

**Proyecto Física Computacional: Dinámica molecular en dos dimensiones basado en el paper: “Traffic jams without bottlenecks-experimental evidence for the physical mechanism of the formation of a jam”**

Gabriel Contreras Abellán - C12251\*  
*Escuela de Física, Universidad de Costa Rica*  
(Dated: 4 de diciembre de 2025)

Desde el punto de vista físico, un embotellamiento es un sistema de partículas que interactúan fuera del equilibrio. El efecto colectivo del sistema de múltiples partículas induce la inestabilidad de un estado de flujo libre causado por la intensificación de las fluctuaciones, y la transición a un estado de congestión ocurre de manera espontánea si la densidad promedio de vehículos supera un cierto valor crítico. Por lo tanto, un cuello de botella es solo un desencadenante y no el origen esencial de un atasco. En este artículo, presentamos la primera evidencia experimental de que la aparición de un atasco es un fenómeno colectivo similar a las transiciones de fase 'dinámicas' y a la formación de patrones.

<b>Contents</b>				
		5.3. Procedimiento de Ejecución		4
<b>1. Introducción</b>	2	<b>6. Resultados y Análisis de la Transición de Fase</b>		4
<b>2. Problema</b>	2	6.1. Dinámica de la Congestión		4
<b>3. Objetivos</b>	2	6.2. Análisis Cuantitativo		4
<b>4. Marco teórico</b>	2	6.2.1. Diagrama Espacio-Tiempo		4
4.1. Dinámica Molecular	2	6.2.2. Diagrama de Flujo Fundamental		4
4.2. Implementación Computacional	2			
4.2.1. Lenguaje y herramientas	2	<b>7. Implementación</b>		4
4.3. Movimiento Browniano y Transiciones de Fase Dinámicas	2	<b>8. Conclusiones</b>		6
4.4. Fluctuaciones fuera del equilibrio en sistemas de partículas	3	<b>9. Bibliografía</b>		6
4.5. Formación de patrones en un sistema fuera del equilibrio	3			
<b>5. Metodología de Simulación Computacional</b>	3			
5.1. Modelo de Seguimiento de Vehículos (IDM)	3			
5.1.1. Ecuación Fundamental del IDM	3			
5.1.2. Integración Temporal	3			
5.2. Configuración del Escenario de Simulación	3			

---

\*Electronic address: gabriel.contrerasabellan@ucr.ac.cr

## 1. Introducción

Simulación de la Transición de Fase en el Tráfico Vehicular. Este proyecto de física computacional se dedica a simular y analizar la formación espontánea de atascos de tráfico, buscando demostrar que la congestión es una manifestación de una transición de fase colectiva, más que el simple resultado de un obstáculo físico.

El proyecto se basa en un paper japonés de 2008 publicado por New Journal of Physics nombrado "Traffic jams without bottlenecks-experimental evidence for the physical mechanism of the formation of a jam" [1]. El experimento se desarrolla en un entorno simulado de una carretera circular de 230 metros de circunferencia, con 22 vehículos. En este caso se modelan los vehículos como partículas. La condición inicial es un movimiento homogéneo con velocidad uniforme, ajustando la densidad promedio de vehículos a un valor crítico que configura el inicio de la inestabilidad.

## 2. Problema

¿Cómo y por qué un flujo vehicular inicialmente homogéneo y estable experimenta una "transición de fase" [2] a un estado congestionado, impulsado únicamente por el movimiento colectivo y las interacciones entre vehículos cuando la densidad excede un valor umbral, imitando la formación de una onda de choque (presa fantasma) sin la intervención de una causa externa?

## 3. Objetivos

- Implementar la solución en Python
- Inicializar el sistema a partir de posiciones conocidas.
- Elaborar subrutinas que evalúan si los discos chocan o cambian de velocidad. Calcular las nuevas velocidades y posiciones del sistema y como afecta el flujo de las moléculas.
- Visualizar por medio de matplotlib la posición de cada uno de las partículas dentro de los carriles y guardar el correspondiente archivo para realizar una película al final del cálculo.
- Aumentar el número de partículas presentes.
- Determinar cuantas partículas llegan a congestionar o colapsar un número n de carriles.

- Visualizar el efecto de presa fantasma en una carretera circular.
- Visualizar/simular el flujo correcto de las partículas.
- Crear documentación.

