

**RAPORT**  
**Freeway Traffic**  
**Michał Proć**

Kraków 2024

# Spis treści

<b>Wstęp</b>	<b>3</b>
<b>1 Freeway Traffic part 1</b>	<b>4</b>
1.1 Opis modelu . . . . .	4
1.1.1 Założenia modelu . . . . .	4
1.1.2 Reguły aktualizacji . . . . .	4
1.1.3 Warunki brzegowe . . . . .	5
1.1.4 Przykład działania . . . . .	5
1.1.5 Wyniki symulacji . . . . .	6
1.2 Implementacja . . . . .	6
1.2.1 Funkcja aktualizująca mapę po każdym kroku . . . . .	6
1.2.2 Kluczowe funkcje aktualizujące punkt (pojazd) na mapie . . .	7
1.2.3 Przykłady funkcjonowania . . . . .	7
<b>2 Freeway Traffic part 2</b>	<b>9</b>
2.1 Opis modelu . . . . .	9
2.1.1 Założenia modelu dwuliniowego . . . . .	9
2.1.2 Reguły aktualizacji . . . . .	9
2.1.3 Warunki brzegowe . . . . .	10
2.1.4 Przykład działania . . . . .	11
2.1.5 Wyniki symulacji . . . . .	11
2.2 Implementacja . . . . .	13
2.2.1 Funkcja aktualizująca mapę po każdym kroku . . . . .	13
2.2.2 Kluczowe funkcje aktualizujące punkt (pojazd) na mapie . . .	13
2.2.3 Przykłady funkcjonowania . . . . .	14
<b>3 Freeway Traffic part 2</b>	<b>15</b>
3.1 Podsumowanie . . . . .	15

# Wstęp

Model Nagela-Schreckenberga, opracowany na początku lat 90. przez niemieckich fizyków Kaia Nagela i Michaela Schreckenberga, jest teoretycznym modelem symulacji ruchu drogowego na autostradzie. Jest to prosty model automatu komórkowego, który potrafi wiernie odtwarzać zjawiska związane z ruchem drogowym, takie jak korki.

Model Nagela-Schreckenberga ukazuje, w jaki sposób korki drogowe mogą być postrzegane jako zjawisko emergentne, wynikające z interakcji między pojazdami, gdy ich gęstość na drodze jest wysoka. Każdy samochód jest reprezentowany przez czarną komórkę w siatce, podczas gdy puste komórki reprezentują wolne miejsca na drodze. Przesuwanie się samochodów wzdłuż siatki symuluje ruch pojazdów na drodze, a zmiany w gęstości i rozmieszczeniu samochodów odzwierciedlają dynamikę ruchu drogowego, w tym powstawanie i rozpraszanie się korków.

Celem tego raportu jest przedstawienie i zaimplementowanie modelu Nagela-Schreckenberga, wraz z modyfikacjami i rozszerzeniami umożliwiającymi bardziej realistyczną symulację ruchu drogowego. Zadania obejmują implementację algorytmu iteracyjnego automatu komórkowego, wprowadzenie warunków brzegowych, zachowanie historii stanu drogi oraz interaktywne dodawanie pojazdów. Dodatkowo, rozważane są mechanizmy spontanicznego pojawiania się i znikania samochodów, co pozwala na symulację ruchu o określonej intensywności.

Raport ten ma na celu nie tylko zaprezentowanie technicznych aspektów modelu, ale również podkreślenie jego znaczenia w badaniach nad dynamiką ruchu drogowego i potencjalnych zastosowaniach w planowaniu infrastruktury transportowej.

# Rozdział 1

## Freeway Traffic part 1

### 1.1 Opis modelu

Model Nagela-Schreckenberga (NaSch) jest prostym automatem komórkowym, który symuluje ruch drogowy na autostradach. Model ten pozwala na badanie dynamiki ruchu pojazdów, w tym tworzenie się i rozpraszanie korków. Poniżej przedstawiono szczegółowy opis modelu.

