



# UNIVERSIDAD NACIONAL AUTONOMA DE MÉXICO

---

FACULTAD DE CIENCIAS

## COMPARACIÓN ENTRE UN OSCILADOR DE DENSIDAD Y UN OSCILADOR PEATONAL

T E S I S

QUE PARA OBTENER EL TÍTULO DE:  
FÍSICO

P R E S E N T A:  
RENÉ MORA MAYA

DIRECTOR DE TESIS:  
M. C. SERGIO HERNÁNDEZ ZAPATA

CIUDAD UNIVERSITARIA, CD. MX., 2023





Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

1. Datos del alumno

Mora

Maya

René

55 23 28 39 36

Universidad Nacional Autónoma de

México

Facultad de Ciencias

Física

304100008

2. Datos del tutor

M. en C.

Sergio

Hernández

Zapata

3. Datos del sinodal 1

Dr.

Pedro Eduardo

Miramontes

Vidal

4. Datos del sinodal 2

Dr.

Marcelo

Del Castillo

Mussot

5. Datos del sinodal 3

Dra.

Norma Yanet

Sánchez

Torres

6. Datos del sinodal 4

Dr.

Erick Javier

López

Sánchez

7. Datos del trabajo escrito

Comparación entre un oscilador de densidad y  
un oscilador peatonal

80 p

2023

# Comparación entre un oscilador de densidad y un oscilador peatonal

Mora Maya René

FACULTAD DE CIENCIAS, UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
(UNAM), AV. UNIVERSIDAD 3000, CIRCUITO EXTERIOR S/N, COYOACÁN, CD.  
UNIVERSITARIA, 04510, CIUDAD DE MÉXICO.

*A mi Princesa que me estuvo agobiando  
todo el tiempo para culminar este trabajo.*

Como dice Octavio Paz, “Si yo dejase hablar a mis sentimientos únicamente, estas palabras serían una larga, interminable, frase de gratitud...” Para cada uno de los que colaboraron en este proyecto, durante este largo tiempo, iniciando por mi tutor M. C. Sergio Hernández Zapata, al Dr. Gerardo Ruiz Chavarría quién me recibió y patrocinó en el Taller de Fluidos y a los doctores Erick, Norma, Pedro Miramontes y Marcelo del Castillo que dedicaron su tiempo en la revisión de esta tesis.

Esta tesis se realizó con el apoyo de la DGAPA-UNAM dentro del proyecto PAPIIT IN113621 “Transporte de partículas, convección y vorticidad”.

## Resumen

En este trabajo se realizó una comparación entre un oscilador de densidad y un oscilador peatonal.

El oscilador de densidad se desarrolló por medio de un software de código abierto llamado Gerris, este software desarrolla las ecuaciones de Navier-Stokes por el método de volumen finito.

El oscilador peatonal se desarrolló con el uso del modelo de Dirk Helbing y Péter Molnár que trata sobre la dinámica de interacción entre peatones. El modelo se trabaja numéricamente por medio de varios programas creados con el lenguaje de programación Julia-1.6.1.

Con el uso de Julia y sus paqueterías se analizaron los datos de los programas creados con este mismo y también los hechos con Gerris para hacer animaciones.

Ambos experimentos numéricos se compararon usando números adimensionales en la frecuencia, periodo y el número de Strouhal.

# Índice general

Resumen	III
Capítulo 1. Introducción	1
1.1. Oscilador de Densidad	1
1.2. Modelo de Helbing y Molnár sobre la dinámica de peatones	3
1.3. Análisis General	7
Capítulo 2. Inestabilidad de Rayleigh-Taylor	9
Capítulo 3. Simulación con Gerris	18
3.1. Método de Volumen Finito	18
3.2. Gerris y la Inestabilidad de Rayleigh-Taylor	18
3.3. Gerris y el Oscilador de Densidad	21
Capítulo 4. Lenguaje de programación Julia y el oscilador peatonal	24
4.1. Peatones cruzando una calle	24
4.2. Peatones cruzando una puerta(oscilador peatonal)	28
Capítulo 5. Comparación entre los osciladores de densidad y peatonal	32
5.1. El parámetro adimensional $\alpha$	32
5.2. El parámetro adimensional $\alpha$ para ambos osciladores	34
5.3. Comparación	35
Capítulo 6. Conclusiones	41
Referencias	43
Apéndice A. Desarrollo de las ecuaciones para el modelo de Helbing y Molnár	44
Apéndice B. Programa en Gerris	47
Apéndice C. Programa en Julia (Calle con peatones).	52
Apéndice D. Programa en Julia (Peatones cruzando una puerta).	63
Apéndice E. Peatones con movimiento ortogonal.	79

## Introducción

La *Inestabilidad de Rayleigh-Taylor* es un fenómeno que se presenta cuando un fluido está sobre otro fluido menos denso. En esta situación se produce, por medio de la interacción entre los dos fluidos, caminos en donde el fluido más denso baja alternándose con caminos en donde el fluido menos denso sube. En el capítulo 2 se da una descripción más detallada de esta inestabilidad, llamada Inestabilidad de Rayleigh-Taylor.

Si se colocan barreras que separen a ambos fluidos, se deja un pequeño orificio de contacto entre estos, a través de la barrera, y se agrega un tercer fluido en la cima que sea mucho menos denso, se tiene un caso especial de la *Inestabilidad de Rayleigh-Taylor* donde en lugar de observar caminos, se aprecia como un solo fluido atraviesa el orificio, este flujo se detiene paulatinamente y comienza un flujo del otro fluido en la dirección opuesta. Este segundo flujo se detiene paulatinamente y de nuevo es el primer fluido el que fluye en el sentido inicial. A este fenómeno se le conoce como oscilador de densidad.

### 1.1. Oscilador de Densidad

El esquema más representativo del fenómeno se muestra en la figura 1.1 donde se tiene una representación en dos dimensiones, un recipiente de radio  $c$  contiene un fluido de densidad  $\rho_w$ , dentro de este se coloca otro recipiente cilíndrico de radio interno  $b$  que contiene un fluido de densidad  $\rho_s$ , en el centro del recipiente pequeño parte inferior se tiene un orificio de radio  $a$ .

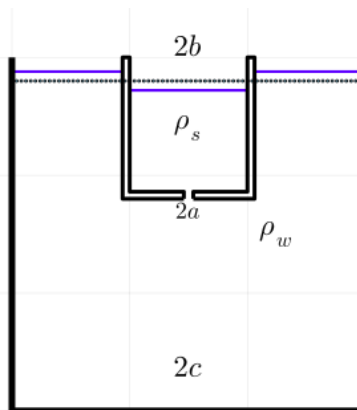


FIGURA 1.1.  $c$  radio del recipiente grande,  $b$  radio interno del recipiente interior,  $a$  radio del orificio,  $\rho_s$  densidad del fluido más denso y  $\rho_w$  densidad del fluido menos denso.

El experimento funciona de la siguiente manera, primero se colocan las superficies libres de ambos recipientes a la misma altura, después se destapa el orificio de radio  $a$ , debido a la diferencia de presiones en el orificio comienza a bajar el fluido más denso del recipiente pequeño, como se ve en la línea continua azul (figura 1.1) la superficie libre del recipiente pequeño baja y la del recipiente grande sube, llega un momento en que la presión en el orificio se invierte y el flujo hacia abajo empieza a frenarse hasta que eventualmente se detiene y el fluido menos denso empieza a moverse por el orificio hacia arriba. Eventualmente este flujo también se detiene y empieza, como al principio. Este es el fenómeno conocido como oscilador de densidad. El proceso continúa hasta que, en el caso que no sean inmiscibles, los dos fluidos adquieren la misma densidad.

El primero en trabajar con el oscilador de densidad fue Seelye Martin [3] en el año de 1970, a diferencia de la figura 1.1, él utilizó jeringas de volúmenes diferentes y agujas de distinto radio 1.2 como recipientes internos, su oscilador es conocido como oscilador salino ya que utiliza agua en el recipiente grande y agua salada en el recipiente pequeño.

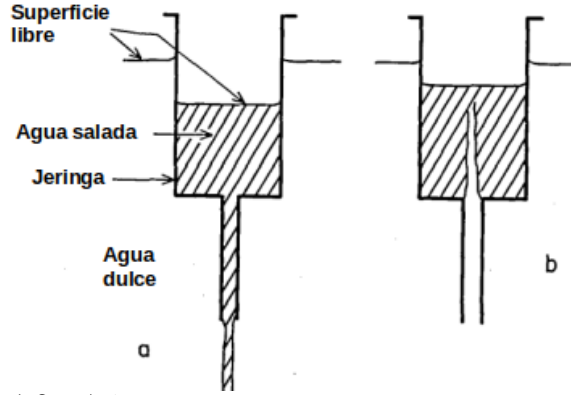


FIGURA 1.2. a) Agua salada fluye al recipiente de agua dulce. b) Agua dulce fluye a la jeringa con agua salada. [3].

El oscilador salino es la forma más sencilla de trabajar con un oscilador de densidad debido al fácil acceso de los fluidos. La mayoría de los investigadores que han trabajado con el oscilador salino [3, 4, 9, 10, 11], han utilizado en sus modelos para describir el movimiento de la superficie libre del recipiente interno las ecuaciones de continuidad (1.1) y Navier-Stokes (1.2) [5]. Separan el movimiento de la superficie libre en dos ecuaciones, una para cuando baja el fluido por el orificio  $a$  y otra para cuando sube.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \quad (1.1)$$

$$\rho \frac{\partial \mathbf{v}}{\partial t} + \rho (\mathbf{v} \cdot \nabla) \mathbf{v} = -\nabla p + \mu \nabla^2 \mathbf{v} + \rho \mathbf{f} \quad (1.2)$$

Se ha trabajado con modelos numéricos con varios osciladores acoplados [10], es decir, un recipiente grande conteniendo a varios pequeños. También se ha presentado una utilización del oscilador salino para medir la densidad [11], que

depende de la geometría del experimento y del punto medio de la oscilación de la superficie libre que tiene una precisión bastante alta, siempre y cuando se elijan los parámetros adecuados.

## 1.2. Modelo de Helbing y Molnár sobre la dinámica de peatones

Desde hace mucho tiempo el hombre ha observado el comportamiento de grandes multitudes de insectos como abejas y hormigas, también de mamíferos como las migraciones de ñus por el continente Africano o aves que vuelan distancias enormes. Se ha encontrado en el movimiento de estos animales ciertos patrones como filas en ambas direcciones en el movimiento de las hormigas, formaciones en “V” cuando las aves vuelan o también ciertos patrones en forma de elipses cuando huyen los ñus de sus depredadores. Aunque el comportamiento de las personas es más complejo, se mueven a su propio ritmo, se detienen, cambian de dirección. Se ha observado que el movimiento de grandes multitudes (por ejemplo, la entrada y salida en estadios, la circulación en el metro, calles muy transitadas, etc.) está condicionado a la interacción entre todas las personas en vez del deseo personal de cada individuo [7].

Alessandro Corbetta y Federico Toschi [8] proponen una escala considerando el tamaño y el intervalo de tiempo en el que se observan las multitudes de peatones. Los elementos de su escala son: escala submicroscópica, se describen variables como la posición, orientación del cuerpo y la cabeza como función del tiempo, para cada peatón; escala microscópica, se describen variables como la posición y velocidad para cada peatón, con una precisión de centímetros y decenas de segundos; escala mesoscópica, se describen las variables de los peatones como grados de libertad en un sentido estadístico, se compara con la teoría cinética de los gases; escala macroscópica, los peatones se describen como un continuo donde se desarrollan densidad y campo de velocidades dependientes del tiempo, la descripción comparte similitudes con la hidrodinámica; escala supermacroscópica, se describe a las multitudes de peatones por medio de ecuaciones de estado que conectan distintas cantidades microscópicas como promedio de velocidades y densidad. En esta tesis se desarrolla el Modelo de Helbing y Molnár que se encuentra en una escala microscópica donde cada persona tiene asociada una posición y una velocidad.

Los científicos Dirk Helbing y Péter Molnár [2] crearon un modelo que describe la interacción de grandes multitudes de personas al cuál llamaron *modelo de fuerza social para la dinámica de peatones*. A cada persona se le nombra como peatón.

Este modelo describe la interacción entre un peatón, los peatones que lo rodean y su entorno por medio de una ecuación diferencial de movimiento y una ecuación que condiciona la velocidad máxima a la que se mueve el peatón.

El peatón está sujeto a tres fuerzas. La primera corresponde a una fuerza de repulsión debida a la incomodidad que siente al estar muy cerca de otro. Si el peatón siente cierta atracción hacia otro peatón o a algún lugar ó evento, por ejemplo, le llama la atención un artista o un aparador, éste fenómeno es descrito por una fuerza de atracción. Por último, el peatón tiene que evitar dañarse con obstáculos que existen en las calles, por ejemplo, paredes, árboles, etc., la situación se describe por una fuerza de repulsión.

En éste trabajo de tesis, por simplicidad sólo se toman en cuenta las fuerzas de repulsión entre peatones y entre los obstáculos que debe evitar.

### Fuerza de aceleración que mueve al peatón

El peatón desea llegar a un punto de destino que se identificará con el vector  $\mathbf{r}_\alpha^k$ , el vector unitario en la dirección deseada por el peatón se expresa de la siguiente manera

$$\mathbf{e}_\alpha(t) = \frac{\mathbf{r}_\alpha^k - \mathbf{r}_\alpha(t)}{\|\mathbf{r}_\alpha^k - \mathbf{r}_\alpha(t)\|} \quad (1.3)$$

$\mathbf{r}_\alpha(t)$  es la posición del peatón  $\alpha$  al tiempo  $t$ . El peatón trata de moverse hacia su punto de destino con la velocidad más confortable para él,  $\mathbf{v}_\alpha^0(t) = v_\alpha^0 \mathbf{e}_\alpha(t)$ . Como tiene que evitar peatones y obstáculos, su velocidad real será  $\mathbf{v}_\alpha(t)$ . Helbing y Molnár describen una fuerza de aceleración que debe guiar al peatón hacia la dirección que desea y depende de la velocidad a la que desea moverse  $\mathbf{v}_\alpha^0$ , la velocidad real  $\mathbf{v}_\alpha$  y un tiempo de relajación  $\tau_\alpha$

$$\mathbf{F}_\alpha^0(\mathbf{v}_\alpha, \mathbf{v}_\alpha^0) = \frac{1}{\tau_\alpha} (v_\alpha^0 \mathbf{e}_\alpha - \mathbf{v}_\alpha). \quad (1.4)$$

El tiempo de relajación describe que tan rápido llega a la velocidad que desea.

### Fuerza de interacción entre peatones

El movimiento del peatón( $\alpha$ ) depende de los demás peatones( $\beta$ ) y de la dirección que éstos tengan.  $\alpha$  se siente más incomodo entre más cerca se encuentre de  $\beta$ , entonces se necesita una fuerza de repulsión que aumente con la cercanía de  $\beta$  y también dependa de su dirección

$$\mathbf{f}_{\alpha\beta}(\mathbf{r}_{\alpha\beta}) = -\nabla_{\mathbf{r}_{\alpha\beta}} V_{\alpha\beta}[b(\mathbf{r}_{\alpha\beta})] \quad (1.5)$$

$V_{\alpha\beta}(b)$  es una función monótona decreciente de  $b$  con líneas equipotenciales que forman una elipse cuyo semieje mayor está en la dirección de movimiento. Como se observa en la figura 1.3, la distancia del paso de  $\beta$  es  $2c$  la distancia entre los focos, la distancia entre el peatón  $\alpha$  al peatón  $\beta$  sumado a la distancia entre  $\alpha$  y el paso de  $\beta$  es  $2a$ , de relación

$$\begin{aligned} a^2 &= b^2 + c^2 \\ 4a^2 &= 4b^2 + 4c^2 \\ (2b)^2 &= (2a)^2 - (2c)^2 \end{aligned}$$

entonces  $b$ , es el semi-eje menor de la elipse, es

$$b = \frac{1}{2} \sqrt{(\|\mathbf{r}_{\alpha\beta}\| + \|\mathbf{r}_{\alpha\beta} - v_\beta \Delta t \mathbf{e}_\beta\|)^2 - (v_\beta \Delta t)^2} \quad (1.6)$$

donde  $\mathbf{r}_{\alpha\beta} = \mathbf{r}_\alpha - \mathbf{r}_\beta$ ,  $v_\beta \Delta t$  es la longitud del paso del peatón  $\beta$ .

### Fuerza de interacción del peatón con las fronteras

El peatón tiene que evitar objetos como paredes, edificios, postes, bardas, etc., entre más cerca de éstos se encuentre más atención tendrá para poder evitar lastimarse. Se necesita una fuerza de repulsión que aumente cuando la distancia a las fronteras sea más corta

$$\mathbf{F}_{\alpha B}(\mathbf{r}_{\alpha B}) = -\nabla_{\mathbf{r}_{\alpha B}} U_{\alpha B}(\|\mathbf{r}_{\alpha B}\|) \quad (1.7)$$

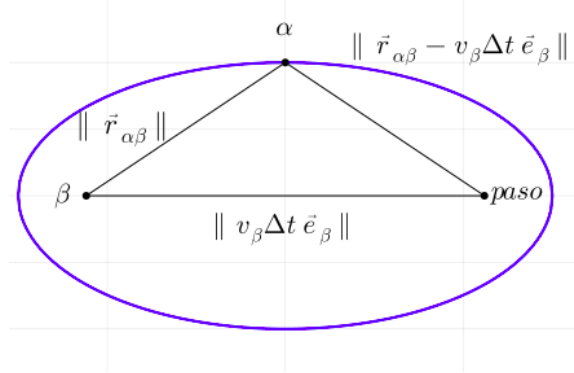


FIGURA 1.3. *Elipse de interacción entre el peatón  $\beta$ , el paso que da y el peatón  $\alpha$ .*

se nombra  $B$  al borde de frontera más cercano al peatón  $\alpha$ ,  $U_{\alpha B}(\|\mathbf{r}_{\alpha B}\|)$  es una función monótona decreciente,  $\mathbf{r}_{\alpha B} = \mathbf{r}_{\alpha} - \mathbf{r}_{\alpha}^B$ , donde  $\mathbf{r}_{\alpha}^B$  representa la localización del borde  $B$  más cercano al peatón  $\alpha$ .

### Ecuación de movimiento del peatón

Para describir la ecuación de movimiento del peatón  $\alpha$  se necesita sumar la interacción que existe entre éste y los demás peatones y fronteras. Sea  $\mathbf{F}_{\alpha}(t)$  la suma de todas las fuerzas representadas en las ecuaciones (1.4), (1.5) y (1.7)

$$\mathbf{F}_{\alpha}(t) = \mathbf{F}_{\alpha}^0(\mathbf{v}_{\alpha}, \mathbf{v}_{\alpha}^0) + \sum_{\beta} \mathbf{f}_{\alpha\beta}(\mathbf{r}_{\alpha\beta}) + \sum_B \mathbf{F}_{\alpha B}(\mathbf{r}_{\alpha B}) \quad (1.8)$$

la ecuación que describe la velocidad del peatón  $\alpha$

$$\frac{d\mathbf{u}_{\alpha}}{dt} = \mathbf{F}_{\alpha}(t) + \text{fluctuaciones} \quad (1.9)$$

el término de fluctuaciones toma en cuenta variaciones aleatorias del comportamiento del peatón, por ejemplo tomar el camino de la izquierda o la derecha al esquivar un obstáculo cuando son equivalentes.