## 4. Marco teórico

### 4.1. Dinámica Molecular

Técnica computacional que simula el movimiento de partículas (átomos o moléculas) mediante integración de las leyes de Newton. Es muy útil para hacer modelos de Física del estado sólido, química computacional, biofísica, entre otras aplicaciones.

### 4.2. Implementación Computacional

La simulación de dinámica molecular fue implementada en **Python 3**, utilizando un enfoque modular y estructurado. El proyecto hace uso de herramientas y bibliotecas científicas comunes, que permiten la ejecución, análisis y visualización interactiva de los resultados.

#### 4.2.1. Lenguaje y herramientas

- **Python 3:** Lenguaje de programación interpretado, de alto nivel, ideal para simulaciones científicas debido a su legibilidad y vasta comunidad.
- **Jupyter Notebook:** Entorno interactivo que permite integrar código, visualizaciones y explicaciones teóricas en un mismo archivo.
- **NumPy:** Biblioteca para manipulación eficiente de arreglos multidimensionales y operaciones vectorizadas.
- **Matplotlib:** Biblioteca de visualización que permite generar gráficos estáticos y animaciones bidimensionales.

### 4.3. Movimiento Browniano y Transiciones de Fase Dinámicas

Las transiciones de fase dinámicas (DPT) son fenómenos interesantes donde el comportamiento del sistema cambia cualitativamente a medida que se varían algunas

condiciones externas o parámetros. Por ejemplo, el agua se convierte en hielo cuando se congela, lo que indica una DPT. Estas fluctuaciones atípicas son eventos raros pero pueden tener impactos significativos en la estabilidad de un sistema o llevar a resultados inesperados. Al estudiar el movimiento browniano, los científicos observan cómo estas fluctuaciones atípicas se relacionan con las DPT, proporcionando información sobre fenómenos desde el movimiento de pequeños motores moleculares hasta el comportamiento de sistemas más grandes cerca de puntos críticos. [2]

#### 4.4. Fluctuaciones fuera del equilibrio en sistemas de partículas

El estudio de las fluctuaciones fuera del equilibrio en sistemas de partículas en interacción implica analizar el comportamiento de partículas que pueden realizar un movimiento dirigido mediante autopropulsión. Estos sistemas rompen el equilibrio detallado a nivel local y presentan fluctuaciones significativas. La investigación se centra en comprender el papel de las condiciones iniciales sobre las fluctuaciones, los efectos de enfriar las direcciones de sesgo inicial, y el comportamiento cerca y más allá de la separación de fase inducida por la motilidad. El análisis generalmente comienza a partir de un perfil de densidad inicial escalonado y considera el promedio sobre realizaciones iniciales con densidad media. Este enfoque permite una comprensión más profunda de la dinámica y las fluctuaciones en estos sistemas. [3]

#### 4.5. Formación de patrones en un sistema fuera del equilibrio

La formación de patrones en sistemas fuera del equilibrio es un fenómeno fascinante que ha sido objeto de estudio en diversas disciplinas. Este proceso se caracteriza por la interacción no lineal entre componentes del sistema, lo que permite la aparición de estructuras complejas y patrones espaciales. Ejemplos de sistemas fuera del equilibrio incluyen reacciones químicas oscilantes, patrones de convección en fluidos calentados y sistemas de ondas químicas [4]. Estos sistemas pueden exhibir comportamientos emergentes que no son predecibles a partir de sus componentes individuales, mostrando cómo el flujo de energía puede inducir autoorganización espontánea. [5]

### 5. Metodología de Simulación Computacional

La presente metodología describe la implementación y ejecución de un modelo de simulación de tráfico vehicular basado en el **Modelo de Conductor Inteligente (IDM)** en una topología circular. El objetivo es estudiar la formación espontánea de atascos (congestión) inducidos por una perturbación periódica.

#### 5.1. Modelo de Seguimiento de Vehículos (IDM)

La dinámica de cada vehículo  $i$  se rige por el IDM, que calcula la aceleración  $a_i$  en función de su velocidad  $v_i$ , la distancia efectiva al vehículo precedente  $g_i$ , y la velocidad del vehículo precedente  $v_{i-1}$ .