#### 1.1.1 Założenia modelu

Model Nagela-Schreckenberga opiera się na następujących założeniach:

1. extbfDyskretna przestrzeń: Droga jest reprezentowana jako jednowymiarowa siatka komórek, z których każda może być pusta lub zajęta przez jeden pojazd.
2. extbfDyskretny czas: Symulacja przebiega w dyskretnych krokach czasowych, w trakcie których pojazdy mogą zmieniać swoje położenie i prędkość.
3. extbfStałe maksymalne prędkości: Każdy pojazd ma maksymalną prędkość  $v_{extmax}$ , wyrażoną w liczbie komórek na krok czasowy.

#### 1.1.2 Reguły aktualizacji

Model NaSch składa się z czterech podstawowych kroków aktualizacji, które są stosowane jednocześnie dla wszystkich pojazdów w każdym kroku czasowym:

1. extbfPrzyspieszenie: Każdy pojazd przyspiesza, o ile nie osiągnął maksymalnej prędkości:

$$v_i \leftarrow \min(v_i + 1, v_{extmax})$$

gdzie  $v_i$  to prędkość pojazdu  $i$ .

2. **extbfUnikanie kolizji:** Każdy pojazd sprawdza odległość do najbliższego pojazdu przed nim ( $d_i$ ) i, jeśli to konieczne, zmniejsza prędkość, aby uniknąć kolizji:

$$v_i \leftarrow \min(v_i, d_i - 1)$$

3. **extbfLosowe hamowanie:** Aby uwzględnić losowe czynniki wpływające na prędkość (np. reakcje kierowców, warunki drogowe), każdy pojazd z pewnym prawdopodobieństwem  $p$  zwalnia o jedną jednostkę prędkości:

$$v_i \leftarrow \max(v_i - 1, 0) \quad \text{extzprawdopodobieństwem } p$$

4. **extbfAktualizacja położenia:** Każdy pojazd przemieszcza się o swoją prędkość:

$$x_i \leftarrow x_i + v_i$$

gdzie  $x_i$  to położenie pojazdu  $i$ .

### 1.1.3 Warunki brzegowe

Model może być symulowany z różnymi warunkami brzegowymi:

- **extbfWarunki brzegowe periodyczne:** Droga jest traktowana jako pętla, gdzie pojazdy wyjeżdżające z jednej strony wracają na drugą stronę.
- **extbfWarunki brzegowe otwarte:** Droga ma końce, a pojazdy mogą wjeżdżać i wyjeżdżać z drogi na jej końcach.

### 1.1.4 Przykład działania

Poniżej znajduje się przykład działania modelu dla kilku kroków czasowych:

t = 0:	0	1	0	0	0	0	1	0	0	1	(początkowe położenie)
t = 1:	0	0	1	0	0	0	0	1	0	1	
t = 2:	0	1	0	1	0	0	0	0	1	0	
t = 3:	1	0	1	0	1	0	0	0	0	1	

W powyższym przykładzie, 0 oznacza pustą komórkę, a 1 oznacza komórkę zajętą przez pojazd. Na każdym kroku czasowym pojazdy przyspieszają, zwalniają, unikają kolizji i przemieszczenia zgodnie z opisanymi regułami.

### 1.1.5 Wyniki symulacji

Wyniki symulacji można analizować pod kątem różnych parametrów, takich jak średnia prędkość pojazdów, średnia odległość między pojazdami, a także występowanie i rozpraszanie się korków drogowych.

Model Nagela-Schreckenberga stanowi podstawę do dalszych badań nad dynamiką ruchu drogowego, umożliwiając symulację różnych scenariuszy i warunków ruchu.