Por sus condiciones físicas el peatón no puede moverse más rápido que una cierta velocidad máxima  $v_{\alpha}^{\max}$ , para representar su movimiento de forma correcta, la función  $g$  asegura que el peatón no sobrepase esta velocidad

$$g\left(\frac{v_{\alpha}^{\max}}{\|\mathbf{u}_{\alpha}\|}\right) = \begin{cases} 1 & \text{si } \|\mathbf{u}_{\alpha}\| \leq v_{\alpha}^{\max} \\ v_{\alpha}^{\max}/\|\mathbf{u}_{\alpha}\| & \text{otro caso} \end{cases} \quad (1.10)$$

la velocidad que resulta de resolver la ecuación (1.9) debe modificarse antes de usarla en la ecuación que describe el movimiento de la siguiente manera

$$\frac{d\mathbf{r}_{\alpha}}{dt} = \mathbf{v}_{\alpha}(t) = \mathbf{u}_{\alpha}(t)g\left(\frac{v_{\alpha}^{\max}}{\|\mathbf{u}_{\alpha}\|}\right). \quad (1.11)$$

Con el objetivo de simplificar el modelo en esta tesis no se consideran fluctuaciones en la ecuación (1.9).

Dirk Helbing y Péter Molnár [2] usan las siguientes ecuaciones potenciales para la descripción de la interacción entre los peatones y sus fronteras

$$V_{\alpha\beta}(b) = V_{\alpha\beta}^0 e^{-b/\sigma}, \quad U(\|\vec{r}_{\alpha B}\|) = U_{\alpha\beta}^0 e^{-\|\vec{r}_{\alpha B}\|/R} \quad (1.12)$$

con los parámetros mostrados en la tabla siguiente

$V_{\alpha\beta}^0$	$2.1\text{m}^2/\text{s}^2$	ancho	10m
$\sigma$	0.3m	largo	50m
$U_{\alpha\beta}^0$	$10\text{m}^2/\text{s}^2$	$\Delta t$	2 s
$R$	0.2m	$\tau_\alpha$	0.5 s

se resuelve la ecuación (1.8) y como resultado obtienen las figuras 1.4, 1.5 y 1.6.

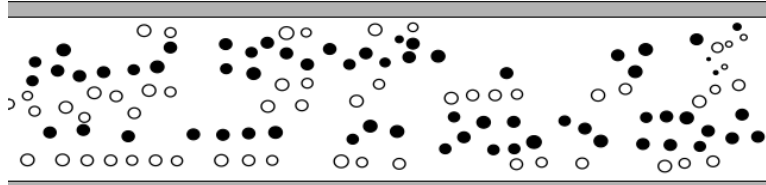


FIGURA 1.4. Se observan líneas de peatones con una dirección de caminata uniforme. Los resultados muestran  $N = 5$  en el camino de 10 m de ancho por 50 m de largo. Círculos vacíos representan a peatones caminando en sentido opuesto a los círculos llenos[2].

La figura 1.4 muestra la alineación de peatones que intentan caminar en la misma dirección pero con sentido opuesto. La figura 1.5 muestra el número de líneas en dependencia con el ancho del camino para una densidad de  $\rho = 0.3$  peatones/ $\text{m}^2$ .

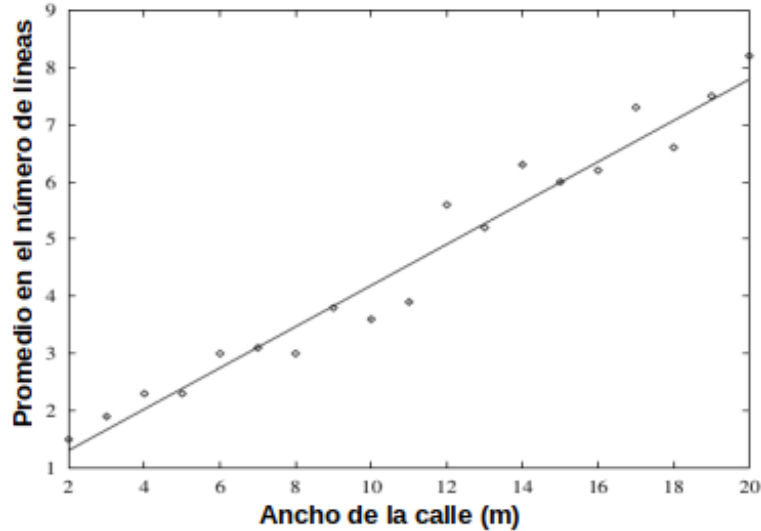


FIGURA 1.5. Número promedio de líneas  $n$  que emergen en el camino con respecto a su ancho ( $N(W) = (0.36\text{m}^{-1})W + 0.59$ ) donde  $W$  es el ancho de la calle en metros.[2]

En el artículo de Helbing y Molnár se muestra que la formación de líneas es consecuencia de la interacción entre peatones y no de las condiciones iniciales.

Por otro lado, la figura 1.6 representa diferentes momentos de dos grupos de peatones que intentan pasar por una puerta estrecha en direcciones opuestas. Una vez que un peatón ha pasado la puerta, los otros que se mueven en la misma dirección son capaces de seguirlo más fácilmente (figura 1.6, a). Sin embargo, el flujo de peatones se detiene después de algún tiempo debido a la oposición del grupo opuesto, ahora los peatones pasan por la puerta en la dirección opuesta (figura 1.6, b).

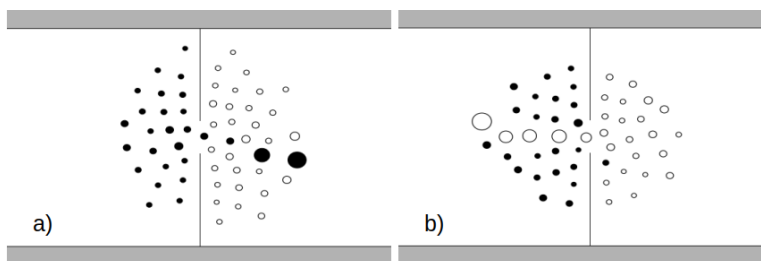


FIGURA 1.6. Si un peatón puede pasar una puerta estrecha, otros peatones con la misma dirección pueden pasar fácilmente mientras los que se mueven en dirección opuesta tienen que esperar. El diámetro de los círculos es una medida de la velocidad actual. a) Peatones negros cruzando la puerta. b) Peatones blancos cruzando la puerta. [2]

### 1.3. Análisis General

Cuando al experimento que trata de la inestabilidad de Rayleigh-Taylor se le colocan fronteras como las mostradas en la figura 1.1 se obtiene el oscilador de densidad. Ahora si a los peatones que se mueven a través de una calle (1.4) de repente se coloca un muro a mitad de la calle y una puerta en el centro de este (1.6), se obtiene el oscilador peatonal.

Entonces, de manera natural el fenómeno que trata sobre la inestabilidad de Rayleigh-Taylor se transforma en el oscilador de densidad añadiendo las condiciones adecuadas y por otra parte, los peatones atravesando una calle cuando se restringe su movimiento a atravesar una puerta crean el oscilador peatonal.

Una parte del trabajo de esta tesis consiste en la simulación de la Inestabilidad de Rayleigh-Taylor por medio del software libre Gerris, una vez obtenido el programa se le implementan las fronteras pertinentes para transformarlo en el oscilador de densidad.

Por otra parte, se implementan las ecuaciones descritas por Dirk Helbing y Peter Molnár (capítulo 1.3) para desarrollar el movimiento de peatones atravesando una calle. El programa se crea con el lenguaje de programación Julia-1.6.1, una vez obtenido se le añade la pared con la puerta y de esta forma se llega al oscilador peatonal.

El trabajo de tesis consiste en la implementación por medio de dos programas numéricos de los fenómenos conocidos como oscilador de densidad y otro al que llamamos oscilador peatonal, ambos fenómenos describen un movimiento oscilatorio. Con los resultados de ambos programas se hará una comparación entre ellos.

La forma de estructurar la tesis será de la siguiente manera:

En el capítulo 2 se describe el fenómeno que trata sobre *La Inestabilidad de Rayleigh-Taylor*.

En el capítulo 3 se implementa un programa numérico que simula el fenómeno de la Inestabilidad de Rayleigh-Taylor y el oscilador de densidad.

En el capítulo 4 se implementa un programa numérico que desarrolla el *Modelo de Dirk Helbing y Péter Molnár* y el oscilador peatonal.

En el capítulo 5 se describe un parámetro adimensional que sirve para establecer una conexión entre el fenómeno de dinámica de fluidos con el descrito por el movimiento de peatones y la manera en que se va a comparar el oscilador de densidad y el oscilador peatonal.

Al final se muestran las conclusiones en el capítulo 6.

El problema principal de éste trabajo de tesis, es observar si existe algún tipo de conexión entre el oscilador de densidad y el oscilador peatonal, para ello se relacionaran varias de sus propiedades entre sí.

## Inestabilidad de Rayleigh-Taylor

Francois Charru [1] describe el fenómeno de la Inestabilidad de Rayleigh-Taylor como:

### Análisis Dimensional

Se tienen dos fluidos de distinta densidad en contacto, uno encima del otro, cada uno de gran profundidad. A la superficie de contacto entre ambos fluidos se le llama la interface. En la figura 2.1a) se muestra la interface de dos fluidos en reposo. Se va a estudiar la estabilidad de la interface en reposo. Utilizando las ideas clásicas de la estabilidad lineal, se considera ahora una perturbación de la interface. Se supone que la perturbación tiene la forma sinusoidal que se muestra en la figura 2.1b). La gráfica de la función que representa la interface con esta perturbación,  $y = f(x, z)$ , solamente depende de la coordenada  $x$  y no de la coordenada  $z$ . Esto permite estudiar la estabilidad de la solución en reposo como un problema bidimensional, debido a que el problema bidimensional es más fácil de estudiar y, como se verá más adelante los resultados básicos de la inestabilidad de Rayleigh-Taylor aparecen con esta aproximación.

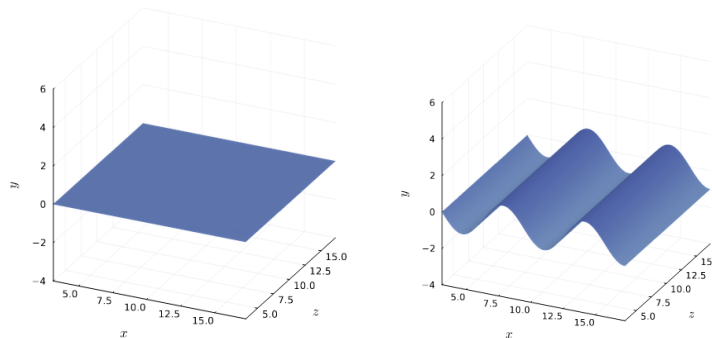


FIGURA 2.1. Interface entre dos fluidos de distinta densidad. El fluido 1 se encuentra en la parte inferior y el 2 en la superior. a) Interface en reposo; b) Interface con perturbación senoidal.

Se corta la figura 2.1 con un plano vertical paralelo al plano  $xy$ , entonces se obtiene la figura 2.2.

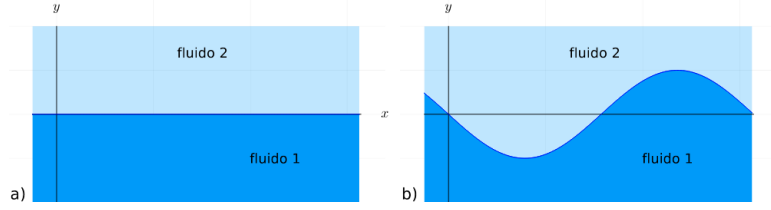


FIGURA 2.2. Plano vertical que corta la figura 2.1. a) Interface en reposo; b) Interface con perturbación senoidal.

Sean  $\rho_1$  y  $\rho_2$ , las densidades del fluido inferior y del superior respectivamente. Sea  $a$  la amplitud y  $k = 2\pi/\lambda$  el número de onda de la perturbación sinusoidal. En la figura (2.3 a) se observa una parte de la onda fuera de la interface en reposo que tiene volumen por unidad de longitud  $V = 2a/k$ , este volumen tiene un peso y una fuerza de flotación, la suma de estas dos fuerzas es la fuerza de Arquímedes

$$F_A = (2a/k)(\rho_2 - \rho_1)g$$

que tiende a reducir la perturbación para valores de  $\rho_1 > \rho_2$ .

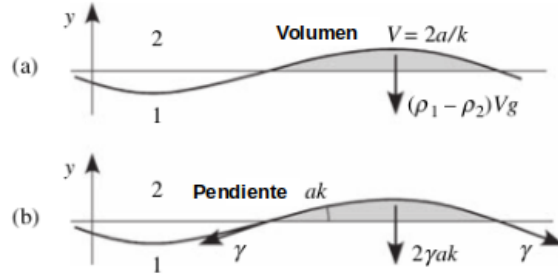


FIGURA 2.3. Fuerzas que actúan sobre la mitad de una onda perturbada de número  $k$  y amplitud  $a$ . a) Fuerza gravitacional neta  $(2a/k)(\rho_2 - \rho_1)g$ ; b) tensión superficial  $\gamma$  y la resultante  $2\gamma ak$ . [1]

En la figura (2.3 b) se observa como actúa la tensión superficial  $\gamma$  sobre la onda, la componente vertical de esta fuerza es  $\gamma ak$ , como se ejerce a ambos lados de la media onda

$$-2\gamma ak.$$

la fuerza de tensión superficial siempre es estabilizante, es decir, se opone a la deformación de la interface y aumenta su valor para números de onda más grandes.

Para valores de  $\rho_1 > \rho_2$  las fuerzas anteriores restauran la interface. El fenómeno se puede comparar con el comportamiento de un resorte con constante  $K = F/a$ , donde  $F = 2\gamma ak + 2(\rho_1 - \rho_2)ga/k$ . Esta fuerza mueve un volumen por unidad de longitud del orden  $1/k^2$ , cuya masa es  $M = (1/k^2)(\rho_1 + \rho_2)$ . La frecuencia de este oscilador es

$$\omega^2 = \frac{K}{M} \sim 2 \frac{\gamma k^3 + (\rho_1 - \rho_2)gk}{\rho_2 + \rho_1}$$

Para el caso  $\rho_1 < \rho_2$ , la gravedad desestabiliza a la interface y la tensión superficial no es suficiente para mantener el equilibrio cuando

$$2(\rho_2 - \rho_1)ag/k - 2\gamma ak > 0,$$

así,

$$k < \frac{1}{\sqrt{\frac{\gamma}{(\rho_2 - \rho_1)g}}}.$$

Con esto se define el número de onda de corte como:

$$k_c = \frac{1}{l_c}, \quad \text{con} \quad l_c = \sqrt{\frac{\gamma}{g(\rho_2 - \rho_1)}} \quad (2.1)$$

a  $l_c$  se le llama longitud de capilaridad. Para números de onda más pequeños que  $k_c$  la interface es inestable.

La configuración de un líquido más denso sobre uno menos denso siempre será inestable, sin embargo, si la interface tiene una longitud menor que  $l_c$  es estable, por ejemplo cuando queda líquido atrapado dentro de un tubo muy estrecho.

### Ecuaciones de Perturbación

Se considera el estado base cuando la interface entre los fluidos está en reposo, la interface se expresará como  $\eta(x, z, t)$ , las ecuaciones que describen este estado son

$$\begin{aligned} \mathbf{U}_{0j} &= \mathbf{0}, \\ \eta_0 &= 0, \quad j = 1, 2 \\ P_{0j}(y) - P_{\text{int}} &= -\rho_j g y \end{aligned} \quad (2.2)$$

donde  $j$  es la etiqueta que se le coloca a cada fluido, el subíndice 0 corresponde al flujo base,  $P_{\text{int}}$  la presión en la interface. Se sigue el método estandar para estudiar la estabilidad del flujo base, se añade a este una perturbación pequeña y se obtiene

$$\begin{aligned} \mathbf{U}_j &= \mathbf{U}_{0j} + \mathbf{u}_j = \mathbf{u}_j, \\ P_j &= P_{0j} + p_j, \\ \eta &= \eta_0 + \eta_p = \eta_p. \end{aligned} \quad (2.3)$$

A partir de la ecuación descrita por Feynman[6]

$$\frac{\partial \boldsymbol{\Omega}_j}{\partial t} + \nabla \times (\boldsymbol{\Omega}_j \times \mathbf{u}_j) = \nu_j \nabla^2 \boldsymbol{\Omega}_j \quad (2.4)$$

donde  $\boldsymbol{\Omega}_j = \nabla \times \mathbf{u}_j$  es el rotacional de la velocidad,  $\nu_j$  la viscosidad cinemática. Como la perturbación es pequeña se consideran únicamente los términos lineales y la ecuación (2.4) queda como

$$\frac{\partial \boldsymbol{\Omega}_j}{\partial t} = \nu_j \nabla^2 \boldsymbol{\Omega}_j$$

además, para simplificar el problema se considera viscosidad nula  $\nu_j = 0$

$$\frac{\partial \boldsymbol{\Omega}_j}{\partial t} = \mathbf{0}$$

de aquí  $\nabla \times \mathbf{u}_j = 0$  y por lo tanto  $\mathbf{u}_j = \nabla \phi_j$ .

Sin pérdida de generalidad, se considera la perturbación en el campo de velocidades como el gradiente de un potencial

$$\mathbf{u}_j = \nabla \phi_j \quad (2.5)$$

De la ecuación de conservación de masa [5]

$$\frac{\partial \rho_j}{\partial t} + \nabla \cdot (\rho_j \mathbf{u}_j) = 0 \quad (2.6)$$

si la densidad  $\rho_j$  es constante (flujo incompresible) la ecuación (2.6) se reduce

a

$$\nabla \cdot \mathbf{u}_j = 0, \quad \Rightarrow \nabla^2 \phi_j = 0$$

entonces

$$\frac{\partial^2 \phi_j}{\partial x^2} + \frac{\partial^2 \phi_j}{\partial y^2} = 0, \quad j = 1, 2 \quad (2.7)$$

La ecuación de Navier-Stokes para un fluido incompresible

$$\rho_j \left[ \frac{\partial \mathbf{u}_j}{\partial t} + (\mathbf{u}_j \cdot \nabla) \mathbf{u}_j \right] = -\nabla P_j - \rho_j g \hat{j} \quad (2.8)$$

el segundo término del lado izquierdo de la ecuación (2.8) es

$$(\mathbf{u}_j \cdot \nabla) \mathbf{u}_j = \mathbf{u}_j \times \nabla \times \mathbf{u}_j + \nabla \left( \frac{u^2}{2} \right) = \nabla \left( \frac{u^2}{2} \right) \quad (2.9)$$

y el último término de la ecuación (2.8)

$$\begin{aligned} \nabla(\rho_j g y) &= \frac{\partial(\rho_j g y)}{\partial x} \hat{i} + \frac{\partial(\rho_j g y)}{\partial y} \hat{j} \\ &= \rho_j g \hat{j} \end{aligned} \quad (2.10)$$

sustituyendo  $\mathbf{u}_j = \nabla \phi_j$  y las ecuaciones (2.9) y (2.10) en (2.8)

$$\rho_j \left[ \frac{\partial}{\partial t} \nabla \phi_j + \nabla \left( \frac{\|\nabla \phi_j\|^2}{2} \right) \right] + \nabla P_j + \nabla(\rho_j g y) = 0 \quad (2.11)$$

entonces

$$\nabla \left( \rho_j \frac{\partial \phi_j}{\partial t} + \frac{\rho_j}{2} \|\nabla \phi_j\|^2 + P_j + \rho_j g y \right) = 0 \quad (2.12)$$

La parte entre paréntesis de la ecuación (2.12) es constante, para simplificar los cálculos la constante es  $P_{\text{int}}$ ,

$$\rho_j \frac{\partial \phi_j}{\partial t} + \frac{1}{2} \rho_j \|\nabla \phi_j\|^2 + P_j(y) + \rho_j g y = P_{\text{int}} \quad (2.13)$$

sustituyendo la ecuación con respecto a la presión (2.3) en (2.13)

$$\rho_j \frac{\partial \phi_j}{\partial t} + \frac{1}{2} \rho_j \|\nabla \phi_j\|^2 + (P_{0j} + p_j) + \rho_j g y = P_{\text{int}}, \quad j = 1, 2 \quad (2.14)$$

y después sustituyendo (2.2) en (2.14)

$$\rho_j \frac{\partial \phi_j}{\partial t} + \frac{1}{2} \rho_j \|\nabla \phi_j\|^2 + p_j = 0, \quad j = 1, 2 \quad (2.15)$$