##### 5.1.1. Ecuación Fundamental del IDM

La aceleración  $a_i$  está definida por:

$$a_i = A_{\max} \left[ 1 - \left( \frac{v_i}{V_0} \right)^4 - \left( \frac{s^*}{g_i} \right)^2 \right]$$

Donde  $s^*$  es la distancia de seguridad deseada, calculada como:

$$s^* = S_0 + \max \left[ 0, v_i T_{\text{headway}} + \frac{v_i(v_i - v_{i-1})}{2\sqrt{A_{\max} B_{\text{decel}}}} \right]$$

##### 5.1.2. Integración Temporal

La posición  $s_i$  y la velocidad  $v_i$  se actualizan en cada paso de tiempo  $\Delta T$  ( $DT = 0,05$  s) mediante el método de integración de Euler, con la condición de frontera periódica:

$$\begin{aligned} v_i(t + \Delta t) &= \max(0, v_i(t) + a_i(t) \cdot \Delta t) \\ s_i(t + \Delta t) &= (s_i(t) + v_i(t + \Delta t) \cdot \Delta t) \quad (\text{mód CIRC}) \end{aligned}$$

#### 5.2. Configuración del Escenario de Simulación

- **Topología:** Circuito cerrado de longitud CIRC = 230,0 m.
- **Vehículos y Densidad:** Se utiliza un número fijo de vehículos  $N = 20$ . La densidad de vehículos es constante  $\rho \approx 0,087$  veh/m.
- **Parámetros Clave del IDM:** La velocidad deseada es  $V_0 = 15,0$  m/s, la máxima aceleración

$A_{\max} = 0,8 \text{ m/s}^2$ , y la deceleración confortable  $B_{\text{decel}} = 4,0 \text{ m/s}^2$ .

- **Perturbación del Líder:** El vehículo  $i = 0$  introduce una perturbación periódica, frenando agresivamente durante  $\text{STOP\_DURATION} = 6,0 \text{ s}$ . El ciclo de repetición de la perturbación es de  $\text{REPEAT\_INTERVAL} = 15,0 \text{ s}$ .

### 5.3. Procedimiento de Ejecución

La simulación se ejecuta por un tiempo total de  $\text{SIM\_TIME} = 1200,0 \text{ s}$ . Las condiciones iniciales establecen a todos los vehículos uniformemente espaciados y moviéndose a la velocidad deseada  $V_0$ . El análisis posterior se enfoca en el **Diagrama Espacio-Tiempo** para visualizar la propagación de las ondas de choque y la transición de fase hacia el tráfico congestionado.

La simulación del flujo de tráfico y la formación de atascos se implementa utilizando el lenguaje de programación Python, adoptando el paradigma de Programación Orientada a Objetos (POO). Este enfoque permite modelar de manera intuitiva las entidades físicas del sistema (vehículos y el entorno de la simulación) como clases interconectadas, facilitando la gestión de la complejidad y la escalabilidad del código.

## 6. Resultados y Análisis de la Transición de Fase

La simulación IDM, ejecutada en una topología circular con una densidad  $\rho \approx 0,087 \text{ veh/m}$ , demuestra la aparición recurrente y la propagación de atascos (congestión) inducidos por una perturbación cíclica en el vehículo líder.

### 6.1. Dinámica de la Congestión

El fenómeno central observado es la **transición de fase** del tráfico:

- **Onda de Choque Inversa:** Cada evento de frenado en el vehículo líder ( $i = 0$ ) genera una **onda de choque** que se propaga **río arriba** (en sentido contrario al movimiento). Esta onda se caracteriza por una brusca disminución de la velocidad, llegando hasta  $\langle v \rangle \approx 0 \text{ m/s}$  en el corazón del atasco.
- **Amplificación de la Perturbación:** Debido a la alta densidad y los parámetros del IDM (especialmente el tiempo de seguridad  $T_{\text{headway}} = 1,6 \text{ s}$ ,

la perturbación inicial se amplifica. Cada vehículo subsiguiente frena más severamente y por un período más largo que el vehículo precedente, lo que facilita la solidificación del estado congestionado.

## 6.2. Análisis Cuantitativo

### 6.2.1. Diagrama Espacio-Tiempo

El diagrama  $(s_i, t)$  es fundamental. Las **regiones de alta pendiente** corresponden al flujo libre, mientras que las **regiones verticales** confirman la formación de atascos. La pendiente de la frontera entre estas regiones proporciona una estimación de la velocidad de propagación de la onda de choque,  $\omega_{\text{shock}}$ , típicamente negativa.

### 6.2.2. Diagrama de Flujo Fundamental

Al analizar la velocidad promedio  $\langle V \rangle$  versus el espacio promedio  $\langle g \rangle$ , se observa la característica **histeresis** del modelo IDM. El sistema transita del estado de flujo libre a un estado de flujo congestionado con una tasa de flujo reducida, confirmando que la densidad utilizada es crítica y está cerca del punto de inestabilidad  $\rho_{\text{crítica}}$ .