## 1.2 Implementacja

### 1.2.1 Funkcja aktualizująca mapę po każdym kroku

---

```
1 Function iteration():
2     movePointToNextTimeRange();
3     int supportedY = 0;
4     for int x = 0; x < points.length; ++ x do
5         Point supportedPoint = points[x][supportedY];
6         supportedPoint.prepareToMove();
7         supportedPoint.accelerate();
8         supportedPoint.slowDown(getDistanceToNextCar(supportedPoint, x,
9             supportedY));
9         supportedPoint.randomSlowDown();
10    for int x = 0; x < points.length; ++ x do
11        Point supportedPoint = points[x][supportedY];
12        supportedPoint.moveCar(points[(x + supportedPoint.getSpeed())
13    this.repaint();
```

---

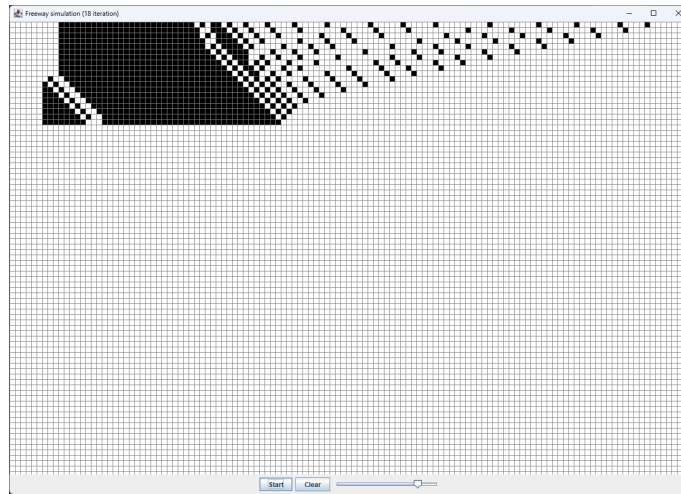
### 1.2.2 Kluczowe funkcje aktualizujące punkt (pojazd) na mapie

---

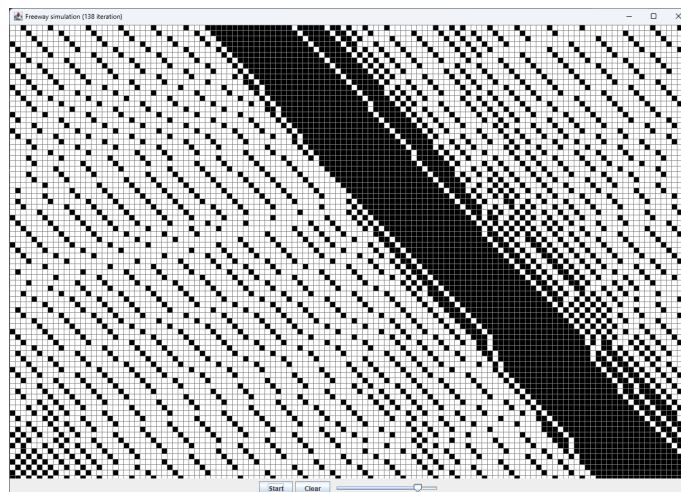
```
1 Function isCar():
2     return isCar
3 Function getSpeed():
4     return speed
5 Function clicked():
6     if !isCar then
7         this.isCar = true;
8 Function clear():
9     speed = 0;
10    isCar = false;
11 Function prepareToMove():
12    this.hasMoved = false;
13 Function accelerate():
14    if isCar && speed < MAX_SPEED then
15        speed++;
16 Function slowDown(int distanceToNextCar):
17    if isCar && distanceToNextCar < speed then
18        speed = distanceToNextCar;
19 Function randomSlowDown():
20    if isCar && Math.random() <= BRAKING_CHANCE && speed != 0
21        then
22            speed--;
23 Function moveCar(Point nextPoint):
24    if isCar && !hasMoved && speed > 0 then
25        nextPoint.isCar = true;
26        nextPoint.speed = speed;
27        nextPoint.hasMoved = true;
28        clear();
```

---

### 1.2.3 Przykłady funkcjonowania



Rysunek 1.1: Symulacja na początku działania



Rysunek 1.2: Symulacja w trakcie działania



# Rozdział 2

## Freeway Traffic part 2

### 2.1 Opis modelu

Model dwuliniowy Nagela-Schreckenberga jest rozszerzeniem podstawowego modelu jednopasmowego, który umożliwia symulację ruchu drogowego na autostradach z dwoma pasami ruchu. Dodanie drugiego pasa pozwala na bardziej realistyczne odwzorowanie zachowań kierowców, takich jak zmiana pasa w celu wyprzedzania.