### Condiciones de frontera

Como los fluidos del problema son muy profundos, la velocidad para regiones muy alejadas de la interface debe disminuir a cero. Por lo tanto, las funciones potenciales para las velocidades de los fluidos deben satisfacer

$$\phi_1 \rightarrow 0 \quad \text{para} \quad y \rightarrow -\infty \quad (2.16a)$$

$$\phi_2 \rightarrow 0 \quad \text{para} \quad y \rightarrow +\infty \quad (2.16b)$$

Se considera que la interface debe satisfacer una función descrita por la ecuación  $H(x, y, z, t) = 0$ , como se observa en la figura 2.4  $\mathbf{n}$  es un vector normal a la superficie

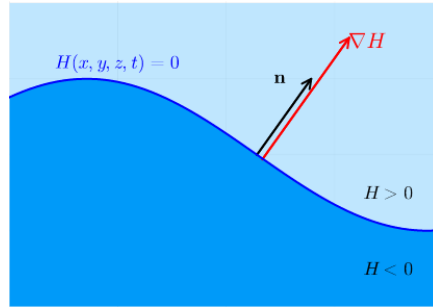


FIGURA 2.4. La normal a la superficie  $H(x, y, z, t) = 0$  es paralela a su gradiente.[1]

La ecuación diferencial construida con la ecuación  $H$  es

$$dH = \frac{\partial H}{\partial t} dt + \nabla H \cdot d\mathbf{r} = 0 \quad (2.17)$$

para un tiempo constante la ecuación (2.17) se reduce

$$\nabla H \cdot d\mathbf{r} = 0 = \mathbf{n} \cdot d\mathbf{r} \quad (2.18)$$

entonces  $\mathbf{n}$  es paralelo a  $\nabla H$ , por lo que  $\mathbf{n}$  puede escribirse como

$$\mathbf{n} = \frac{\nabla H}{\|\nabla H\|} \quad (2.19)$$

sea  $d\mathbf{r} = \mathbf{w} dt$ , con  $\mathbf{w}$  la velocidad de un punto sobre la superficie y se divide la ecuación (2.17) entre  $\|\nabla H\|$

$$\frac{1}{\|\nabla H\|} \frac{\partial H}{\partial t} dt + \mathbf{n} \cdot \mathbf{w} dt = 0$$

entonces

$$\mathbf{w} \cdot \mathbf{n} = -\frac{1}{\|\nabla H\|} \frac{\partial H}{\partial t} \quad (2.20)$$

Aquí se considera un problema en dos dimensiones con una interface definida por

$$H(x, y, t) = y - \eta(x, t) = 0 \quad (2.21)$$

el gradiente de la ecuación (2.21) es

$$\begin{aligned} \nabla H &= \nabla (y - \eta(x, t)) \\ &= \frac{\partial}{\partial x} (y - \eta(x, t)) \hat{i} + \frac{\partial}{\partial y} (y - \eta(x, t)) \hat{j} \\ &= -\frac{\partial \eta(x, t)}{\partial x} \hat{i} + \hat{j} \\ &= \left( -\frac{\partial \eta}{\partial x}, 1 \right) \end{aligned} \quad (2.22)$$

la norma de la ecuación (2.22) es

$$\begin{aligned} \|\nabla H\| &= \sqrt{\left( -\frac{\partial \eta}{\partial x} \right)^2 + 1^2} \\ &= \sqrt{1 + \left( \frac{\partial \eta}{\partial x} \right)^2} \end{aligned} \quad (2.23)$$

la derivada parcial de (2.21) con respecto al tiempo es

$$\begin{aligned} \frac{\partial H}{\partial t} &= \frac{\partial y}{\partial t} - \frac{\partial \eta(x, t)}{\partial t} \\ &= -\frac{\partial \eta}{\partial t} \end{aligned} \quad (2.24)$$

sustituyendo los valores de las ecuaciones (2.24), (2.23) y (2.22) dentro de las ecuaciones (2.20) y (2.19), se tiene

$$\mathbf{n} = \frac{(-\partial \eta / \partial x, 1)}{\sqrt{1 + (\partial \eta / \partial x)^2}} \quad \mathbf{w} \cdot \mathbf{n} = \frac{\partial \eta / \partial t}{\sqrt{1 + (\partial \eta / \partial x)^2}} \quad (2.25)$$

la velocidad normal en la interface debe ser igual a la componente normal en el campo de velocidades

$$\mathbf{U}_j \cdot \mathbf{n} = \mathbf{w} \cdot \mathbf{n} \quad \text{cuando} \quad y = \eta(x, t), \quad j = 1, 2 \quad (2.26)$$

ó de otra forma, de las ecuaciones (2.2), (2.3), (2.5) y (2.26)

$$\frac{\partial \phi_1}{\partial x} \left( -\frac{\partial \eta}{\partial x} \right) + \frac{\partial \phi_1}{\partial y} = \frac{\partial \eta}{\partial t} \quad \text{para} \quad y = \eta \quad (2.27a)$$

$$\frac{\partial \phi_2}{\partial x} \left( -\frac{\partial \eta}{\partial x} \right) + \frac{\partial \phi_2}{\partial y} = \frac{\partial \eta}{\partial t} \quad \text{para} \quad y = \eta \quad (2.27b)$$

La viscosidad es despreciable por lo que el esfuerzo es normal en la interface e igual a  $-P_1 \mathbf{n}$  en el fluido inferior y  $-P_2(-\mathbf{n})$  en el fluido superior. El balance de éstas fuerzas produce la relación Laplace-Young [1]

$$P_1 - P_2 = \frac{\gamma}{R} \quad \text{cuando} \quad y = \eta \quad (2.28)$$

con  $\gamma$  la tensión superficial y  $R^{-1} = \nabla \cdot \mathbf{n}$  el radio de curvatura de la interface

$$\frac{1}{R} = \nabla \cdot \mathbf{n} = \frac{-\partial^2 \eta / \partial x^2}{[1 + (\partial \eta / \partial x)^2]^{3/2}} \quad (2.29)$$

### Linearización, modos normales y la relación de dispersión

Las ecuaciones que describen el fenómeno son

$$\rho_j \frac{\partial \phi_j}{\partial t} + \frac{1}{2} \rho_j (\nabla \phi_j)^2 + p_j = 0 \quad (2.30a)$$

$$\frac{\partial^2 \phi_j}{\partial x^2} + \frac{\partial^2 \phi_j}{\partial y^2} = 0, \quad j = 1, 2 \quad (2.30b)$$

$$-\frac{\partial \phi_j}{\partial x} \frac{\partial \eta}{\partial x} + \frac{\partial \phi_j}{\partial y} = \frac{\partial \eta}{\partial t} \quad \text{para} \quad y = \eta \quad (2.30c)$$

$$P_1 - P_2 = \frac{-\partial^2 \eta / \partial x^2}{[1 + (\partial \eta / \partial x)^2]^{3/2}} \gamma \quad \text{para} \quad y = \eta \quad (2.30d)$$

Si la perturbación en la ecuación (2.3) es muy pequeña

$$\begin{aligned} (\nabla \phi_j)^2 &\rightarrow 0 \\ \frac{\partial \phi_j}{\partial x} \frac{\partial \eta}{\partial x} &\rightarrow 0 \\ \left( \frac{\partial \eta}{\partial x} \right)^2 &\rightarrow 0 \end{aligned}$$

se linearizan las ecuaciones (2.30) y ahora las ecuaciones que se necesitan resolver son

$$\rho_j \frac{\partial \phi_j}{\partial t} + p_j = 0 \quad (2.31a)$$

$$\frac{\partial^2 \phi_j}{\partial x^2} + \frac{\partial^2 \phi_j}{\partial y^2} = 0, \quad j = 1, 2 \quad (2.31b)$$

$$\frac{\partial \phi_j}{\partial y} = \frac{\partial \eta}{\partial t} \quad \text{para} \quad y = 0 \quad (2.31c)$$

$$P_1 - P_2 = -\frac{\partial^2 \eta}{\partial x^2} \gamma \quad \text{para} \quad y = 0 \quad (2.31d)$$

Se propone como solución de las ecuaciones (2.31) las funciones en variables separables de la siguiente forma

$$\begin{aligned} \phi_j &= \frac{1}{2} \hat{\phi}_j(y) e^{i(kx - \omega t)} + c_1 \\ p_j &= \frac{1}{2} \hat{p}_j(y) e^{i(kx - \omega t)} + c_2 \quad j = 1, 2 \\ \eta &= \frac{1}{2} \hat{\eta} e^{i(kx - \omega t)} + c_3 \end{aligned} \quad (2.32)$$

resolviendo las ecuaciones (2.31) con la solución (2.32),

$$-i\omega\rho_j\hat{\phi}_j + \hat{p}_j = 0 \quad (2.33a)$$

$$\left(\frac{\partial^2}{\partial y^2} - k^2\right)\hat{\phi}_j = 0, \quad j = 1, 2 \quad (2.33b)$$

$$\frac{\partial\hat{\phi}_j}{\partial y} = -i\omega\hat{\eta} \quad \text{para } y = 0 \quad (2.33c)$$

$$(\hat{p}_1 - \rho_1 g\hat{\eta}) - (\hat{p}_2 - \rho_2 g\hat{\eta}) = k^2\gamma\hat{\eta} \quad \text{para } y = 0 \quad (2.33d)$$

la ecuación (2.33d) se obtiene a partir de las ecuaciones (2.2), (2.3) y (2.32).

La solución general de la ecuación (2.33b)

$$\hat{\phi}_j = A_j e^{-ky} + B_j e^{ky} \quad j = 1, 2 \quad (2.34)$$

sin pérdida de generalidad sea  $k > 0$ , con el uso de las ecuaciones de frontera (2.16a) y (2.16b)

$$\hat{\phi}_1 = B_1 e^{ky} \quad \hat{\phi}_2 = A_1 e^{-ky}$$

de las ecuaciones (2.33a), (2.33c) y (2.33d) se obtiene el sistema en forma matricial

$$\begin{pmatrix} 0 & k & i\omega \\ -k & 0 & i\omega \\ i\rho_2\omega & -i\rho_1\omega & (\rho_1 - \rho_2)g + k^2\gamma \end{pmatrix} \begin{pmatrix} A_2 \\ B_1 \\ \hat{\eta} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad (2.35)$$

el sistema tiene solución distinta de la trivial  $A_2 = B_1 = \hat{\eta} = 0$  siempre y cuando su determinante sea igual a cero

$$(\rho_1 + \rho_2)\omega^2 - ((\rho_1 - \rho_2)gk + k^3\gamma) = 0. \quad (2.36)$$

Un parámetro adimensional que involucra las densidades de dos fluidos  $\rho_1$  y  $\rho_2$  es el número de Atwood

$$\text{At} = \frac{\rho_1 - \rho_2}{\rho_1 + \rho_2} \quad (2.37)$$

Se resuelve la ecuación (2.36) para encontrar a  $\omega$

$$\begin{aligned} \omega^2 &= \frac{\rho_1 - \rho_2}{(\rho_1 + \rho_2)}gk + \frac{(\rho_1 - \rho_2)}{(\rho_1 + \rho_2)(\rho_1 - \rho_2)}k^3\gamma \\ &= \text{At}gk + \frac{\text{At}}{(\rho_1 - \rho_2)}k^3\gamma \quad \text{por la ecuación (2.37)} \end{aligned}$$

Si  $\text{At} > 0$  existen dos raíces reales positivas para  $\omega$ , el fenómeno consiste en ondas de superficie neutralmente estables.

Si  $\text{At} < 0$

$$\omega^2 = -(-\text{At})gk + \frac{-\text{At}}{|\rho_1 - \rho_2|}k^3\gamma$$

se da el caso anterior a menos que

$$(-\text{At})g > \frac{-\text{At}}{|\rho_1 - \rho_2|}k^2\gamma$$

$$\frac{g|\rho_1 - \rho_2|}{\gamma} > k^2$$

con el uso de la ecuación (2.1)

$$kl_c < 1, \quad \text{con} \quad l_c = \sqrt{\frac{\gamma}{g|\rho_1 - \rho_2|}} \quad (2.38)$$

para el producto del número de onda  $k$  y la longitud de capilaridad  $l_c$  menores que 1, se tienen dos soluciones para la ecuación (2.36) con valores imaginarios  $\omega_{\pm}$  en la ecuación (2.32)  $e^{-i\omega_{-}t}$  disminuye con el tiempo, pero  $e^{-i\omega_{+}t}$  aumenta con el tiempo. Entonces, en periodos de tiempo largos, la perturbación de la ecuación (2.36) aumenta, esto provoca que la interface se deforme y a este fenómeno se le conoce como la *Inestabilidad de Rayleigh-Taylor*.

El efecto de la viscosidad se manifiesta como números de onda más grandes para los cuales los gradientes de velocidad son más grandes- la viscosidad disipa energía para longitudes de onda pequeñas. Para una tensión superficial nula, las longitudes de onda cortas permanecen inestables con una razón de disminución como  $1/k$  (Figura 2.5). Para una tensión superficial distinta de cero, longitudes de onda cortas son estables y su crecimiento varía como  $-k$ . Un resultado sorprendente es que el número de onda límite  $kl_c = 1$  es independiente de la viscosidad. [1]

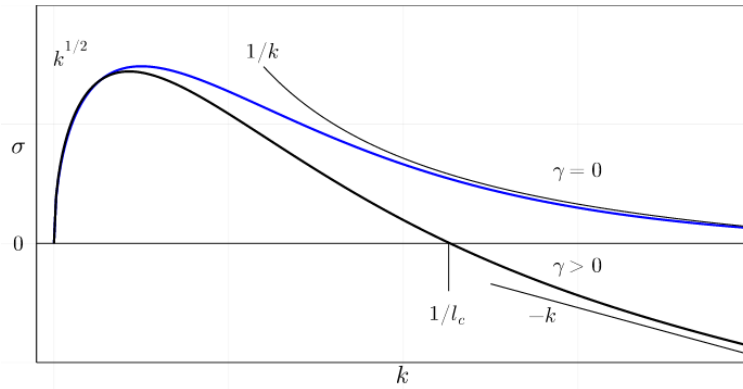


FIGURA 2.5. Razón de crecimiento  $\sigma(k)$  cuando los efectos de la viscosidad no son despreciados, con y sin tensión superficial  $\gamma$ , y comportamiento asintótico para números de onda grandes y pequeños. [1]

En este trabajo se tiene el caso  $\gamma = 0$  representado por la línea azul en la figura 2.5.

## Simulación con Gerris

Gerris es un software de distribución libre que sirve para resolver ecuaciones diferenciales parciales que describen el movimiento de un fluido por medio del método de volumen finito. En particular, resuelve las ecuaciones de Navier-Stokes dependientes del tiempo para un fluido incompresible.

Gerris fue creado por Stéphane Popinet quien es director de investigación en el Centro Nacional de Investigación Científica (CNRS) con sede en el Instituto Jean le Rond de Alembert de la Universidad de la Sorbona, París, Francia [17]. Actualmente Gerris es soportado por este mismo instituto.

### 3.1. Método de Volumen Finito

Randall J. Le Veque [13] menciona que las leyes de conservación surgen naturalmente de leyes físicas que combinan una ecuación diferencial con una integral. Para cualesquiera dos puntos  $x_1$  y  $x_2$

$$\frac{d}{dt} \int_{x_1}^{x_2} q(x, t) dx = f(q(x_2, t)) - f(q(x_1, t)) \quad (3.1)$$

cada componente de  $q$  mide la densidad de alguna cantidad que se conserva y la ecuación (3.1) menciona que el total de esta cantidad entre cualesquiera dos puntos puede cambiar solo debido al flujo que atraviesa ambos puntos. [13]

Los métodos de diferencias finitas resuelven ecuaciones diferenciales aproximando las derivadas por medio de cocientes de diferencias finitas. Estos métodos suelen fallar a menudo donde surgen discontinuidades en la solución de una ecuación diferencial. [13]

El método de volumen finito consiste en resolver ecuaciones diferenciales en su forma integral, como en la ecuación (3.1). Para resolver las ecuaciones se parte el dominio en pequeñas celdas, se aproxima la integral total de  $q$  sobre cada celda, al final se suma el valor total de  $q$  de todas las celdas y así se obtiene la integral sobre el dominio inicial. Esos valores son modificados en cada paso temporal por el flujo que pasa a través de los bordes de la celda. El problema principal es determinar buenas funciones de flujo numéricas de los datos disponibles. [13]

### 3.2. Gerris y la Inestabilidad de Rayleigh-Taylor

En Gerris se encuentra implementado un programa que simula la Inestabilidad de Rayleigh-Taylor [18]. Define el número de cuadros y como están unidos (figura 3.1 a) números rojos), cada uno tiene una unidad de longitud sin dimensiones físicas, en el cuadro número 1 se encuentra el origen del sistema coordenado (figura 3.1 a), entonces el dominio del experimento es  $x \in [-0.5, 0.5]$  y  $y \in [-1.5, 2.5]$ . Además se define una interface  $y(x) = 1/2 - (5/100) \cos(2\pi x)$  que separa a dos fluidos de

densidad  $\rho_1 = 0.1694$  y  $\rho_2 = 1.225$ , donde el fluido más denso está arriba de ella y el menos denso debajo (figura 3.1 b) con  $t = 0$ ). También se necesita la viscosidad dinámica para ambos fluidos  $\mu = 0.00313$  sin unidades físicas.

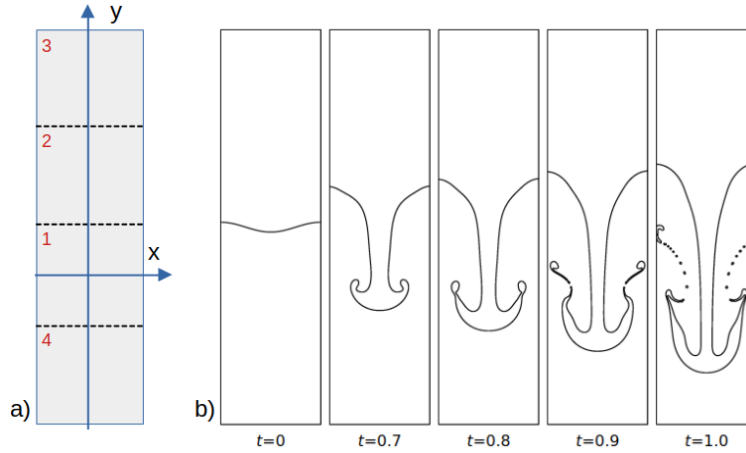


FIGURA 3.1. Imágenes de la Inestabilidad de Rayleigh-Taylor desarrollada con el software Gerris. a) Sistema coordinado de una unidad horizontal por cuatro verticales. b) Desarrollo de la inestabilidad para distintos tiempos. En  $t = 0$  se muestra la interface  $y(x) = 1/2 - (5/100) \cos(2\pi x)$ . [18]

Una vez descritos todos estos datos iniciales se corre el programa en una terminal y se obtienen las imágenes (figura 3.1 b) con  $t = 0.7, 0.8, 0.9, 1.0$ ) y también dos videos que muestran el movimiento de los fluidos (figura 3.2 a) y la vorticidad (figura 3.2 b).

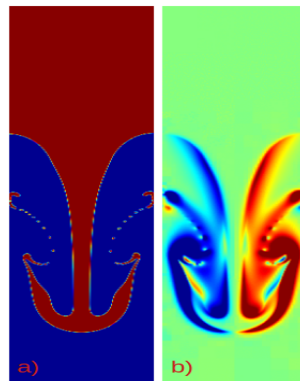


FIGURA 3.2. Inestabilidad de Rayleigh-Taylor cuando  $t = 1.0$ . a) Densidades  $\rho_1$  en azul y  $\rho_2$  en rojo. b) vorticidad. [18]

A este programa se le agregan cuadros en la dirección horizontal para aumentar el dominio en la coordenada  $x$ , al incluir una columna con cuatro cuadros se aumenta

en una unidad el ancho del experimento, también se cambia la ecuación que describe la interface entre los dos fluidos, de ser una función  $y(x) = 1/2 - (5/100) \cos(2\pi x)$  se cambia a una función  $y(x) = 0.5$ , este último cambio se necesita para evitar la formación de perturbaciones simétricas y poco realistas físicamente.