## 7. Implementación

A continuación se presenta el código de la simulación de tráfico.

```
import numpy as np
import matplotlib.pyplot as plt
import math

class TrafficSimulation:
    def __init__(self):
        # Parámetros Globales
        self.N = 20
        self.CIRC = 230.0
        self.DT = 0.05
        self.SIM_TIME = 1200.0

        # Parámetros Físicos
        self.L_VEHICLE = 4.0
        self.S0 = 2.0

        # Modelo IDM
        self.V0 = 15.0
        self.A_MAX = 0.8
        self.B_DECEL = 4.0
```

```

self.T_HEADWAY = 1.6

# Eventos de parada
self.STOP_DURATION = 6.0 # Duración del frenazo (suficiente para parar la fila)
self.FIRST_STOP = 2.0
self.REPEAT_INTERVAL = 15.0

# Estado inicial
self.s = np.linspace(0, self.CIRC, self.N, endpoint=False) # Línea circular
self.v = np.ones(self.N) * self.V0

# Visualización
self.fig, self.ax = None, None

# FUNCIONES DEL MODELO

def get_gap(self, i, s_curr):
    """Distancia al vehículo de adelante."""
    leader_idx = (i - 1 + self.N) % self.N
    dist = s_curr[leader_idx] - s_curr[i]
    if dist < 0:
        dist += self.CIRC
    return dist - self.L_VEHICLE

def idm_accel(self, v_curr, v_leader, gap):
    """Aceleración IDM."""
    delta_v = v_curr - v_leader

    s_star = self.S0 + max(
        0.0,
        v_curr * self.T_HEADWAY + (v_curr * delta_v) / (2 * np.pi * self.A_MAX * self.B_DECEL))
    effective_gap = max(0.01, gap)

    accel = self.A_MAX * (1 - (v_curr / self.V0)**4 - (s_star - effective_gap)**2)
    return accel

def leader_stopped(self, t):
    """Determina si el líder debe detenerse."""
    time_since_start = t - self.FIRST_STOP
    if time_since_start >= 0:
        cycle_pos = time_since_start % self.REPEAT_INTERVAL
        return cycle_pos < self.STOP_DURATION
    return False

def run_step(self, t):
    s_new = np.copy(self.s)
    v_new = np.copy(self.v)

    accel = np.zeros(self.N)
    p0_is_stopped = self.leader_stopped(t)

    # Cálculo de aceleraciones
    for i in range(self.N):
        if i == 0:
            # LÍDER
            if p0_is_stopped:
                if self.v[i] > 0:
                    accel[i] = -10.0
                else:
                    self.v[i] = 0.0
            else:
                accel[i] = self.A_MAX * (1 - (self.v[i] / self.V0)**4)
        else:
            leader_idx = (i - 1 + self.N) % self.N
            gap = self.get_gap(i, self.s)
            leader_v = self.v[leader_idx]
            accel[i] = self.idm_accel(self.v[i], leader_v, gap)

    # Integración (Euler)
    for i in range(self.N):
        if i == 0 and p0_is_stopped and v_new[i] == 0:
            continue

        v_new[i] = max(0.0, self.v[i] + accel[i] * self.DT)
        s_new[i] = (self.s[i] + v_new[i] * self.DT) % self.CIRC

        if i != 0:
            gap = self.get_gap(i, self.s)
            if (v_new[i] * self.DT) > (gap - 0.5):
                v_new[i] = max(0.0, (gap - 0.5) / self.DT)
                s_new[i] = (self.s[i] + v_new[i] * self.DT) % self.CIRC

    self.s = s_new
    self.v = v_new

# * VISUALIZACIÓN
def setup_draw(self):
    plt.ioff()
    self.fig, self.ax = plt.subplots(figsize=(7, 7))

    def draw_frame(self, t):
        R = self.CIRC / (2 * np.pi)
        self.ax.clear()
        self.ax.set_box_aspect(1)
        self.ax.set_xlim(-R * 1.3, R * 1.3)
        self.ax.set_ylim(-R * 1.3, R * 1.3)
        self.ax.axis('off')

        circle = plt.Circle((0, 0), R, color='gray', fill=False)
        self.ax.add_artist(circle)

        angles = self.s / R
        X = R * np.cos(angles)
        Y = R * np.sin(angles)

        colors = ['blue' if i == 0 else '#d62728' for i in range(self.N)]
        self.ax.scatter(X, Y, s=100, c=colors, edgecolors='black')

        self.ax.set_title(f"Simulación de Tráfico\nTiempo: {t:.2f} s")
        plt.pause(0.001)

    # LOOP PRINCIPAL
    def run(self):
        while True:
            draw_frame(self, self.t)
            self.t += self.REPEAT_INTERVAL