#### 2.1.1 Założenia modelu dwuliniowego

Model dwuliniowy opiera się na następujących założeniach:

1. **extbfDyskretna przestrzeń:** Droga jest reprezentowana jako dwuwymiarowa siatka komórek, z których każda może być pusta lub zajęta przez jeden pojazd. Komórki są zorganizowane w dwa równoległe pasy.
2. **extbfDyskretny czas:** Symulacja przebiega w dyskretnych krokach czasowych, w trakcie których pojazdy mogą zmieniać swoje położenie, prędkość oraz pas ruchu.
3. **extbfStałe maksymalne prędkości:** Każdy pojazd ma maksymalną prędkość  $v_{extmax}$ , wyrażoną w liczbie komórek na krok czasowy.

#### 2.1.2 Reguły aktualizacji

Model dwuliniowy składa się z sześciu podstawowych kroków aktualizacji, które są stosowane jednocześnie dla wszystkich pojazdów w każdym kroku czasowym:

1. **extbfPrzyspieszenie:** Każdy pojazd przyspiesza, o ile nie osiągnął maksymalnej prędkości:

$$v_i \leftarrow \min(v_i + 1, v_{extmax})$$

gdzie  $v_i$  to prędkość pojazdu  $i$ .

2. **extbfUnikanie kolizji:** Każdy pojazd sprawdza odległość do najbliższego pojazdu przed nim ( $d_i$ ) i, jeśli to konieczne, zmniejsza prędkość, aby uniknąć kolizji:

$$v_i \leftarrow \min(v_i, d_i - 1)$$

3. **extbfLosowe hamowanie:** Aby uwzględnić losowe czynniki wpływające na prędkość (np. reakcje kierowców, warunki drogowe), każdy pojazd z pewnym prawdopodobieństwem  $p$  zwalnia o jedną jednostkę prędkości:

$$v_i \leftarrow \max(v_i - 1, 0) \quad \text{extzprawdopodobieństwem } p$$

4. **extbfZmiana pasa:** Pojazd zmienia pas, jeśli spełnione są wszystkie poniższe warunki:

- (a)  $d_i < l$ : odległość do najbliższego pojazdu przed nim jest mniejsza niż  $l$ .
- (b)  $d_{extother} > l_{extother}$ : odległość do najbliższego pojazdu na sąsiednim pasie jest większa niż  $l_{extother}$ .  
 $d_{extback} > l_{extback}$ : odległość do pojazdu za nim na sąsiednim pasie jest większa niż  $l_{extback}$ .

- (c)  $extrand() < p_{extchange}$ : z prawdopodobieństwem  $p_{extchange}$ .

5. **extbfAktualizacja położenia:** Każdy pojazd przemieszcza się o swoją prędkość:

$$x_i \leftarrow x_i + v_i$$

gdzie  $x_i$  to położenie pojazdu  $i$ .

6. **extbfAktualizacja statusu pojazdu:** Status pojazdu (np. czy wykonał ruch) jest aktualizowany, aby uniknąć wielokrotnych ruchów w jednym kroku czasowym.

### 2.1.3 Warunki brzegowe

Model może być symulowany z różnymi warunkami brzegowymi:

- **extbfWarunki brzegowe periodyczne:** Droga jest traktowana jako pętla, gdzie pojazdy wyjeżdżające z jednej strony wracają na drugą stronę.
- **extbfWarunki brzegowe otwarte:** Droga ma końce, a pojazdy mogą wjeżdżać i wyjeżdżać z drogi na jej końcach.