En la figura 3.3 se ejecuta el programa con los nuevos cambios. A medida que transcurre el tiempo se observa la generación de caminos de fluido que se intercalan, caminos de fluido más denso desplazándose hacia abajo intercalados con caminos de fluido menos denso que asciende.

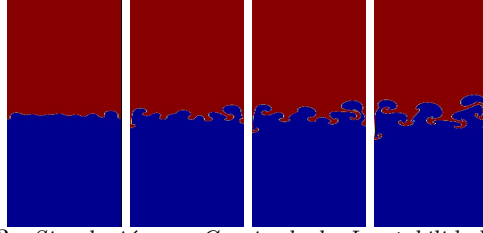


FIGURA 3.3. Simulación en Gerris de la Inestabilidad de Rayleigh-Taylor. Sistema coordinado de una unidad horizontal por cuatro unidades verticales. Viscosidad dinámica  $\mu = 0.00313$  para ambos fluidos. En este problema la viscosidad es adimensional y proporcional al inverso del número de Reynolds. [19]

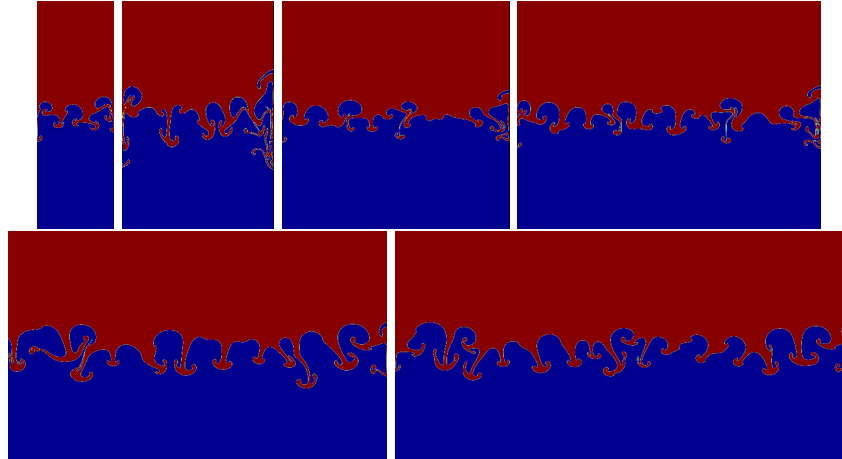


FIGURA 3.4. Simulación en Gerris de la Inestabilidad de Rayleigh-Taylor. Sistema coordinado de una unidad hasta seis unidades horizontales por cuatro verticales. Imágenes de izquierda a derecha de arriba a abajo. Viscosidad dinámica  $\mu = 0.00313$  para ambos fluidos.

En la figura 3.4 se aprecia el incremento de los caminos de fluido cuando se aumenta el ancho en la interacción entre ambos fluidos.

Se corre la simulación variando la longitud de la interacción de los fluidos de una unidad hasta quince unidades. Se mide el número de líneas  $N$  con respecto al

incremento de las unidades de ancho  $W$  en el experimento, como se aprecia en la figura 3.5.

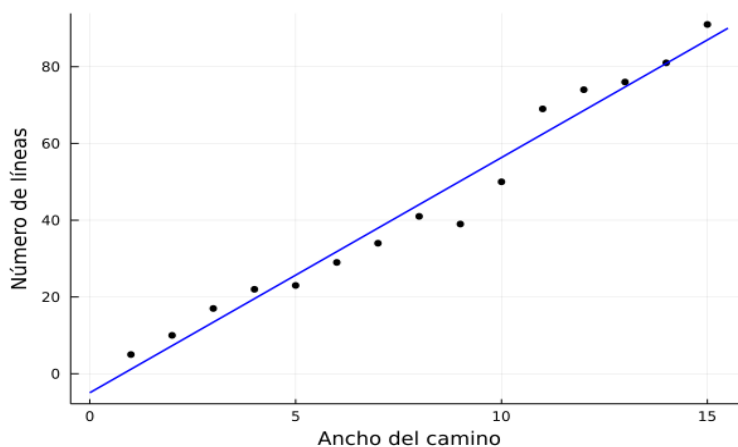


FIGURA 3.5. Número de líneas  $N$  con respecto al ancho  $W$  representado por los puntos negros. El ajuste lineal calculado por el método de mínimos cuadrados está dado como  $N(W) = (6.12857)W - 4.9619$  y se representa con la recta azul.

En este caso se realiza un experimento numérico para cada valor de  $W$  y así se determina el número de líneas  $N$ . El número de líneas  $N$  crece como una función lineal del ancho  $W$  y es representado por la ecuación lineal  $N(W) = (6.12857)W - 4.9619$  que se obtiene por un análisis de regresión lineal.

### 3.3. Gerris y el Oscilador de Densidad

El programa de Gerris [18] se adapta para simular el sistema: primero se colocan fronteras como las descritas por la figura 1.1, se usan las funciones que existen en el programa para implementar las fronteras o bien se pueden importar de otros softwares; segundo, se agrega otro fluido mucho menos denso que representa al aire; tercero, el programa produce un archivo de texto csv, el cual es analizado con el lenguaje de programación julia, para de esta manera producir un archivo gif y cualquier gráfica que sea necesaria.

Gerris divide cada unidad de su dominio en múltiplos de 2, para todos los experimentos mostrados más adelante se divide la unidad del dominio en  $2^7$  intervalos, por lo tanto la unidad más pequeña tiene una longitud de  $l = (1/2)^7$ . Las variables físicas representadas en el programa tienen unidades que simplifican los cálculos.

Se realiza una simulación con un orificio para el oscilador de densidad de  $a = 10l = 0.078125$ , una viscosidad dinámica de  $\mu = 0.00113$ , las densidades del fluido más denso al menos denso  $\rho_1 = 1.15\rho_2$ ,  $\rho_2 = 1.0$  y  $\rho_3 = 0.006\rho_2$ , donde se está utilizando la densidad  $\rho_2$  para adimensionalizar todas las densidades. Como resultado se obtienen las imágenes de la figura 3.6.

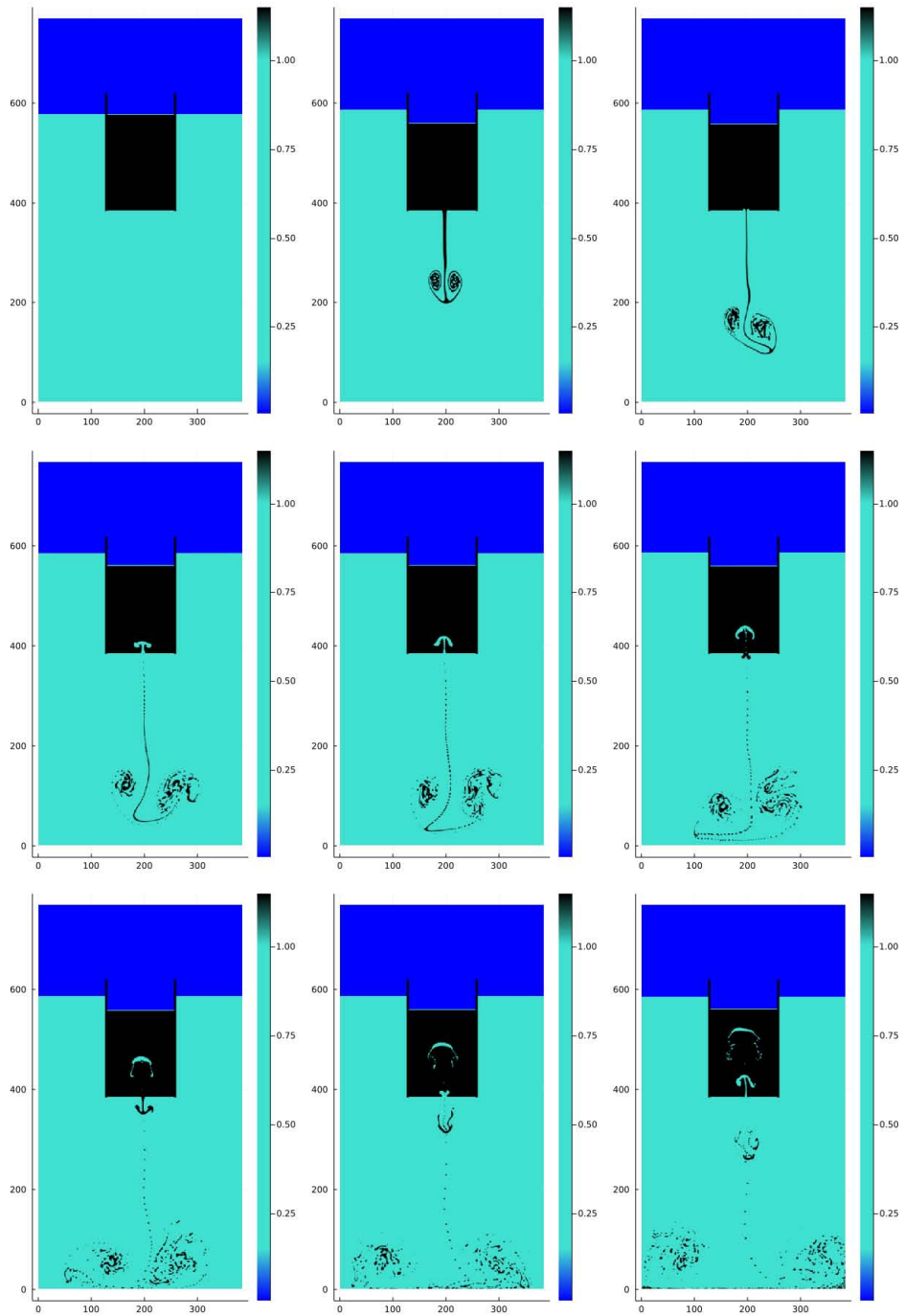


FIGURA 3.6. Oscilador de densidad con un orificio  $a = 10l = 0.078125$ ,  $\mu = 0.00113$ . Las imágenes muestran los lapsos temporales  $t = 0, 2, 3.2, 4.4, 4.8, 5.6, 6.6, 7.6$  y  $8.8$ .

El sistema coordinado del experimento es  $-1.5 < x < 1.5$  y  $-3.5 < y < 2.5$ , donde el centro del orificio  $a$  se encuentra en el punto  $(0, -0.5)$ . Este dominio se divide en intervalos de longitud  $l$  para ambas coordenadas y la nueva escala muestra el número de unidades  $l$ . La gráfica 3.6 muestra el valor de la densidad por medio de colores donde el azul representa la densidad más baja y el negro la más alta.

Para obtener la frecuencia del movimiento oscilatorio se determina la diferencia de presión  $\Delta P$  entre un punto dentro y otro fuera del recipiente pequeño que se encuentren a la misma altura. Sean  $(0.484375, -0.484375)$  y  $(0.515625, -0.484375)$  tales puntos. Como resultado se encuentra para el movimiento oscilatorio una frecuencia adimensional de  $f = 0.2639$  y de tal frecuencia se tiene el periodo del movimiento como  $T = 1/f = 3.79$ . Por lo tanto el ciclo se repite cada 3.79 unidades.

El ciclo del movimiento oscilatorio se representa en la figura 3.7.

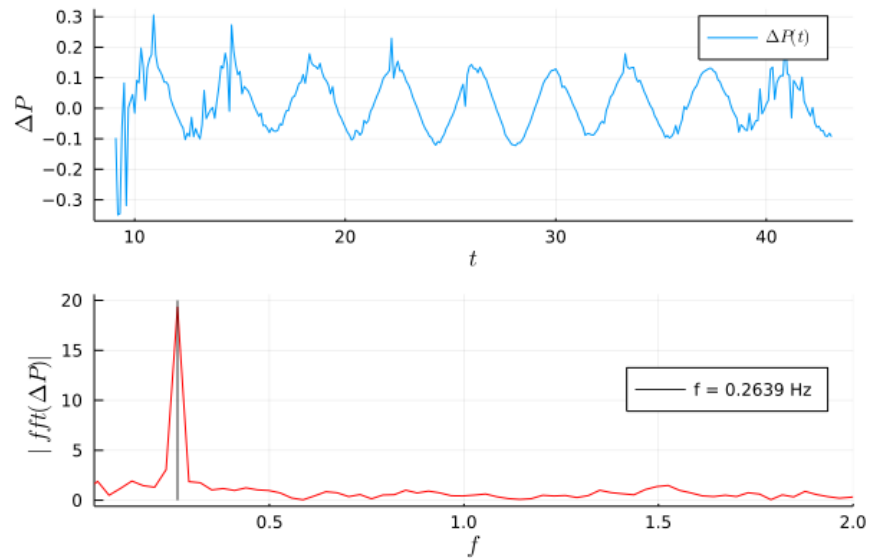


FIGURA 3.7. Diferencia de presión  $\Delta P$ , como función del tiempo entre los puntos  $(0.484375, -0.484375)$  y  $(0.515625, -0.484375)$  para el oscilador de densidad de la figura 3.6. El centro del orificio  $a$  se encuentra en  $(0, -0.5)$ .  $f = 0.2639$  es la frecuencia del movimiento oscilatorio.

## Lenguaje de programación Julia y el oscilador peatonal

En el presente capítulo se desarrolla el modelo de Dirk Helbing y Peter Molnár mostrado en la sección 1.3 (pág. 7) con el uso del lenguaje de programación julia[16].

### 4.1. Peatones cruzando una calle

Se desarrolla un programa en el que se simula a un grupo de peatones que cruzan una calle de largo `c_a` y ancho `c_b`, la mitad de ellos desea moverse hacia la izquierda y la otra mitad hacia la derecha. Para simplificar el problema los peatones tienen una velocidad de magnitud constante `v_0` a la que se desean mover, también se implementa la solución a las ecuaciones diferenciales (1.9) y (1.11) con el uso del método de Euler.

En el anexo C se muestra el programa `MFS calle.jl` que implementa las ecuaciones (1.3), (1.4), (1.5), (1.7), (1.8), (1.9), (1.11) y (1.10). Además se utiliza el análisis sobre las ecuaciones descrita en el apéndice A. En el programa `Ejec-calle-h.jl` se ejecuta el movimiento de los peatones implementando las variables necesarias para el uso de la ecuación (1.12) descrita por Helbing.

Se utilizan los valores de Helbing y Molnár descritos en el cuadro 4.1

CUADRO 4.1. *Valores necesarios en la implementación de las ecuaciones (1.12).*

$V_{\alpha\beta}^0$	$2.1\text{m}^2/\text{s}^2$	ancho ( <code>c_b</code> )	10m
$\sigma$	0.3m	largo ( <code>c_a</code> )	50m
$U_{\alpha\beta}^0$	$10\text{m}^2/\text{s}^2$	$\Delta t$	$(1.0/2.0)^4 \text{ s}$
$R$	0.2m	$\tau_\alpha$	0.5 s

con excepción del tiempo involucrado  $\Delta t$  en el paso que da el peatón  $\beta$ , como se observa en la ecuación (1.6). Como resultado se obtiene la figura 4.1, donde el movimiento comienza con una posición aleatoria para todos los peatones y una densidad de 0.3 peatones por metro cuadrado. Como se observa, para tiempos largos se ve claramente la formación de líneas, independientemente de la posición inicial.

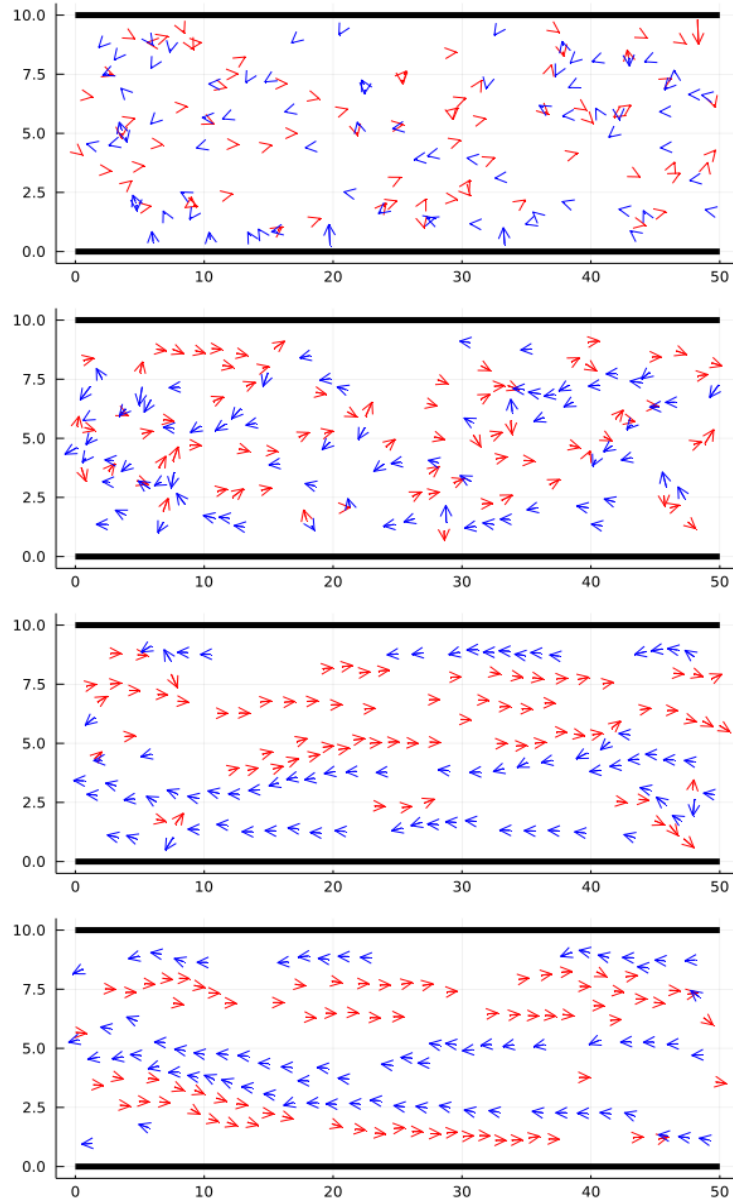


FIGURA 4.1. Posiciones de los peatones para diferentes intervalos temporales de arriba al fondo,  $t = 0$  s,  $t = 56(1/2)^4$  s,  $t = 3230(1/2)^4$  s y  $t = 4000(1/2)^4$  s. Con los parámetros de la tabla 4.1. Los peatones rojos se mueven a la derecha y los azules a la izquierda. El programa tiene condiciones periódicas a la frontera, en los extremos de la calle.

Ahora se cambia el ancho  $c.b$  de la calle, manteniendo la densidad de peatones constante. El resultado se muestra en la figura 4.2 donde se observa cómo a medida que aumenta el ancho de la calle también lo hace el número de filas de peatones.