```

```

    self.setup_draw()

    t = 0.0
    print("Iniciando simulación...")
    print(f"PO frena a los {self.FIRST_STOP}s y

    while t < self.SIM_TIME:
        self.run_step(t)

        if int(t / self.DT) % 2 == 0:
            self.draw(t)

        t += self.DT

    plt.ioff()
    plt.show()

# MAIN
if __name__ == "__main__":
    sim = TrafficSimulation()
    sim.run()

```

## 8. Conclusiones

El estudio, basado en la simulación del Modelo de Conducto Inteligente (IDM) bajo condiciones de alta densidad, confirma la aparición de **fenómenos colectivos** inherentes a la dinámica del tráfico vehicular.

### ■ Inestabilidad del Flujo y Transición de Fase:

La simulación demuestra que el sistema es intrínsecamente inestable. Una simple perturbación periódica de velocidad en el vehículo líder es suficiente para forzar una **transición de fase** completa, moviendo el sistema del estado de flujo libre a un estado de tráfico congestionado. Esto valida la predicción de que la formación de atascos es un fenómeno de **auto-organización** y no requiere de obstáculos fijos.

### ■ Propagación de Ondas de Choque:

Se confirma la formación recurrente y la propagación de **ondas de choque** ( $\omega_{\text{shock}}$ ) en dirección opuesta al movimiento de los vehículos. El IDM reproduce la **amplificación** de la perturbación, donde la desaceleración requerida es mayor para los vehículos posteriores, solidificando así el atasco.

### ■ Evidencia de Histeresis:

El análisis cuantitativo (típicamente a través del Diagrama de Flujo Fundamental) revela la presencia de **histeresis**. Esto significa que la tasa de flujo que el sistema puede sostener en el estado congestionado es significativamente menor que la del estado libre a la misma

densidad, lo cual tiene implicaciones críticas en el rendimiento y la gestión de la infraestructura vial.

En resumen, los resultados obtenidos del modelo IDM corroboran la comprensión de que la congestión es una manifestación de la inestabilidad de la dinámica vehicular, regida por los parámetros de seguridad y reacción de los conductores.

## 9. Bibliografía

### Referencias

- [1] Sugiyama, Fukui, Kikuchi, Hasebe, Nakayama, Nishinari, et al. Traffic jams without bottlenecks: Experimental evidence for the physical mechanism of the formation of a jam. *Science*. 2002;288(5469):1331-5. Available from: [https://www.researchgate.net/publication/228692769\\_Traffic\\_jams\\_without\\_bottlenecks\\_-\\_experimental\\_evidence\\_for\\_the\\_physical\\_mechanism\\_of\\_theFormation\\_of\\_a\\_jam](https://www.researchgate.net/publication/228692769_Traffic_jams_without_bottlenecks_-_experimental_evidence_for_the_physical_mechanism_of_theFormation_of_a_jam).
- [2] Liu J, Aden NM, Sarker D, Song C. El Sorprendente Mundo de las Transiciones de Fase Dinámicas. SciSimple. 2025 mar. Available from: <https://scisimple.com/es/articles/2025-03-24-el-sorprendente-mundo-de-las-transiciones-de-fase-dinamicas--a9p0n66>.
- [3] Milton, Menezes. Non-equilibrium fluctuations of interacting particle systems. arXiv preprint arXiv:181009526. 2018. Available from: <https://arxiv.org/pdf/1810.09526.pdf>.
- [4] Ricardo R. Dinámica de Sistemas Químicos Fuera del Equilibrio: Fundamentos y Aplicaciones. eStudyando. 2025 may. Available from: <https://estudyando.com/dinamica-de-sistemas-quimicos-fuera-del-equilibrio-fundamentos-y-aplicaciones/>.
- [5] Lab A. Termodinámica de no equilibrio. Academia Lab. 2023. Available from: <https://academia-lab.com/enciclopedia/termodinamica-de-no-equilibrio/>.