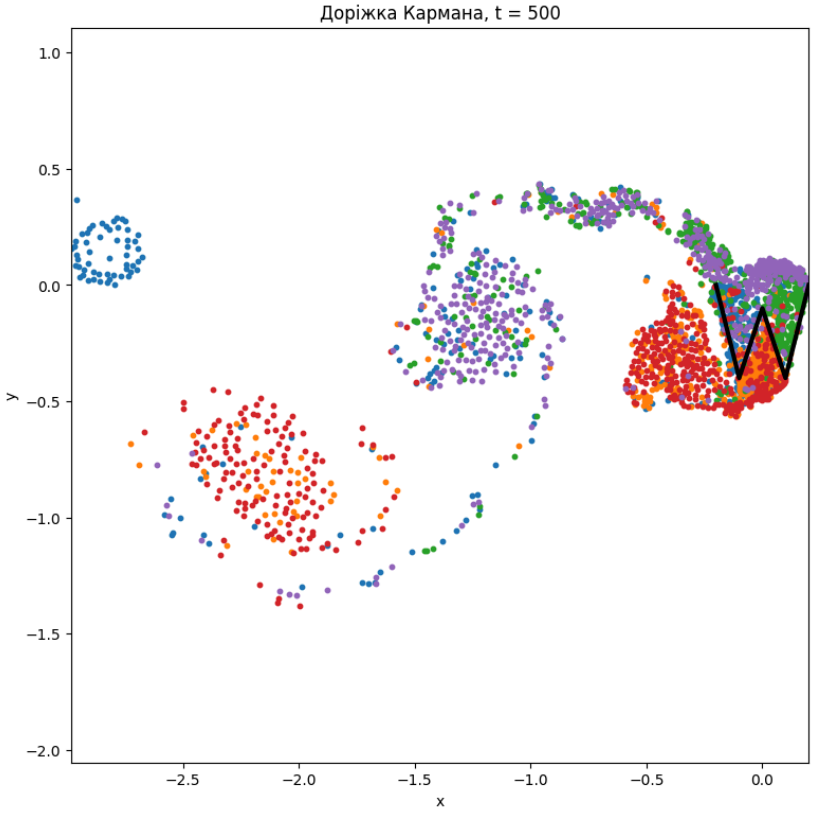
14. t=100 n=150



15. *t=300 n=300*


Доріжка Кармана, t = 300
0.2
0.0
−0.2
−0.4
−0.6
y
−0.6
−0.4
−0.2
0.0
0.2
x

16. t=500 n=100


Доріжка Кармана, t = 500
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
y
−2.5
−2.0
−1.5
−1.0
−0.5
0.0
x

ВИСНОВОК

У ході лабораторної роботи було змодельовано нестаціонарний процес обтікання перешкоди у вигляді літери ***W***. Використовуючи чисельні методи та Python, побудовано модель вихороутворення у вигляді доріжки Кармана. Моделювання продемонструвало вплив кількості точок дискретизації та часового кроку на точність результатів.