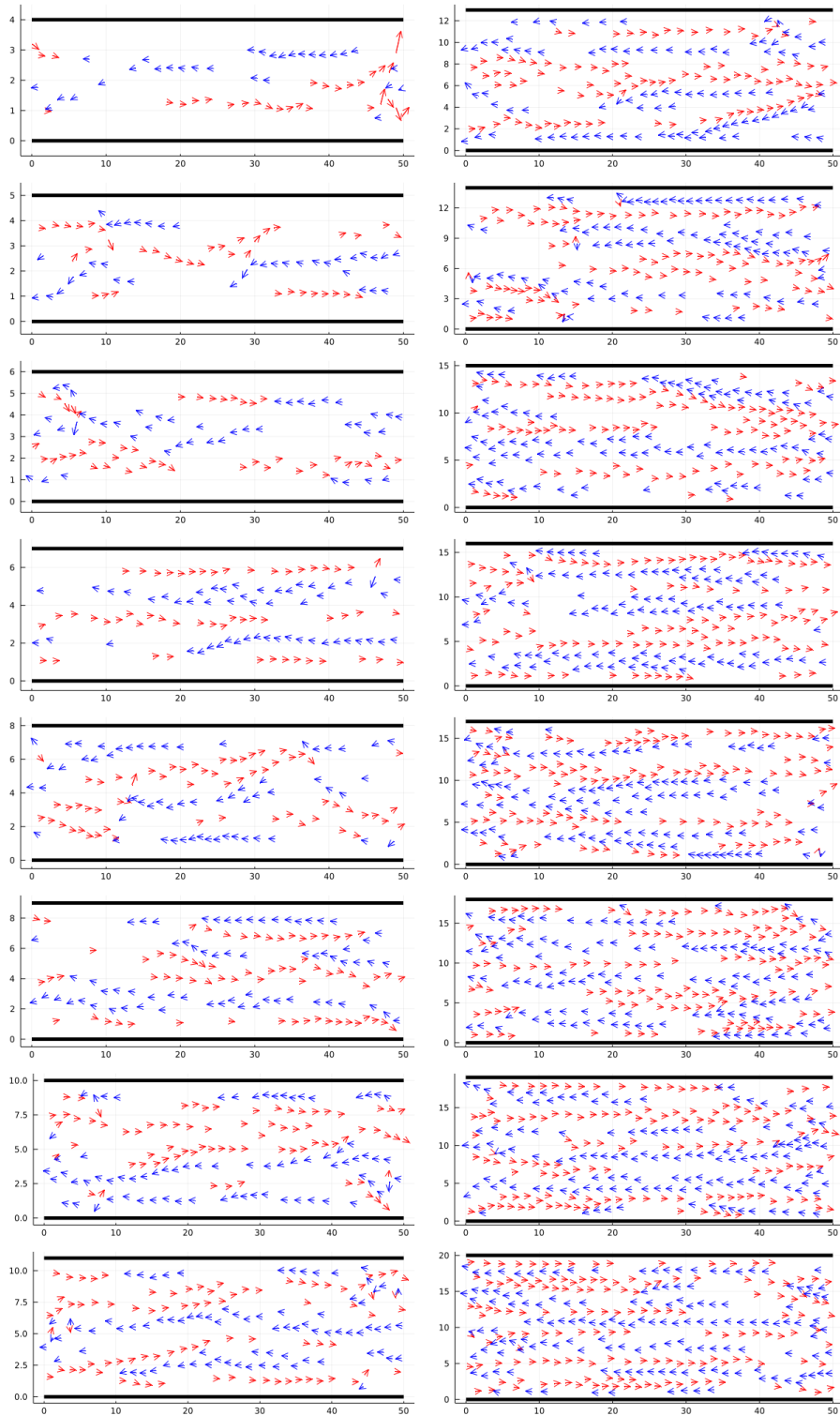


FIGURA 4.2. Posiciones de los peatones obtenidas con los parámetros del cuadro 4.1. El ancho de la calle cambia desde  $c_b = 4$  hasta  $c_b = 20$  metros. Los peatones rojos se mueven a la derecha y los azules a la izquierda. Todos los programas tienen condiciones periódicas a la frontera en la dirección horizontal.

Para determinar tal relación se grafica el promedio en el número de filas con respecto al ancho de la calle y se obtiene como resultado la figura 4.3. El promedio se calcula ejecutando 10 veces un programa para cada valor  $c_b$  que corresponde al ancho del camino.

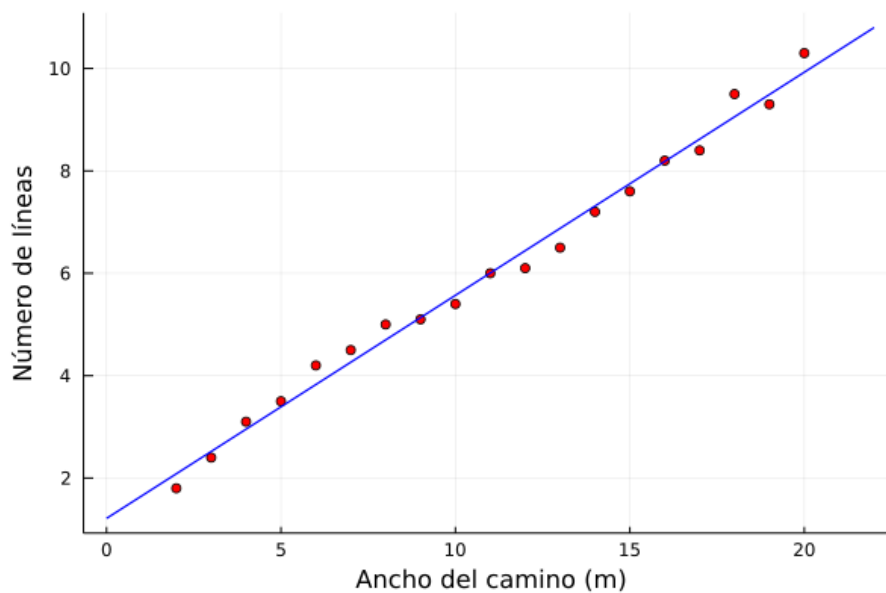


FIGURA 4.3. Número de líneas  $N$  con respecto al ancho de la calle  $W$  representado por los puntos rojos. El ajuste lineal está dado como  $N(W) = (0.435789 \text{ m}^{-1})W + 1.21158$  y se representa con la recta azul. Cada punto corresponde al promedio de 10 programas ejecutados.

Por lo mostrado en la figura 4.3 se aprecia una dependencia lineal del número de filas  $N$  con respecto al ancho de la calle  $W$ , con la relación  $N(W) = (0.435789 \text{ m}^{-1})W + 1.21158$  obtenida por medio de un análisis de regresión lineal.

Por último se realizó el experimento dejando a los peatones azules fijos mientras los rojos se mueven hacia la derecha, como se ve en la figura 4.4. Se forman líneas de peatones rojos y éstos tienden a alinear a los peatones azules que simplemente se mueven para permitir el flujo de los peatones rojos.

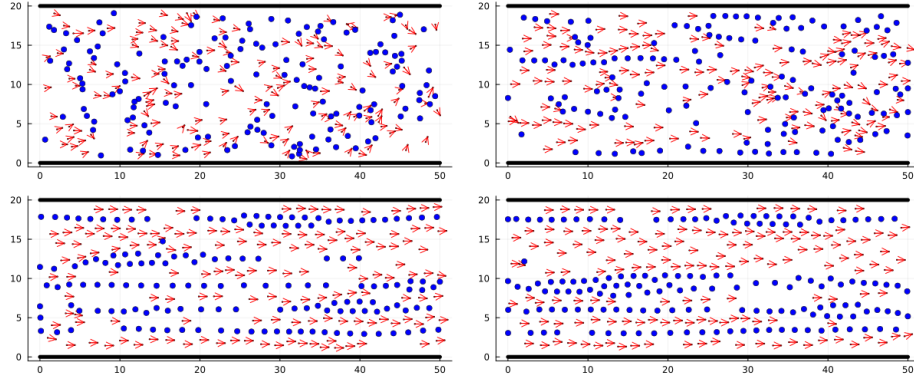


FIGURA 4.4. Posiciones de los peatones para diferentes intervalos temporales de arriba a abajo, izquierda a derecha,  $t = 10(1/2)^4 s$ ,  $t = 500(1/2)^4 s$ ,  $t = 2000(1/2)^4 s$  y  $t = 4000(1/2)^4 s$ . El ancho de la calle corresponde a  $c\_b = 20.0$  y los parámetros de la tabla 4.1. Los peatones rojos se mueven a la derecha y los azules tienen velocidad deseada cero. Todos los programas tienen condiciones periódicas a la frontera.

En general se ha trabajado con el movimiento de los peatones con direcciones opuestas, un caso interesante se muestra en el apéndice E donde dos grupos de peatones se mueven en dirección ortogonal uno con respecto al otro.

#### 4.2. Peatones cruzando una puerta(oscilador peatonal)

En el apéndice D se desarrolla el programa `MFSpuerta.jl` que implementa las funciones necesarias para simular el movimiento de peatones tratando de cruzar una puerta en ambos sentidos, siguiendo el esquema de la sección 1.3 7. En el programa `Ejec-puerta.jl` se ejecuta la simulación de los peatones y para ello se utilizan las variables mostradas en el cuadro 4.2

CUADRO 4.2. Valores necesarios en la implementación de las ecuaciones (1.12).

$V_{\alpha\beta}^0$	$2.1\text{m}^2/\text{s}^2$	ancho (c_b)	20m
$\sigma$	0.3m	largo (c_a)	20m
$U_{\alpha\beta}^0$	$10.0\text{m}^2/\text{s}^2$	$\Delta t$	$(1.0/2.0)^4 s$
$R$	0.2m	$\tau_\alpha$	0.5 s
$v_\alpha^0$	1.0m/s	puerta ( $L_p$ )	0.5 m

La simulación consiste en lo siguiente: una calle con  $c\_a = 20$  metros de largo y  $c\_b = 20$  de ancho, a la mitad de este cuadrado se coloca una puerta de  $L_p = 0.5$  m, a continuación se colocan de manera aleatoria 125 peatones del lado izquierdo de la puerta y otros 125 peatones del lado derecho, los peatones desean moverse hacia un punto designado al otro lado de la calle cruzando la puerta, pero para ello deben atravesar la puerta.

A partir de la ejecución del programa `Ejec-puerta.jl` y los datos de la tabla 4.2 se obtienen las imágenes de la figura 4.5 donde se observa para un tiempo  $t = 235(1/2)^4$  s el momento en que los peatones naranjas cruzan la puerta mientras los azules esperan, después para un tiempo  $t = 485(1/2)^4$  s el proceso se invierte y esta vez son los peatones azules los que cruzan la puerta y los otros esperan.

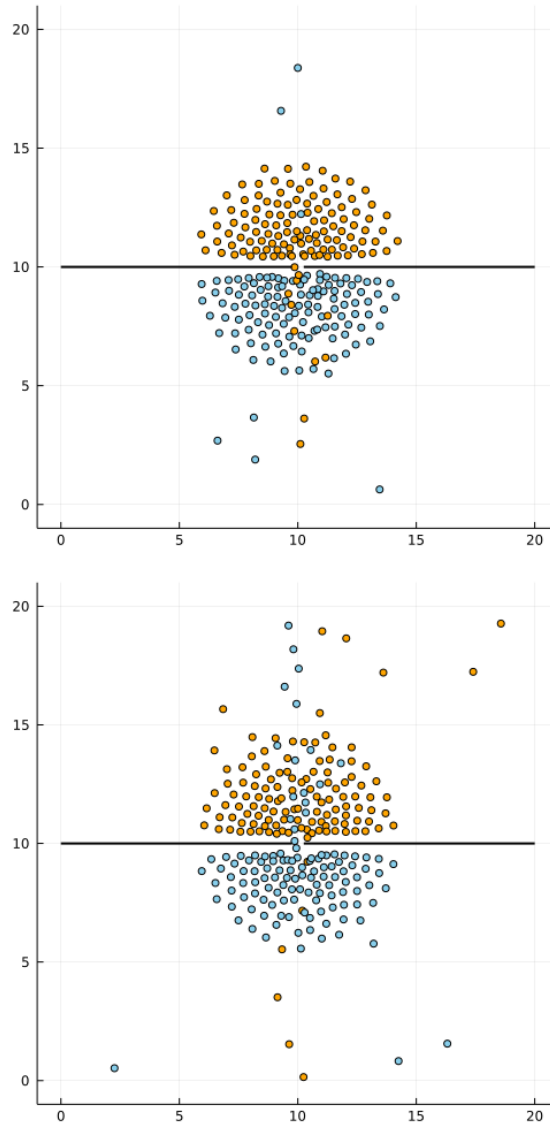


FIGURA 4.5. Posiciones de los peatones para diferentes intervalos temporales,  $t = 235(1/2)^4$  s y  $t = 485(1/2)^4$  s. Las variables corresponden a los datos de la tabla 4.2. Los peatones azules se mueven hacia arriba y los naranjas hacia abajo.

Para determinar si el movimiento es periódico, se tiene calcula el flujo de peatones por unidad de tiempo y después se calcula la función de autocorrelación asociada a este flujo para poder determinar la frecuencia del movimiento. El flujo se define como la cantidad de peatones que cruzan a la derecha menos los que cruzan a la izquierda en un cierto intervalo de tiempo. La función de autocorrelación sin normalizar [12] asociada a un fenómeno discreto es

$$Acorr(\tau) = \sum_{i=1}^{N-\tau_{\max}} fl(i + \tau) fl(i) \quad (4.1)$$

donde  $fl$  representa el flujo e  $i$  es un índice con el que se mide cuanto tiempo ha transcurrido.

Para este experimento se calcula el flujo neto utilizando un intervalo de tiempo de  $\Delta t = 10(1/2)^4$  s,  $fl$  es entonces el flujo neto como función del tiempo transcurrido,  $i$  veces  $\Delta t$ . Después se determina la función de autocorrelación del flujo  $fl$  como función de  $\tau$ . Cuando  $\tau$  es igual al periodo o a un múltiplo del periodo se obtiene un máximo de la función de autocorrelación. Por otro lado, cuando  $\tau$  es la mitad del periodo o la mitad del periodo más un múltiplo del periodo se obtiene un mínimo, revelándose el carácter periódico de la dinámica. Por último se obtiene el valor absoluto de la transformada rápida de Fourier de la autocorrelación para obtener la frecuencia de la oscilación.

El fenómeno ha sido estudiado una duración temporal de  $t = 30000 h$ , donde  $h = (1/2)^4$  s es el paso de integración para la fórmula de Euler, el intervalo se divide en subintervalos de diez unidades, por lo que el flujo neto presenta 3000 elementos y para este caso la autocorrelación tiene 1000 elementos. En la figura 4.6 se muestra la función del flujo, la correlación del flujo y la frecuencia asociada a la autocorrelación.

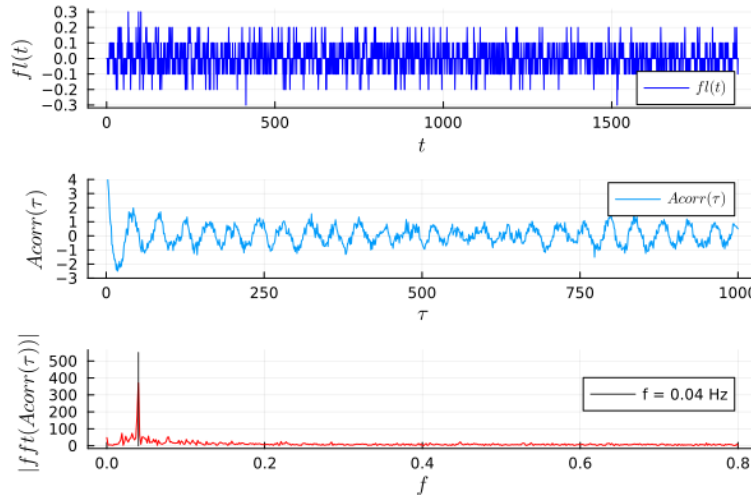


FIGURA 4.6. Flujo, autocorrelación y frecuencia asociada al movimiento periódico de los peatones que atraviesan la puerta. Las variables corresponden a los datos de la tabla 4.2. Donde  $1 < i < 3000$ ,  $1 < \tau < 1000$ . La frecuencia del movimiento es  $f = 0.04$  Hz.

Se aprecia una frecuencia de  $f = 0.04$  Hz en la correlación por lo que el periodo es  $1/f = 25$  s, es decir, el fenómeno se repite aproximadamente cada 25 segundos. Por lo tanto se concluye que el movimiento de peatones tratando de cruzar una puerta se comporta como un fenómeno oscilatorio. A este fenómeno lo llamamos “*Oscilador Peatonal*”.

## Comparación entre los osciladores de densidad y peatonal

El oscilador peatonal es un fenómeno discreto estocástico y el oscilador de densidad es un fenómeno de un medio continuo en la dinámica de fluidos, por tal motivo para compararlos se necesita una manera especial de abordarlos, para ello se realiza un análisis entre las principales variables que describen ambos fenómenos en la siguiente sección.

### 5.1. El parámetro adimensional $\alpha$

Esta sección esta basada en una idea que desarrolló Florence Raynal.

De acuerdo con la sección 1.3 (7) tres parámetros importantes para la descripción del fenómeno son:

- La velocidad deseada  $\mathbf{v}_\alpha^0(t)$  de cada peatón  $\alpha$ .
- Un valor de agresividad  $\tau$  que define el tiempo en el cual los peatones alcanzan la velocidad deseada a partir de una velocidad aleatoria inicial.
- El tamaño de la puerta  $R$ .

Con estos tres parámetros se puede construir una variable adimensional  $\alpha$  de la siguiente manera:

$$\alpha = \frac{\mathbf{v}_\alpha^0 \tau}{R} \quad (5.1)$$

Ahora lo importante para hacer la comparación entre el oscilador de densidad y el oscilador peatonal es definir los parámetros anteriores en términos de valores del oscilador de densidad 1.

Para simplificar el problema del oscilador de densidad se considera una partícula de fluido esférica con velocidad  $\mathbf{v}$ , masa  $m$  y radio  $R$ , tiene que satisfacer la ecuación diferencial

$$m \frac{d\mathbf{v}}{dt} = \mathbf{F}_{\text{Arq}} + \mathbf{F}_{\text{Sto}} \quad (5.2)$$

donde  $\mathbf{F}_{\text{Arq}} = (4/3\pi R^3 \Delta\rho \mathbf{g})$  y  $\mathbf{F}_{\text{Sto}} = -6\pi\mu R \mathbf{v}$ . La fuerza de Arquímedes  $\mathbf{F}_{\text{Arq}}$  es la diferencia entre la fuerza de flotación y el peso de la partícula de fluido con  $\Delta\rho$  la diferencia entre las densidades en la interface y  $\mathbf{g}$  el vector que representa la gravedad terrestre. La fuerza de Stokes  $\mathbf{F}_{\text{Sto}}$  es la fuerza de resistencia que ejerce un fluido viscoso sobre un cuerpo esférico que se mueve a través de este, donde  $\mu$  representa la viscosidad dinámica del fluido.