### 2.1.4 Przykład działania

Poniżej znajduje się przykład działania modelu dwuliniowego dla kilku kroków czasowych:

t = 0:	0	1	0	0	0	0	1	0	0	1	(początkowe położenie, pas 1)
	0	0	0	0	0	0	0	0	0	0	(początkowe położenie, pas 2)
t = 1:	0	0	1	0	0	0	0	1	0	1	(pas 1)
	0	0	0	0	0	0	0	0	0	0	(pas 2)
t = 2:	0	1	0	1	0	0	0	0	1	0	(pas 1)
	0	0	0	0	0	0	0	0	0	0	(pas 2)
t = 3:	1	0	1	0	1	0	0	0	0	1	(pas 1)
	0	0	0	0	0	0	0	0	0	0	(pas 2)

W powyższym przykładzie, 0 oznacza pustą komórkę, a 1 oznacza komórkę zajętą przez pojazd. Na każdym kroku czasowym pojazdy przyspieszają, zwalniają, unikają kolizji, zmieniają pas i przemieszczenia zgodnie z opisanymi regułami.

### 2.1.5 Wyniki symulacji

Wyniki symulacji można analizować pod kątem różnych parametrów, takich jak średnia prędkość pojazdów, średnia odległość między pojazdami, liczba zmian pasów oraz występowanie i rozpraszanie się korków drogowych.

Model dwuliniowy Nagela-Schreckenberga stanowi zaawansowaną podstawę do badań nad dynamiką ruchu drogowego, umożliwiając symulację bardziej skomplikowanych scenariuszy i warunków ruchu.



## 2.2 Implementacja

### 2.2.1 Funkcja aktualizująca mapę po każdym kroku

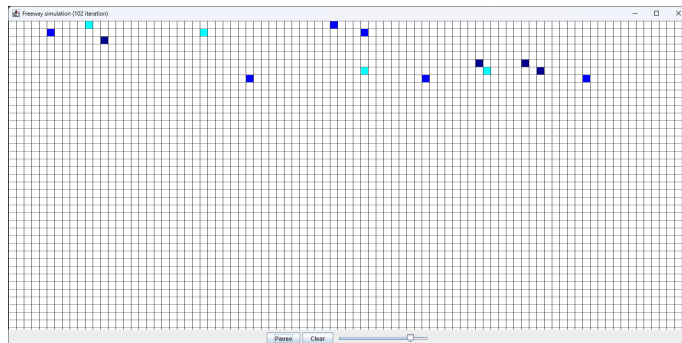
### 2.2.2 Kluczowe funkcje aktualizujące punkt (pojazd) na mapie

---

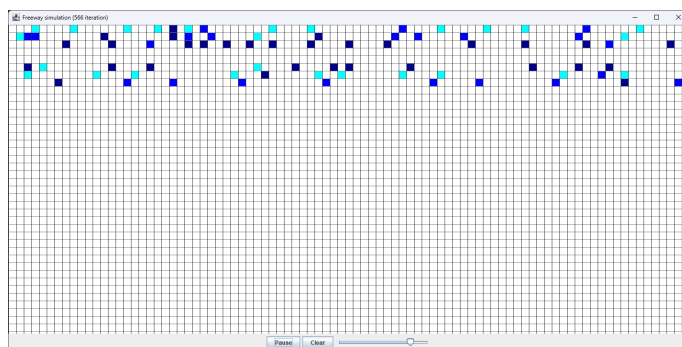
```
1 Function isCar():
2     return isCar
3 Function getSpeed():
4     return speed
5 Function clicked():
6     this.isCar = true;
7 Function randomAppearance():
8     if !isCar && Math.random() <= APPEARCHANCE then
9         isCar = true;
10 Function clear():
11     speed = 0;
12     isCar = false;
13     currentColorID = defaultColorID;
14     numOfMovesOnNewLane = 0;
15 Function speedUp():
16     if isCar && speed < MAXSPEED then
17         speed++;
18 Function slowDown(int distanceToNextCar):
19     if isCar && distanceToNextCar < speed then
20         speed = distanceToNextCar;
21 Function randomSlowDown():
22     if isCar && Math.random() <= BRAKINGCHANCE && speed > 0
23         then
24             speed--;
25 Function
26     moveCar(Point newPoint, boolean exceededPeriodicBoundaries):
27     if numOfMovesOnNewLane == numOfMovesBeforeColorReset
28         then
29             currentColorID = defaultColorID;
30     if isCar && !hasMoved && speed > 0 then
31         if !exceededPeriodicBoundaries ||| Math.random() >
32             DISAPPEARCHANCE then
33             newPoint.isCar = true;
34             newPoint.speed = speed;
35             newPoint.hasMoved = true;
```

