

## Protokół z ćwiczenia: Symulacja automatów komórkowych (LAB4)

### Wykonawcy:

- Barbara Pielecha
- Maria Piasecka

## 2. Opis implementacji

### 2.1 Siatka i reguły

- **Rozmiar siatki:** 50×50 komórek (GRID\_SIZE = 50)
- **Stany komórek:** 1 = żywa, 0 = martwa
- **Sąsiedztwo:** Moore’a (8 sąsiadów)
- **Warunki brzegowe:** okresowe (zawijanie siatki – torus)
- **Reguły przejścia (Conway’s Game of Life):**
  - żywa komórka z <2 lub >3 sąsiadami umiera
  - martwa komórka dokładnie z 3 sąsiadami ożywa
  - w pozostałych przypadkach stan nie ulega zmianie

### 2.2 Struktura kodu

- **Biblioteki:**
  - NumPy (reprezentacja siatki jako listy list lub macierzy)
  - Matplotlib (pyplot, animation.FuncAnimation)
- **Moduły i funkcje:**
  1. create\_grid(rows, cols) – inicjalizacja siatki zerami
  2. neighbour\_count(grid, x, y) – licznik żywych sąsiadów z zawijaniem
  3. transition\_rules(grid) – generowanie kolejnej epoki na podstawie reguł
  4. insert\_pattern(grid, pattern, x, y) – nakładanie wzorców (block, beehive, boat, blinker, toad)
  5. glider\_gun(grid, x, y) – inicjalizacja działa Gosper Glider Gun
  6. matuzalek(grid, x, y) – inicjalizacja wzorca Matuzalekh
  7. animate(grid, steps, interval) – animacja kolejnych epok, wyświetlanie liczby żywych komórek, zapis klatek do GIF/JPEG

8. main() – interfejs tekstowy wyboru trybu symulacji

### 3. Wzorce początkowe

Typ	Nazwa patternu	Pozycja (x,y)	Liczba epok	Efekt	Końcowa liczba żywych komórek
Still life	block	(5, 5)	5	niezmienność wzorca	15
	beehive	(20, 20)	5	stabilna struktura	—
	boat	(35, 35)	5	stabilna struktura	—
Oscylatory	blinker	(5, 5)	10	okres cyklu 2	9
	toad	(20, 20)	10	okres cyklu 2	—
Działo	glider_gun	(10, 10)	100	generuje glidery co kilkanaście epok	46
Matuzalekh	matuzalek	(10, 10)	250	złożona, długo ewoluująca struktura	60

### 4. Ilustracje wyników

Poniżej podsumowanie animacji zapisanych jako pliki GIF

1. **Still life** (po 5 epokach)
2. **Oscylatory** (po 10 epokach)
3. **Gosper Glider Gun** (po 100 epokach)
4. **Matuzalekh** (po 250 epokach)

## 5. Wnioski końcowe

### 1. Zgodność z teorią:

- Otrzymane wzorce odpowiadają przewidywaniom dla still lives i oscylatorów.
- Glider Gun poprawnie generuje glidery, a Matuzalekh tworzy złożone, powtarzalne fragmenty.

### 2. Wydajność obliczeniowa:

- Złożoność pojedynczej iteracji to  $O(N^2)$  gdzie  $N=50$  (2500 komórek).
- Animacja 100 epok trwa  $\sim 1,5$  s na przeciętnym laptopie.

### 3. Możliwości rozwoju:

- Zamiana list na macierze NumPy i wektorowe operacje (przyspieszenie).
- Użycie zapisu JIT (Numba) lub paralelizacja.