La ecuación diferencial (5.2) se transforma en

$$\begin{aligned}\frac{4}{3}\pi R^3 \rho \frac{d\mathbf{v}}{dt} &= \frac{4}{3}\pi R^3 \Delta\rho \mathbf{g} - 6\pi\mu R \mathbf{v} \\ \frac{d\mathbf{v}}{dt} + \frac{6\pi\mu R}{\frac{4}{3}\pi R^3 \rho} \mathbf{v} &= \frac{\frac{4}{3}\pi R^3 \Delta\rho}{\frac{4}{3}\pi R^3 \rho} \mathbf{g} \quad \text{como } \nu = \mu/\rho \\ \frac{d\mathbf{v}}{dt} + \frac{9\nu}{2R^2} \mathbf{v} &= \frac{\Delta\rho}{\rho} \mathbf{g} \quad \text{si } \tau = \frac{2R^2}{9\nu} \\ \frac{d\mathbf{v}}{dt} + \frac{1}{\tau} \mathbf{v} &= \frac{\Delta\rho}{\rho} \mathbf{g}\end{aligned}$$

La ecuación vectorial anterior representa tres ecuaciones escalares. Sin pérdida de generalidad sea la velocidad inicial perpendicular a la gravedad, entonces  $\mathbf{v}_0 \perp \mathbf{g}$

$$\frac{dv_x}{dt} + \frac{1}{\tau}v_x = 0 \quad \frac{dv_y}{dt} + \frac{1}{\tau}v_y = 0 \quad \frac{dv_z}{dt} + \frac{1}{\tau}v_z = -\frac{\Delta\rho}{\rho}g$$

La solución de la primera ecuación es

$$\begin{aligned}\frac{dv_x}{dt} + \frac{1}{\tau}v_x &= 0 \\ \frac{dv_x}{dt} &= -\frac{1}{\tau}v_x \\ \int \frac{dv_x}{v_x} &= -\frac{1}{\tau} \int dt \\ \ln(v_x) &= -\frac{t}{\tau} + K \\ v_x &= K_x e^{-\frac{t}{\tau}} \quad \text{como } v_x(t=0) = v_{0x} \\ v_x &= v_{0x} e^{-\frac{t}{\tau}}\end{aligned}$$

de igual manera para la componente  $y$ ,  $v_y = v_{0y}e^{-\frac{t}{\tau}}$ . Por último para la componente  $z$

$$\begin{aligned}\frac{dv_z}{dt} + \frac{1}{\tau}v_z &= -\frac{\Delta\rho}{\rho}g \\ \frac{dv_z}{dt} &= -\frac{1}{\tau} \left( v_z + \frac{\Delta\rho}{\rho}g\tau \right) \\ \int \frac{dv_z}{v_z + \frac{\Delta\rho}{\rho}g\tau} &= -\frac{1}{\tau} \int dt \\ \ln \left( v_z + \frac{\Delta\rho}{\rho}g\tau \right) &= -\frac{t}{\tau} + K \\ v_z &= K_z e^{-\frac{t}{\tau}} - \frac{\Delta\rho}{\rho}g\tau \quad \text{como } v_z(t=0) = 0 \\ v_z &= \frac{\Delta\rho}{\rho}(-g)\tau \left( 1 - e^{-\frac{t}{\tau}} \right)\end{aligned}$$

Por lo tanto, la solución de la ecuación (5.2) es

$$\mathbf{v} = \mathbf{v}_0 e^{-\frac{t}{\tau}} + \frac{\Delta\rho}{\rho} \mathbf{g} \tau \left(1 - e^{-\frac{t}{\tau}}\right) \quad (5.3)$$

con  $\mathbf{v}_0 \perp \mathbf{g}$  y  $\tau = 2R^2/9\nu$ , con  $\nu = \mu/\rho$  la viscosidad cinemática. La velocidad deseada en el oscilador peatonal  $\mathbf{v}_\alpha^0(t)$  se puede comparar con la velocidad terminal en el oscilador de densidad y ésta se alcanza cuando el tiempo es muy grande  $t \rightarrow \infty$ , entonces

$$\begin{aligned} \mathbf{v}_\alpha^0 &= \lim_{t \rightarrow \infty} \mathbf{v} = \lim_{t \rightarrow \infty} \left[ \mathbf{v}_0 e^{-\frac{t}{\tau}} + \frac{\Delta\rho}{\rho} \mathbf{g} \tau \left(1 - e^{-\frac{t}{\tau}}\right) \right] \\ &= \frac{\Delta\rho}{\rho} \mathbf{g} \tau = \frac{\Delta\rho}{\rho} \mathbf{g} \frac{2R^2}{9\nu} = \mathbf{g} \Delta\rho \frac{2R^2}{9\mu} \end{aligned} \quad (5.4)$$

Como las escalas de tiempo y velocidades son muy distintas entre el oscilador de densidad y el oscilador peatonal se necesita un parámetro adimensional para realizar la comparación. La ecuación 5.1 que representa a un oscilador peatonal, con el uso de las ecuaciones (5.3) y (5.4) se transforma en

$$\alpha = \frac{\mathbf{v}_\alpha^0 \tau}{R} = \Delta\rho \frac{2R^2}{9\mu} \mathbf{g} \frac{2R^2}{9\nu} \frac{1}{R} = \frac{4}{81} \frac{\Delta\rho}{\rho} \frac{R^3}{\nu^2} \mathbf{g} \quad (5.5)$$

que que es un parámetro adimensional asociado al oscilador de densidad.

## 5.2. El parámetro adimensional $\alpha$ para ambos osciladores

Un valor muy usado en dinámica de fluidos es el número de Grashof que se define como:

$$\text{Gr} = \frac{\Delta\rho}{\rho} \frac{R^3}{\nu^2} \mathbf{g} = \rho \Delta\rho \frac{R^3}{\mu^2} \mathbf{g} \quad (5.6)$$

De la ecuación (5.5) y (5.6)

$$\alpha = \frac{4}{81} \text{Gr} = \frac{4}{81} \rho \Delta\rho \frac{R^3}{\mu^2} \mathbf{g} \quad (5.7)$$

El primer paso es establecer las variables necesarias para determinar el parámetro  $\alpha$  en ambos osciladores. En el cuadro 5.1 se tienen las variables características de ambos osciladores

CUADRO 5.1. Variables involucradas en la comparación de ambos osciladores.

Oscilador Peatonal	Oscilador de Densidad
Longitud de la puerta, $L$	Radio del orificio, $R$
Tiempo de relajación del peatón, $\tau$	Tiempo de relajación de la esfera de fluido, $\frac{2\rho R^2}{9\mu}$
Velocidad deseada, $\mathbf{v}_\alpha^0$	Velocidad terminal de la esfera de fluido, $\mathbf{g} \Delta\rho \frac{2R^2}{9\mu}$
$\alpha = \frac{\mathbf{v}_\alpha^0 \tau}{L}$	$\alpha = \frac{4}{81} \text{Gr}$

donde la magnitud de la gravedad es  $g$ ,  $\rho$  es la densidad del fluido más denso,  $\Delta\rho$  la diferencia de densidad entre ambos fluidos y  $\mu$  la viscosidad dinámica entre los fluidos.

### 5.3. Comparación

Después de correr varias veces los programas de los apéndices B y D con los valores más adecuados para describir tanto al oscilador peatonal como al oscilador de densidad, se encontraron los valores del cuadro 5.2.

CUADRO 5.2. *Valores elegidos para obtener valores de  $\alpha$  similares para los dos osciladores.*

Oscilador Peatonal	Oscilador de Densidad
$L = 0.5 \text{ m}$	$R = (7/2)(1/2)^7$
$\tau = 0.5 \text{ s}$	$\rho = 1.15$
$v_\alpha^0 = 1.34 \text{ m/s}$	$\Delta\rho = (1.15 - 1.0)$
	$\mu = 0.00113$
	$g = 9.81$
$\alpha = 1.34$	$\alpha = 1.34$

Una vez que se tienen simulaciones con el mismo parámetro adimensional  $\alpha$  se procede a hacer el análisis de cada una de ellas y realizar su comparación.

Para el oscilador de densidad se obtienen las imágenes de la figura 5.1.

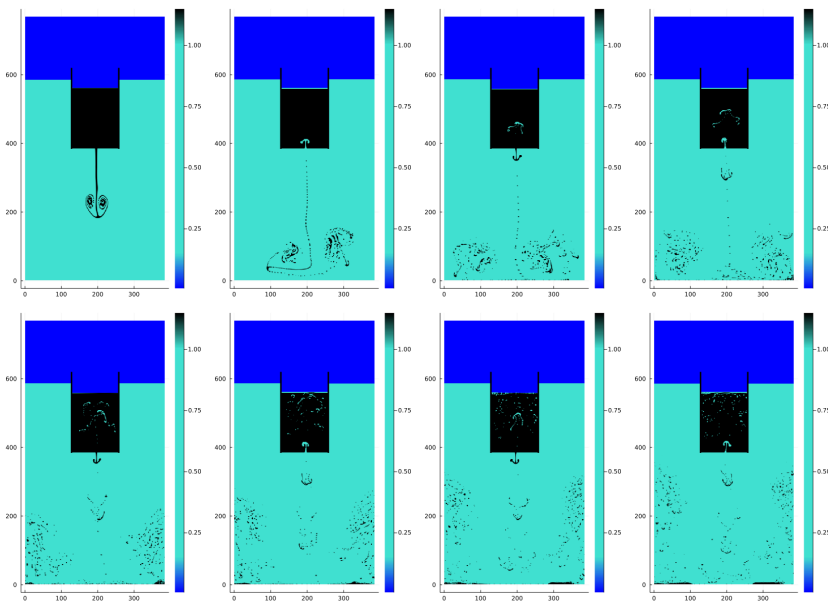


FIGURA 5.1. *Oscilador de densidad con un orificio  $R = (7/2)(1/2)^7$ ,  $\mu = 0.00113$ . Las imágenes muestran los lapsos temporales  $t = 2.3, 5.9, 8.1, 9.8, 12.2, 14.1, 16.4$ , y  $18.4$ .*

A partir de la medición de la diferencia de presiones  $\Delta P$  entre los puntos  $(0.484375, -0.484375)$  y  $(0.515625, -0.484375)$  se obtiene la imagen 5.2, donde se tiene una frecuencia de  $f_d = 0.2564$  y un periodo de  $T_d = 1/f_d = 3.9$ .

$$f_d = 0.2564, \quad T_d = 1/f_d = 3.9 \quad (5.8)$$

con el subíndice  $d$  representando al oscilador de densidad.

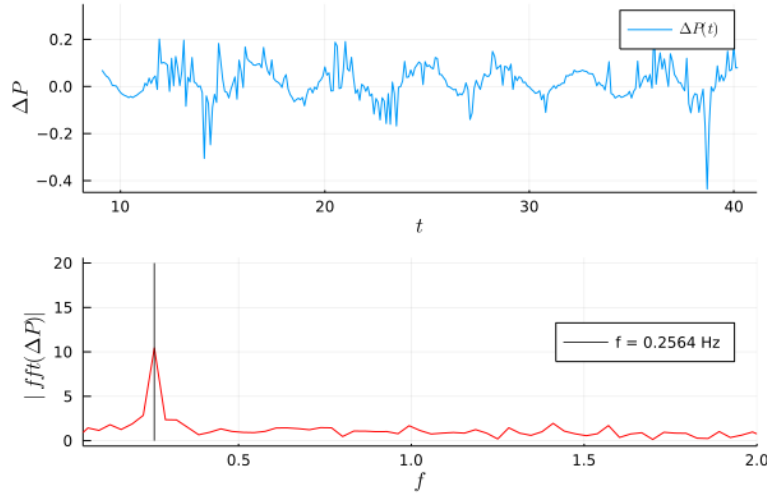


FIGURA 5.2. Diferencia de presión  $\Delta P$ , como función del tiempo entre los puntos  $(0.484375, -0.484375)$  y  $(0.515625, -0.484375)$  para el oscilador de densidad de la figura 5.1. El centro del orificio  $R$  se encuentra en  $(0, -0.5)$ .  $f = 0.2564$  es la frecuencia del movimiento oscilatorio.

Como las escalas de tiempo para el oscilador peatonal y para el oscilador de densidad son muy distintas, una comparación entre ambos fenómenos necesita de un parámetro adimensional adicional, en este caso para medir la frecuencia y el periodo de los experimentos se tomará el cociente entre el periodo  $T_p$  y el tiempo de relajación  $T_p/\tau$  para el oscilador peatonal y el cociente entre el periodo y la correspondiente expresión de  $\tau$  para el oscilador de densidad  $T_d/\left(\frac{2\rho R^2}{9\mu}\right)$ . Entonces para el oscilador de densidad se tiene una frecuencia adimensional  $f_{da}$  y un periodo  $T_{da}$  dado por

$$f_{da} = 0.0434, \quad T_{da} = 1/f_{da} = 23.07. \quad (5.9)$$

Por otro lado para el oscilador peatonal con los datos del cuadro 5.2, se obtienen las imágenes de la figura 5.3.

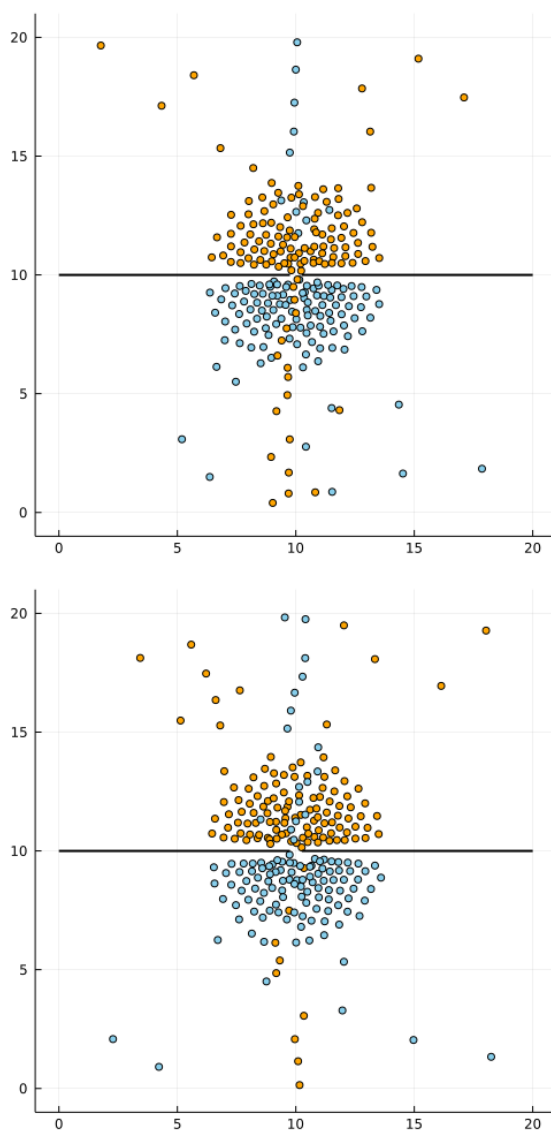


FIGURA 5.3. Posiciones de los peatones para diferentes intervalos temporales,  $t = 20709(1/2)^4$  s y  $t = 20807(1/2)^4$  s. Las variables corresponden a los datos del cuadro 4.2. Los peatones azules se mueven hacia arriba y los naranjas hacia abajo.

Se hace una comparación con la diferencia de presiones del oscilador de densidad de la siguiente manera: se calcula el número de peatones que desean moverse hacia arriba en un área rectangular al lado de la puerta que encierre a los peatones azules, de igual manera para los peatones naranja, observar la figura 5.4; se calcula la diferencia entre ambos números para cada lapso temporal. A la diferencia de estos números se le considerara como una especie de presión  $\Delta P$ . En la figura 5.5 se muestra esta presión y la frecuencia asociada al movimiento.

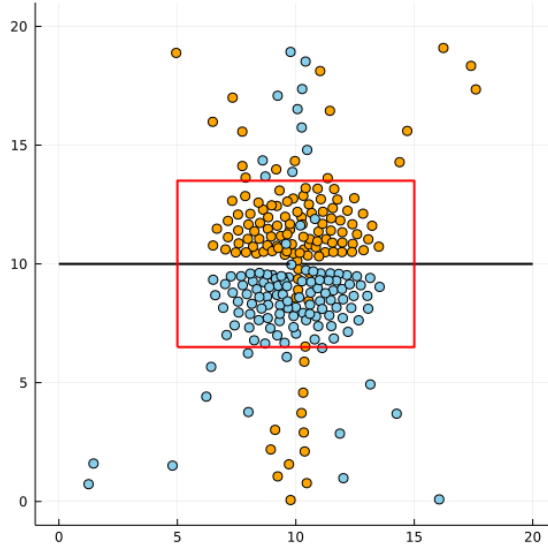


FIGURA 5.4. Área usada para el cálculo de la diferencia de presión a través de la puerta.

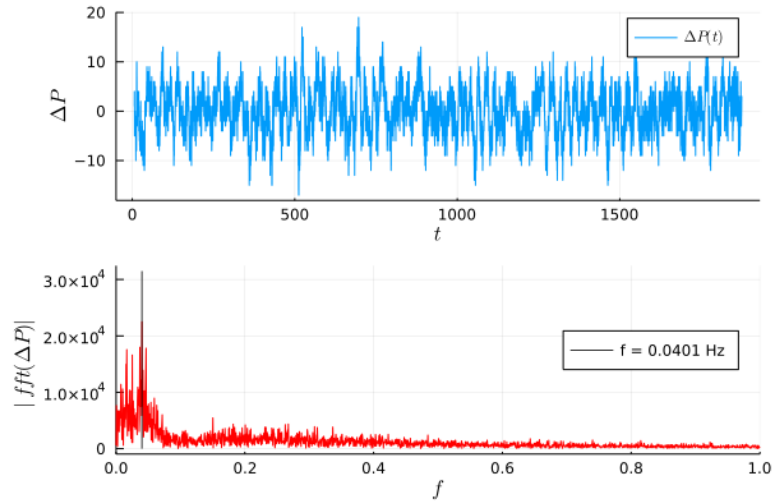


FIGURA 5.5. Diferencia de presión  $\Delta P$ , como función del tiempo para los peatones representados en la figura 5.4. La gráfica roja muestra la transformada de Fourier y la línea negra el pico de la frecuencia.

Por otra parte, usando la autocorrelación (4.1) del flujo de peatones con respecto al tiempo se obtienen las gráficas 5.6, donde se muestra el flujo, su autocorrelación y la transformada de Fourier de la autocorrelación.

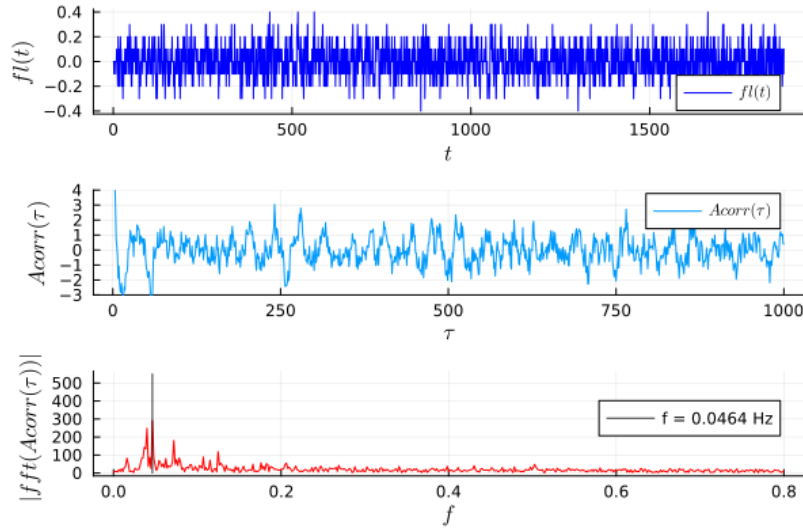


FIGURA 5.6. Funciones de flujo  $f_l(t)$ , autocorrelación  $A_{corr}(\tau)$  y el valor absoluto de la transformada de Fourier  $|f_{ft}(A_{corr}(\tau))|$ , cuya frecuencia tiene como resultado  $f = 0.0464$  Hz.

Del análisis realizado en las figuras 5.5 y 5.6 se tiene el cuadro 5.3 con las respectivas frecuencias y periodos del movimiento además de sus formas adimensionales.

CUADRO 5.3. Periodos, frecuencias y sus valores adimensionales.

$\Delta P$	$A_{corr}(\tau)$
$f_p = 0.0401$ Hz $T_p = 24.94$ s	$f_p = 0.0464$ Hz $T_p = 21.55$ s
$f_{ap} = 0.02005$ $T_{ap} = 49.88$	$f_{ap} = 0.0232$ $T_{ap} = 43.10$

el subíndice  $p$  hace referencia a una variable del oscilador peatonal.

Otra manera de obtener la adimensionalización de las frecuencias es usando el número de Strouhal [14]

$$St = \frac{fR}{v} \quad (5.10)$$

donde  $f$  es la frecuencia del fluido (o la frecuencia del flujo neto en el oscilador peatonal),  $R$  el diámetro del orificio (o el diámetro de la puerta en el oscilador peatonal) y  $v$  la velocidad terminal en el fluido (la velocidad deseada en el oscilador peatonal).

A partir de los datos de la tabla 5.3 y las ecuaciones (5.9) y (5.10), se tienen los parámetros adimensionales en la descripción de ambos osciladores, ver cuadro 5.4.