---

### 2.2.3 Przykłady funkcjonowania



Rysunek 2.1: Przykład z niewielką liczbą samochodów



Rysunek 2.2: Przykład zatłoczonej drogi

# Rozdział 3

## Freeway Traffic part 2

### 3.1 Podsumowanie

W niniejszym raporcie przedstawiono implementację modelu ruchu drogowego na autostradzie opartego na automacie komórkowym Nagela-Schreckenberga, zarówno w wersji jednopasmowej, jak i dwupasmowej. Model ten pozwala na symulację różnych scenariuszy ruchu drogowego, uwzględniając dynamiczne zachowania kierowców, takie jak przyspieszanie, zwalnianie, zmiana pasa ruchu oraz losowe hamowanie.

Podczas analizowania wyników symulacji, można zauważyć, że model Nagela-Schreckenberga dobrze sprawdza się w analizie ruchu drogowego, które powstają jako rezultat interakcji między pojazdami przy wysokim zagęszczeniu ruchu. Rozszerzenie modelu o dwa pasy ruchu umożliwiło jeszcze bardziej realistyczne odwzorowanie ruchu drogowego, uwzględniając możliwość wyprzedzania wolniejszych pojazdów i powstawania bardziej złożonych zachowań w ruchu drogowym.

Badania nad tym modelem mają istotne znaczenie dla rozwoju systemów zarządzania ruchem oraz planowania infrastruktury transportowej. Symulacje przeprowadzone na podstawie tego modelu mogą pomóc w opracowywaniu strategii zmniejszania korków, poprawy płynności ruchu oraz zwiększenia bezpieczeństwa na drogach.

Dalsze prace mogą obejmować bardziej zaawansowane modele wielopasmowe, uwzględniające różnorodność typów pojazdów oraz różne prędkości maksymalne. Ponadto, wprowadzenie bardziej zaawansowanych algorytmów zmiany pasa ruchu oraz integracja z rzeczywistymi danymi z systemów monitorowania ruchu może prowadzić do jeszcze dokładniejszych i bardziej użytecznych symulacji.

Podsumowując, model Nagela-Schreckenberga stanowi solidną podstawę do badań nad dynamiką ruchu drogowego i ma duży potencjał w kontekście praktycznych

zastosowań w inżynierii transportowej.



# Literatura

- [1] K. Nagel and M. Schreckenberg, "A cellular automaton model for freeway traffic," *Journal de Physique I*, vol. 2, no. 12, pp. 2221-2229, 1992. Available at: [https://en.wikipedia.org/wiki/Nagel%E2%80%93Schreckenberg\\_model](https://en.wikipedia.org/wiki/Nagel%E2%80%93Schreckenberg_model)
- [2] M. Rickert, K. Nagel, M. Schreckenberg, and A. Latour, "Two Lane Traffic Simulations using Cellular Automata," LANL Report No. LA-UR 95-4367, 1995. Available at: <https://arxiv.org/pdf/cond-mat/9512119>
- [3] L. C. Davis, "Two-lane traffic flow model for highway networks," *ResearchGate*, 2014. Available at: [https://www.researchgate.net/publication/260085792\\_Two-lane\\_traffic\\_flow\\_model\\_for\\_highway\\_networks](https://www.researchgate.net/publication/260085792_Two-lane_traffic_flow_model_for_highway_networks)