CUADRO 5.4. *Periodos, frecuencias y número de Strouhal.*

	Frecuencia		Periodo	Número de Strouhal
Oscilador de Densidad	$(\Delta P)$	0.0434	23.07	0.0648
Oscilador Peatonal	$(\Delta P)$	0.02005	49.88	0.0150
	$(Acorr(\tau))$	0.0232	43.10	0.0173

Como se puede observar del cuadro 5.4 la frecuencia del oscilador de densidad es 2.17 veces más grande que la frecuencia del oscilador peatonal, mientras que el periodo es 0.46 veces más pequeño, lo anterior para el análisis del oscilador peatonal por medio de la presión ejercida sobre la puerta. Con el uso de la función de autocorrelación se tiene la frecuencia del oscilador de densidad es 1.87 veces más grande que la frecuencia del oscilador peatonal, mientras que el periodo es 0.54 veces más pequeño. Por otra parte, el número de Strouhal del oscilador de densidad es 4.34 veces más grande que el del oscilador peatonal en el análisis de presiones y el número de Strouhal del oscilador de densidad es 3.74 veces más grande que el del oscilador peatonal.

Las conclusiones preliminares son

$$\begin{aligned}
 f_{ad} &\rightarrow 2.17f_{ap}, 1.87f_{ap} \\
 T_{ad} &\rightarrow 0.46T_{ap}, 0.54T_{ap} \\
 St_d &\rightarrow 4.34St_p, 3.74St_p
 \end{aligned}
 \tag{5.11}$$

donde el subíndice  $a$  se refiere a un número adimensional y los subíndices  $p$  y  $d$ , oscilador peatonal y al oscilador de densidad respectivamente. La coma del lado derecho de la ecuación (5.11) separa los resultados del análisis sobre el oscilador peatonal primero por medio de la función presión y segundo con la autocorrelación.

Este resultado, aunque no es concluyente, permite pensar que se puede profundizar entre el oscilador de densidad y el oscilador peatonal.

## Conclusiones

La motivación de este trabajo fue la comparación cuantitativa de dos fenómenos que a primera vista parecieran muy similares cualitativamente. El contexto de los dos fenómenos es muy diferente, aunque en ambos casos ocurre un flujo a través de un orificio, flujo que se frena y luego se invierte. Para hacer la comparación se utilizaron números adimensionales según una propuesta de Florence Raynal. Los resultados son interesantes, en el sentido de que se obtienen para los mismos valores de los números adimensionales  $\alpha$  frecuencias adimensionales del mismo orden de magnitud. Sin embargo, una comparación más detallada es necesaria. Se necesita comparar más casos en los que  $\alpha$  coincida para los dos tipos de oscilador. El resultado principal de la comparación se resalta en el cuadro 5.4 y la ecuación (5.11) donde se observa una relación aproximada de  $2 : (1/2) : 4$  para las frecuencias, periodos y el número de Strouhal entre el oscilador de densidad y el oscilador peatonal, es decir, el oscilador de densidad tiene el doble de frecuencia, la mitad del periodo y cuatro veces el número de Strouhal comparado con el oscilador peatonal. Como conclusión, se encontró una analogía entre ambos experimentos, al comparar la frecuencia, periodo y el número de Strouhal y notar que presentan valores con una cierta relación.

Otro aspecto del trabajo que nos parece interesante es que bajo ciertas situaciones se pueden comparar ciertos fenómenos de la Física con problemas sociales, en este caso se compararon algunas ideas de Mecánica de Fluidos con problemas de tráfico peatonal y probablemente esta comparación se pueda extender a otros tipos de tráfico.

También se compararon fenómenos de formación de filas peatonales en el modelo de Helbing-Molnár con las filas que se forman en la inestabilidad de Rayleigh-Taylor. En la capítulo 3.2 se simula la Inestabilidad de Rayleigh-Taylor y se encuentra una relación lineal entre el número de líneas de flujo y el ancho de la interacción entre ambos fluidos, la relación lineal es  $N(W) = (6.12857)W - 4.9619$ . Para los peatones que atraviesan una calle en el capítulo 4.1 se encuentra que existe también una relación lineal entre el número de filas de peatones y el ancho de la calle dada de la forma  $N(W) = (0.435789 \text{ m}^{-1})W + 1.21158$ . Ambos fenómenos presentan una relación lineal en esta interacción.

Algo que se muestra en esta Tesis es que los fenómenos de formación de filas son un antecedente de la formación de osciladores. Si en lugar de que toda la calle este abierta, para el movimiento opuesto de dos grupos de peatones, se pone una constricción, una pared con una puerta por ejemplo, surge naturalmente el oscilador peatonal. Como muestra Jearl Walker en “La Feria Ambulante de la Física” [15], la formación de los dedos de agua salada y agua pura en la inestabilidad de Rayleigh Taylor es un antecedente a la formación del oscilador salino, la forma más simple de realizar el oscilador de densidad experimentalmente.

En el caso del oscilador de densidad lo que produce la inversión del flujo son los cambios de presión al cambiar los niveles de líquido en los recipientes. Para hacer más profunda la analogía se utilizó en este trabajo de Tesis, en el caso del oscilador peatonal, una especie de presión que mide la aglomeración de peatones alrededor del orificio. Se definió entonces una diferencia de presiones también en este caso. La frecuencia de esta diferencia de presiones coincide con la frecuencia del oscilador peatonal y la idea es que cuando esta diferencia de presiones es adversa al flujo en un sentido empieza a frenar este hasta que eventualmente se detiene y se invierte.

En la tesis se habló de que existe un enfoque supermacroscópico en los fenómenos de tráfico peatonal y se hizo notar que en este contexto la dinámica se parece mucho a un fenómeno hidrodinámico. El modelo de Helbing-Molnár es microscópico, por lo que se comporta mucho más como un fenómeno discreto. Una comparación con el modelo supermacroscópico es una de las posibilidades de un trabajo a futuro.

Para dar más solidez a la presente comparación entre el oscilador de densidad y el oscilador peatonal se necesita comparar más experimentos con distintos parámetros  $\alpha$ , además de mejorar el método empleado para la solución de las ecuaciones diferenciales y los tiempos de ejecución de los programas para implementar más métodos estadísticos asociados. Se deja como trabajo a futuro.

Otro aspecto que no puede dejarse de lado es hacer estudios experimentales en el caso del oscilador de densidad y observaciones de fenómenos que surgen en la dinámica real de peatones en calles y lugares con una fuerte aglomeración de gente como el metro de la Ciudad de México.

## Referencias

- [1] Francois Charru, *Hydrodynamic Instabilities*, Cambridge University Press, 2011.
- [2] Dirk Helbing and Péter Molnár, *Social force model for pedestrian dynamics*. Physical Review E 51, 4282–4286, 1995.
- [3] Seelye Martin, *A hydrodynamic curiosity: The salt oscillator*, *Geophysical Fluid Dynamics*, 1970, 1:1-2, 143-160.
- [4] Kenji Aoki, *Mathematical model of a saline oscillator*, *Physica D* 147 (2000) 187–203.
- [5] Iain G. Currie, *Fundamental Mechanics of Fluids*, Third Edition, Marcel Dekker, Inc., 1996.
- [6] Richard P. Feynman, *The Feynman Lectures on Physics*, Addison-Wesley Publishing Company, Inc., 1964.
- [7] Philip Ball, *Critical Mass, How one thing leads to another*, Farrar, Straus and Giroux, New York, 2004.
- [8] Alessandro Corbetta and Federico Toschi, *Physics of Human Crowds*, *Annu. Rev. Condens. Matter Phys.* 2023. 14:311–33.
- [9] Per-Henrik Alfredsson and Torgny Lagerstedt *The behavior of the density oscillator*, *Physics of Fluids* 24, 10 (1981).
- [10] W. Fokou Kenfack, M. Siewe Siewe, T.C. Kofane *Nonlinear dynamics and synchronization of saline oscillator's model*, *Chaos, Solitons and Fractals* 82 (2016) 72–82.
- [11] W. Fokou Kenfack, M. Siewe Siewe, T.C. Kofane *Sliding motion and bifurcation in saline oscillator's model and liquid's density measurement using saline oscillator*, *Chaos, Solitons and Fractals* 114 (2018) 321–331.
- [12] Donald A. McQuarrie, *Statistical Mechanics*, Harper & Row Publishers, 1973.
- [13] Randall J. Le Veque, *Finite-Volume Methods for Hyperbolic Problems*, Cambridge University Press, 2004.
- [14] E. J. López, G. Ruiz, *Transport of Particles in a Periodically Forced Flow*, *Experimental and Computational Fluid Mechanics*, Springer, 2014, 271-277.
- [15] Jearl Walker, *The Flying Circus of Physics WITH ANSWERS*, John Wiley & Sons, 1977, 80.
- [16] <https://julialang.org/>
- [17] [https://gfs.sourceforge.net/wiki/index.php/Main\\_Page](https://gfs.sourceforge.net/wiki/index.php/Main_Page)
- [18] <http://gerris.dalembert.upmc.fr/gerris/examples/examples/rt.html#htoc6>
- [19] [https://gfs.sourceforge.net/wiki/index.php/FAQ#Physics\\_and\\_dimensioning](https://gfs.sourceforge.net/wiki/index.php/FAQ#Physics_and_dimensioning)

## Desarrollo de las ecuaciones para el modelo de Helbing y Molnár

Se analizan los elementos de la ecuación (1.6)

$$\begin{aligned}\|\mathbf{r}_{\alpha\beta}\| &= \|\mathbf{r}_\alpha - \mathbf{r}_\beta\| \\ &= \|(x_\alpha - x_\beta)\mathbf{e}_x - (y_\alpha - y_\beta)\mathbf{e}_y\| \\ &= [(x_\alpha - x_\beta)^2 + (y_\alpha - y_\beta)^2]^{1/2}\end{aligned}$$

la derivada parcial

$$\begin{aligned}\frac{\partial\|\mathbf{r}_{\alpha\beta}\|}{\partial(x_\alpha - x_\beta)} &= \frac{1}{2} [(x_\alpha - x_\beta)^2 + (y_\alpha - y_\beta)^2]^{-1/2} 2(x_\alpha - x_\beta) \\ &= \frac{x_\alpha - x_\beta}{\|\mathbf{r}_{\alpha\beta}\|}\end{aligned}$$

de la misma manera

$$\frac{\partial\|\mathbf{r}_{\alpha\beta}\|}{\partial(y_\alpha - y_\beta)} = \frac{y_\alpha - y_\beta}{\|\mathbf{r}_{\alpha\beta}\|}$$

$$\begin{aligned}\nabla_{\mathbf{r}_{\alpha\beta}}(\|\mathbf{r}_{\alpha\beta}\|) &= \frac{\partial\|\mathbf{r}_{\alpha\beta}\|}{\partial(x_\alpha - x_\beta)}\mathbf{e}_x + \frac{\partial\|\mathbf{r}_{\alpha\beta}\|}{\partial(y_\alpha - y_\beta)}\mathbf{e}_y \\ &= \frac{x_\alpha - x_\beta}{\|\mathbf{r}_{\alpha\beta}\|}\mathbf{e}_x + \frac{y_\alpha - y_\beta}{\|\mathbf{r}_{\alpha\beta}\|}\mathbf{e}_y \\ &= \frac{\mathbf{r}_{\alpha\beta}}{\|\mathbf{r}_{\alpha\beta}\|}\end{aligned}$$

$$\begin{aligned}\frac{\partial\|\mathbf{r}_{\alpha\beta} - v_\beta\Delta t\mathbf{e}_\beta\|}{\partial(x_\alpha - x_\beta)} &= \frac{\partial}{\partial(x_\alpha - x_\beta)} [(x_\alpha - x_\beta - v_\beta\Delta t e_{\beta x})^2 + (y_\alpha - y_\beta - v_\beta\Delta t e_{\beta y})^2]^{1/2} \\ &= \frac{1}{2} [(x_\alpha - x_\beta - v_\beta\Delta t e_{\beta x})^2 + (y_\alpha - y_\beta - v_\beta\Delta t e_{\beta y})^2]^{-1/2} 2(x_\alpha - x_\beta - v_\beta\Delta t e_{\beta x}) \\ &= \frac{(x_\alpha - x_\beta - v_\beta\Delta t e_{\beta x})}{\|\mathbf{r}_{\alpha\beta} - v_\beta\Delta t\mathbf{e}_\beta\|}\end{aligned}$$

en forma análoga para la otra componente

$$\frac{\partial\|\mathbf{r}_{\alpha\beta} - v_\beta\Delta t\mathbf{e}_\beta\|}{\partial(y_\alpha - y_\beta)} = \frac{(y_\alpha - y_\beta - v_\beta\Delta t e_{\beta y})}{\|\mathbf{r}_{\alpha\beta} - v_\beta\Delta t\mathbf{e}_\beta\|}$$

entonces

$$\begin{aligned}\nabla_{\mathbf{r}_{\alpha\beta}}(\|\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}\|) &= \frac{\partial\|\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}\|}{\partial(x_{\alpha} - x_{\beta})}\mathbf{e}_x + \frac{\partial\|\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}\|}{\partial(y_{\alpha} - y_{\beta})}\mathbf{e}_y \\ &= \frac{\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}}{\|\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}\|}\end{aligned}$$

Ahora la función gradiente de la ecuación (1.6)

$$\begin{aligned}\nabla_{\mathbf{r}_{\alpha\beta}}b(\mathbf{r}_{\alpha\beta}) &= \nabla_{\mathbf{r}_{\alpha\beta}}\left\{\frac{1}{2}\left[\left(\|\mathbf{r}_{\alpha\beta}\| + \|\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}\|\right)^2 - (v_{\beta}\Delta t)^2\right]^{1/2}\right\} \\ &= \frac{1}{4}\left[\left(\|\mathbf{r}_{\alpha\beta}\| + \|\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}\|\right)^2 - (v_{\beta}\Delta t)^2\right]^{-1/2} \\ &\quad \nabla_{\mathbf{r}_{\alpha\beta}}\left\{\left(\|\mathbf{r}_{\alpha\beta}\| + \|\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}\|\right)^2 - (v_{\beta}\Delta t)^2\right\} \\ &= \frac{1}{8}\frac{1}{\frac{1}{2}\left[\left(\|\mathbf{r}_{\alpha\beta}\| + \|\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}\|\right)^2 - (v_{\beta}\Delta t)^2\right]^{1/2}} \\ &\quad \left\{\nabla_{\mathbf{r}_{\alpha\beta}}\left(\|\mathbf{r}_{\alpha\beta}\| + \|\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}\|\right)^2 - \nabla_{\mathbf{r}_{\alpha\beta}}(v_{\beta}\Delta t)^2\right\} \\ &= \frac{1}{8b}2\left(\|\mathbf{r}_{\alpha\beta}\| + \|\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}\|\right)\left\{\nabla_{\mathbf{r}_{\alpha\beta}}\|\mathbf{r}_{\alpha\beta}\| + \nabla_{\mathbf{r}_{\alpha\beta}}\|\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}\|\right\} \\ &= \frac{\left(\|\mathbf{r}_{\alpha\beta}\| + \|\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}\|\right)}{4b}\left\{\frac{\mathbf{r}_{\alpha\beta}}{\|\mathbf{r}_{\alpha\beta}\|} + \frac{\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}}{\|\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}\|}\right\} \\ &= \frac{1}{4b}\left\{\mathbf{r}_{\alpha\beta} + \frac{\mathbf{r}_{\alpha\beta}}{\|\mathbf{r}_{\alpha\beta}\|} + \frac{\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}}{\|\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}\|}\right\} \\ &= \frac{1}{4b}\left\{\mathbf{r}_{\alpha\beta} + \frac{\|\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}\|}{\|\mathbf{r}_{\alpha\beta}\|}\mathbf{r}_{\alpha\beta} + \mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta} + \frac{\|\mathbf{r}_{\alpha\beta}\|}{\|\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}\|}(\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta})\right\} \\ &= \frac{1}{4b}\left\{\left(\frac{\|\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}\|}{\|\mathbf{r}_{\alpha\beta}\|} + 1\right)\mathbf{r}_{\alpha\beta} + \left(1 + \frac{\|\mathbf{r}_{\alpha\beta}\|}{\|\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}\|}\right)(\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta})\right\} \\ &= \frac{1}{4b}\left\{\left(2\frac{\|\mathbf{r}_{\alpha\beta}\|\|\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}\|}{\|\mathbf{r}_{\alpha\beta}\|\|\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}\|} + \frac{\|\mathbf{r}_{\alpha\beta}\|^2}{\|\mathbf{r}_{\alpha\beta}\|\|\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}\|} + \frac{\|\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}\|^2}{\|\mathbf{r}_{\alpha\beta}\|\|\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}\|}\right)\mathbf{r}_{\alpha\beta}\right. \\ &\quad \left.- \left(\frac{\|\mathbf{r}_{\alpha\beta}\|\|\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}\|}{\|\mathbf{r}_{\alpha\beta}\|\|\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}\|} + \frac{\|\mathbf{r}_{\alpha\beta}\|^2}{\|\mathbf{r}_{\alpha\beta}\|\|\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}\|}\right)v_{\beta}\Delta t \mathbf{e}_{\beta}\right\} \\ &= \frac{1}{4b}\left\{\left[\frac{\left(\|\mathbf{r}_{\alpha\beta}\| + \|\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}\|\right)^2}{\|\mathbf{r}_{\alpha\beta}\|\|\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}\|}\right]\mathbf{r}_{\alpha\beta} - \left(\frac{\|\mathbf{r}_{\alpha\beta}\|\|\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}\| + \|\mathbf{r}_{\alpha\beta}\|^2}{\|\mathbf{r}_{\alpha\beta}\|\|\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}\|}\right)v_{\beta}\Delta t \mathbf{e}_{\beta}\right\}\end{aligned}$$

para simplificar la expresión anterior sean

$$f = \|\mathbf{r}_{\alpha\beta}\| \quad \text{y} \quad g = \|\mathbf{r}_{\alpha\beta} - v_{\beta}\Delta t \mathbf{e}_{\beta}\|$$

entonces

$$\nabla_{\mathbf{r}_{\alpha\beta}}b(\mathbf{r}_{\alpha\beta}) = \frac{1}{4b}\left\{\left[\frac{(f+g)^2}{fg}\right]\mathbf{r}_{\alpha\beta} - \left(\frac{fg+f^2}{fg}\right)v_{\beta}\Delta t \mathbf{e}_{\beta}\right\}$$

$$\begin{aligned}\nabla_{\mathbf{r}_{\alpha\beta}} b(\mathbf{r}_{\alpha\beta}) &= \frac{1}{4b} \left\{ \left[ \frac{(f+g)^2}{fg} \right] \mathbf{r}_{\alpha\beta} - \left( \frac{fg+f^2}{fg} \right) v_\beta \Delta t \mathbf{e}_\beta \right\} \\ &= \frac{1}{4bfg} \left\{ (f+g)^2 \mathbf{r}_{\alpha\beta} - f(f+g) v_\beta \Delta t \mathbf{e}_\beta \right\}\end{aligned}$$

Ahora una función que depende de  $b$  tiene gradiente

$$\begin{aligned}\nabla_{\mathbf{r}_{\alpha\beta}} V_{\alpha\beta}[b(\mathbf{r}_{\alpha\beta})] &= \frac{d}{db} V_{\alpha\beta}(b) \nabla_{\mathbf{r}_{\alpha\beta}} b(\mathbf{r}_{\alpha\beta}) \\ &= \frac{dV_{\alpha\beta}(b)}{db} \frac{1}{4bfg} \left\{ (f+g)^2 \mathbf{r}_{\alpha\beta} - f(f+g) v_\beta \Delta t \mathbf{e}_\beta \right\}\end{aligned}$$

La función que depende de  $\|\mathbf{r}_{\alpha B}\|$  tiene gradiente

$$\begin{aligned}\nabla_{\mathbf{r}_{\alpha B}} U_{\alpha B}(\|\mathbf{r}_{\alpha B}\|) &= \frac{d}{d\|\mathbf{r}_{\alpha B}\|} U_{\alpha B}(\|\mathbf{r}_{\alpha B}\|) \nabla_{\mathbf{r}_{\alpha B}} (\|\mathbf{r}_{\alpha B}\|) \\ &= \frac{d}{d\|\mathbf{r}_{\alpha B}\|} U_{\alpha B}(\|\mathbf{r}_{\alpha B}\|) \frac{\mathbf{r}_{\alpha B}}{\|\mathbf{r}_{\alpha B}\|}\end{aligned}$$

Los principales resultados de las ecuaciones anteriores son

$$\begin{aligned}f &= \|\mathbf{r}_{\alpha\beta}\| \quad \text{y} \quad g = \|\mathbf{r}_{\alpha\beta} - v_\beta \Delta t \mathbf{e}_\beta\| \\ b(\mathbf{r}_{\alpha\beta}) &= \frac{1}{2} \sqrt{(f+g)^2 - (v_\beta \Delta t)^2} \\ \nabla_{\mathbf{r}_{\alpha\beta}} b(\mathbf{r}_{\alpha\beta}) &= \frac{1}{4bfg} \left\{ (f+g)^2 \mathbf{r}_{\alpha\beta} - f(f+g) v_\beta \Delta t \mathbf{e}_\beta \right\} \\ \nabla_{\mathbf{r}_{\alpha\beta}} V_{\alpha\beta}[b(\mathbf{r}_{\alpha\beta})] &= \frac{d}{db} V_{\alpha\beta}(b) \nabla_{\mathbf{r}_{\alpha\beta}} b(\mathbf{r}_{\alpha\beta}) \\ \nabla_{\mathbf{r}_{\alpha B}} U_{\alpha B}(\|\mathbf{r}_{\alpha B}\|) &= \frac{d}{d\|\mathbf{r}_{\alpha B}\|} U_{\alpha B}(\|\mathbf{r}_{\alpha B}\|) \frac{\mathbf{r}_{\alpha B}}{\|\mathbf{r}_{\alpha B}\|}\end{aligned}$$

## Apéndice B

### Programa en Gerris

Se toma como base el programa que trata sobre la inestabilidad de Rayleigh-Taylor implementado en Gerris por Stéphane Popinet [17]. Al programa se le cambia la viscosidad dinámica y la ecuación que separa los dos fluidos con distinta densidad.

```
# Title: Rayleigh-Taylor instability
#
# Description:
#
# A classical test case for the flow of two fluids of different
# densities. A sinusoidal interface separates the two fluids. The
# heavier fluid is on top. A mushroom-shaped instability develops in
# time.

# Author: St\`ephane Popinet
# Command: gerris2D rt.gfs

16 24 GfsSimulation GfsBox GfsGEdge {} {
  Time { end = 3 dtmax = 5e-3 }
  Refine 7

  # The tracer T is used to track both phases
  VariableTracerVOF {} T

  # The initial constant interface (translated by 1.5 along the y-axis)
  InitFraction {} T (0.0 + y) { ty = 1.5 }

  AdaptVorticity { istep = 1 } { maxlevel = 7 cmax = 2e-2 }
  AdaptGradient { istep = 1 } { maxlevel = 7 cmax = 1e-2 } T

  # The dynamic viscosity for both phases
  SourceViscosity {} 0.00313

  # This defines the inverse of the density of the fluids as a
  # function of T
  PhysicalParams { alpha = 1./(T*1.225 + (1. - T)*0.1694) }

  # We also need gravity
  Source {} V -9.81

  OutputTime { istep = 10 } stderr
```



```

14 15 top
15 11 left
15 16 top
16 12 left

```

Al programa de Stéphane Popinet, se le cambia el valor para la viscosidad dinámica, se colocan fronteras de tal forma que se represente la figura 1.1, se agrega una un tercer fluido con una densidad casi nula y dos ecuaciones que representan la división entre los fluidos.

```

# Author: St\`ephane Popinet
# Command: gerris2D rt.gfs
# Version: 1.1.0

```

```

# Dominio de celdas

```

```

# |13|3 |7 |
# |14|2 |8 |
# |15|1 |9 |
# |16|4 |10|
# |17|5 |11|
# |18|6 |12|

```

```

# El origen se encuentra en el centro de la primera celda

```

```

18 23 GfsSimulation GfsBox GfsGEdge {} {

```

```

  Time { end = 50 dtmax = 5e-3 }

```

```

  Refine 7

```

```

  # Con Gfsview se observan  $2^7 = 128$  cuadros por celda

```

```

  # The tracer T is used to track both phases

```

```

  VariableTracerVOF {} T

```

```

  VariableTracerVOF {} T1

```

```

  # Interface entre el fluido denso dentro del recipiente y

```

```

  # el fluido ligero fuera del recipiente

```

```

  #  $\tanh[A(|x-x_0| - b)](-c/2) + y_0$ 

```

```

  # -----

```

```

  #       | . |

```

```

  #       |_____|

```

```

  # (x0, y0) coordenadas del punto en la figura

```

```

  # b es la distancia del punto a las paredes

```

```

  # c es la altura de las paredes

```

```

  # A corresponde a la pendiente de las paredes

```

```

  InitFraction {} T (( -(1.5/2.0)*tanh(100000*(fabs(x)-0.5))) + y) { ty = 0.25 }

```

```

  InitFraction {} T1 (-1.0 + y) { ty = 0.0}

```

```

  # Se insertan dos fronteras sólidas correspondientes a dos prismas

```

```

  # rectangulares

```

```

  # que se construyen con la función cube(...){sy = ...}

```

```

# Con Gfsview se observan 2^6 = 64 cuadros de longitud con centro
# (-33, -64)=(-0.2578125, -0.5)
Solid (cube(0.0, 0.0, 0.0, 0.5)) { sy = 0.015625 tx = -0.2578125 ty = -0.5}

# Orificio con 2 cuadros = 0.015625
# El primer sólido deja un espacio de un cuadro a la izquierda del centro
# el segundo de un cuadro a la derecha del centro

# Con Gfsview se observan 2^6 = 64 cuadros de longitud con centro
# (33, -64)=(0.2578125, -0.5)
# 2u = 2*(1/(2^7)) = 0.015625
Solid (cube(0.0, 0.0, 0.0, 0.5)) { sy = 0.015625 tx = 0.2578125 ty = -0.5}

AdaptVorticity { istep = 1 } { maxlevel = 7 cmax = 2e-2 }
AdaptGradient { istep = 1 } { maxlevel = 7 cmax = 1e-2 } T
AdaptGradient { istep = 1 } { maxlevel = 7 cmax = 1e-2 } T1

# Viscosidad Dinámica para ambas fases
SourceViscosity {} 0.00113

# Inverso de la densidad de los fluidos como función de T y T1
PhysicalParams { alpha = 1./((1-T1)*(T*1.15 + (1. - T)*1.0) + T1*0.006) }

# Aceleración de la gravedad
Source {} V -9.81

OutputLocation { step = 0.1 } datos/osc-den-%3.1f.dat XYZ.dat
OutputTime { istep = 10 } stderr
OutputBalance { istep = 10 } stderr
OutputProjectionStats { istep = 10 } stderr
OutputDiffusionStats { istep = 10 } stderr
OutputPPM { istep = 2 } { ppm2mpeg > vort.mpg } {
  min = -30 max = 30 v = Vorticity
}
OutputPPM { istep = 2 } { ppm2mpeg > t.mpg } {
  min = 0 max = 1 v = T
}
OutputTiming { start = end } stderr
OutputSimulation { step = 0.1 } stdout
}
GfsBox {}
GfsBox {}
GfsBox {}
GfsBox {}
GfsBox {}
GfsBox {}

```

```
GfsBox {}
GfsBox {}
GfsBox {}
GfsBox {}
GfsBox {}
GfsBox {}
GfsBox {}
GfsBox {}
GfsBox {}
GfsBox {}
GfsBox {}
GfsBox {}
1 2 top
1 4 bottom
2 3 top
3 13 left
4 16 left
4 10 right
4 5 bottom
5 17 left
5 11 right
5 6 bottom
6 18 left
6 12 right
12 11 top
11 10 top
10 9 top
9 8 top
8 7 top
18 17 top
17 16 top
16 15 top
15 14 top
14 13 top
3 7 right
```

## Programa en Julia (Calle con peatones).

A continuación se muestra un programa creado con el lenguaje de programación Julia[16], llamado “MFS`calle.jl`”, donde se implementan las ecuaciones necesarias para representar el movimiento de los peatones que atraviesan una calle en ambas direcciones. El programa se basa en el modelo descrito por Helbing y Molnar[2].

En la primera parte del programa se implementan conjuntos de datos `Peaton`, `ParPeaton`, `ParEnt`, `DatosGif`, que determinan como se van a agrupar los valores necesarios correspondientes al inicio del experimento, es decir, las variables involucradas para cada peatón.

La segunda parte consiste en diseñar las ecuaciones del modelo de Helbing y Molnár y como van a operar con los conjuntos descritos en el párrafo anterior. También se crean las funciones para graficar y crear gifs.

Por último se desarrolla una función general `fun_Matrix_Pos` que utiliza todas las funciones de la segunda parte y crea una matriz con las posiciones y velocidades de todos los peatones para cada intervalo de tiempo. Y finalmente se guarda esta matriz como un archivo de datos con extensión CSV.

```
#####
# Funciones Modelo de Fuerza Social
#####

#=#
En este módulo las fórmulas para el Modelo de Fuerza Social.

René Mora Maya

Lunes 06-02-23
=#
# La siguiente instrucción sirve para *precompilar* el módulo
__precompile__(true)

module MFS
    import Base: +, -, *, /, ^, ==, &, |
    import Plots: @animate
    using Plots

mutable struct Peaton
    pos::Vector{Float64} # Posición
    vel::Vector{Float64} # Velocidad
    dir::Vector{Float64} # Punto de destino
    v_0::Float64        # Magnitud de vel. deseada
end
```

```

    v_max::Float64      # Magnitud de vel. máxima
end
mutable struct ParPeaton
    tau::Float64      # Tiempo de relajación
    delta_t::Float64 # Tiempo asociado al paso de beta
    V_0::Float64      # Inter. entre peatones
    sigma::Float64    # Inter. entre peatones
    U_0::Float64      # Inter. entre fronteras
    R::Float64        # Inter. entre fronteras
end
mutable struct ParEnt
    largo::Float64    # Largo de la calle
    ancho::Float64    # Ancho de la calle
    orientacion::String # horizontal o vertical
    cc::Float64       # Distancia de interacción entre peatones
end
mutable struct DatosGif
    # Intervalos del gif (a[1]=inicio, a[2]=paso, a[3]=fin)
    a::Vector{Int64}
    # Colores de las gráficas (:red, :blue)
    colores::Vector{Symbol}
    # Tamaño de los puntos graficados ([3,4])
    puntos::Vector{Int64}
    # Tamaño de la gráfica ([500, 500])
    tam::Vector{Int64}
    # Intervalo de los ejes ([[0.0, 50.0), (0.0, 10.0)])
    int_ejes::Vector{Tuple{Float64, Float64}}
    # Archivo con terminación "datos.gif"
    arch::String
    # Número de imágenes por segundo
    fps::Int64
    # Directorio con imágenes
    imag::String
end
function fun_arch_datos(M::Matrix{Float64},
    arch_nombre::String)

    ARC = open( arch_nombre, "w" );

    for i in 1:length(M[1:end, 1])
        for j in 1:length(M[1, 1:end])
            write(ARC, "$(M[i, j]) ")
        end
        write(ARC, "\n")
    end
    close(ARC)
end
function fun_trayectoria(M::Matrix{Float64},

```

```

        Ni::Int64,
        Nd::Int64,
        graf::MFS.DatosGif,
        j::Int64) # Peatón j

scatter(M[1:end, j], M[1:end, Ni + Nd + j], color = graf.colores[1],
        legend =:false, marker = graf.puntos[1], size = graf.tam)
xlims!(graf.int_ejes[1])
ylims!(graf.int_ejes[2])
end
function fun_animacion_color(M::Matrix{Float64},
        Ni::Int64,          # Peatones del lado izquierdo
        Nd::Int64,          # Peatones del lado derecho
        graf::MFS.DatosGif,
        pe::MFS.ParEnt)

N = Ni + Nd;
anim = @animate for i in graf.a[1]:graf.a[2]:graf.a[3]
    scatter(M[i, 1:Ni], M[i, N+1:N+Ni], color = graf.colores[1],
            legend =:false, marker = graf.puntos[1], size = graf.tam)
    scatter!(M[i, Ni + 1:N], M[i, N+Ni+1:2N ], color = graf.colores[2],
            legend =:false, marker = graf.puntos[2])
    if(pe.orientacion == "horizontal")
        plot!([0.0, pe.largo], [0.0, 0.0], color = graf.colores[3],
            legend =:false, line = 5)
        plot!([0.0, pe.largo], [pe.anch, pe.anch],
            color = graf.colores[3], legend =:false, line = 5)
    elseif(pe.orientacion == "vertical")
        plot!([0.0, 0.0], [0.0, pe.anch], color = graf.colores[3],
            legend =:false, line = 5)
        plot!([pe.largo, pe.largo], [0.0, pe.anch],
            color = graf.colores[3], legend =:false, line = 5)
    end
    xlims!(graf.int_ejes[1])
    ylims!(graf.int_ejes[2])
    savefig(graf.imag*"$i")
end

return gif(anim, graf.arch*".gif", fps = graf.fps)
end
##### DataFrames #####
function fun_trayectoria(M::Matrix{Union{Missing, Float64}},
        Ni::Int64,
        Nd::Int64,
        graf::MFS.DatosGif,
        j::Int64) # Peatón j

scatter(M[1:end, j], M[1:end, Ni + Nd + j], color = graf.colores[1],

```

```

        legend =:false, marker = graf.puntos[1], size = graf.tam)
    xlims!(graf.int_ejes[1])
    ylims!(graf.int_ejes[2])
end
function fun_animacion_color(M::Matrix{Union{Missing, Float64}},
    Ni::Int64,          # Peatones del lado izquierdo
    Nd::Int64,          # Peatones del lado derecho
    graf::MFS.DatosGif,
    pe::MFS.ParEnt)

    N = Ni + Nd;
    anim = @animate for i in graf.a[1]:graf.a[2]:graf.a[3]
        scatter(M[i, 1:Ni], M[i, N+1:N+Ni], color = graf.colores[1],
            legend =:false, marker = graf.puntos[1], size = graf.tam)
        scatter!(M[i, Ni + 1:N], M[i, N+Ni+1:2N ], color = graf.colores[2],
            legend =:false, marker = graf.puntos[2])
        if(pe.orientacion == "horizontal")
            plot!([0.0, pe.largo], [0.0, 0.0], color = graf.colores[3],
                legend =:false, line = 5)
            plot!([0.0, pe.largo], [pe.ancho, pe.ancho],
                color = graf.colores[3], legend =:false, line = 5)
        elseif(pe.orientacion == "vertical")
            plot!([0.0, 0.0], [0.0, pe.ancho], color = graf.colores[3],
                legend =:false, line = 5)
            plot!([pe.largo, pe.largo], [0.0, pe.ancho],
                color = graf.colores[3], legend =:false, line = 5)
        end
        xlims!(graf.int_ejes[1])
        ylims!(graf.int_ejes[2])
        savefig(graf.imag*"${i}")
    end

    return gif(anim, graf.arch*".gif", fps = graf.fps)
end

fun_norma(x::Vector{Float64}) = (x[1]^2 + x[2]^2)^(1/2);

# Fuerza de aceleración
function fun_e_a(x::Peaton,
    pe::ParEnt)

    return (x.dir - x.pos)/fun_norma(x.dir - x.pos);
end
fun_Fa(x::Peaton, y::ParPeaton, pe::ParEnt) =
(x.v_0 * fun_e_a(x, pe) - x.vel)/y.tau;

# Interacción con los peatones
function fun_fab(x::Peaton,

```

```

        y::Peaton,
        par::ParPeaton,
        pe::ParEnt)

    r_ab = x.pos - y.pos;
    v_b = fun_norma(y.vel);
    e_b = fun_e_a(y, pe);
    dt = par.delta_t;

    f = fun_norma(r_ab);
    g = fun_norma(r_ab - v_b * dt * e_b);
    b = 0.5 * ( (f + g)^2 - (v_b * dt)^2 )^(1/2);

    grad_b = ( (f + g)^2 * r_ab - f * (f + g) * v_b * dt * e_b ) /
              (4.0 * b * f * g);
    V_ab = par.V_0 * exp(-b/par.sigma);
    der_V_ab = -V_ab/par.sigma;

    return -der_V_ab * grad_b;
end
function fun_Sfab(Mult::Vector{Peaton},
                 par::ParPeaton,
                 pe::ParEnt,
                 i::Int64)

    h = [0.0, 0.0];
    N = length(Mult);

    for j in 1:N
        if((i != j) && fun_norma(Mult[i].pos - Mult[j].pos) < pe.cc )
            h += fun_fab(Mult[i], Mult[j], par, pe);
        end
    end

    return h;
end

# Interacción con las fronteras
function fun_FaB(x::Peaton,
                par::ParPeaton,
                y::Vector{Float64})

    r_aB = x.pos - y;
    nr_aB = fun_norma(r_aB);
    U_nraB = par.U_0 * exp(-nr_aB/par.R);
    der_U_nraB = -U_nraB/par.R;

    return -der_U_nraB * r_aB / nr_aB;
end

```

```

end
function fun_SFaB(x::Peaton,
    par::ParPeaton,
    pe::ParEnt)

    if (pe.orientacion == "horizontal")
        return fun_FaB(x, par, [x.pos[1], 0.0]) +
            fun_FaB(x, par, [x.pos[1], pe.anchos]);

    elseif (pe.orientacion == "vertical")
        return fun_FaB(x, par, [0.0, x.pos[2]]) +
            fun_FaB(x, par, [pe.largo, x.pos[2]]);
    else
        return [0.0, 0.0];
    end
end

# Funciones generales
function fun_euler(x::Peaton,
    F::Vector{Float64},
    h::Float64)

    x.pos = x.pos + h * x.vel;
    x.vel = x.vel + h * F;

    return x
end

function fun_g(x::Float64, y::Float64)
    if x > y
        return 1.0
    else
        return x/y
    end
end

function fun_condiciones(x::Peaton,
    pe::ParEnt)

    # Evaluacion de la funcion g
    h_2 = fun_g(x.v_max, fun_norma(x.vel)); #constante auxiliar
    x.vel = h_2 * x.vel;                    # velocidad corregida

    if (x.pos[1] < 0.0)
        x.pos[1] = x.pos[1] + pe.largo;
    elseif (x.pos[1] > pe.largo)
        x.pos[1] = x.pos[1] - pe.largo;
    end

    if (x.pos[2] < 0.0)

```

```

        x.pos[2] = x.pos[2] + pe.ancho;
    elseif (x.pos[2] > pe.ancho)
        x.pos[2] = x.pos[2] - pe.ancho;
    end

    return x
end
function fun_Posicion_sig(Mult::Vector{Peaton},
    par::ParPeaton,
    pe::ParEnt,
    h::Float64)

    N = length(Mult);
    copia = Array{Peaton}(undef, N);

    for i in 1:N

        A = fun_Fa(Mult[i], par, pe);
        B = fun_Sfab(Mult, par, pe, i);
        C = fun_SFAB(Mult[i], par, pe);

        F = A + B + C;

        # Función de Euler
        copia[i] = fun_euler(Mult[i], F, h);
        # Condiciones de frontera y velocidad maxima
        copia[i] = fun_condiciones(copia[i], pe);

    end

    return copia
end

function fun_Matrix_Pos(Mult::Vector{Peaton},
    par::ParPeaton,
    pe::ParEnt,
    h::Float64,
    num_h::Int64)

    N = length(Mult);
    M = zeros(Float64, num_h + 1, 4N);

    for l in 1:num_h
        for i in 1:N
            M[l, i] = Mult[i].pos[1];
            M[l, i + N] = Mult[i].pos[2];
            M[l, i + 2N] = Mult[i].vel[1];
            M[l, i + 3N] = Mult[i].vel[2];
        end
    end
end

```

```

end

Mult = fun_Posicion_sig(Mult, par, pe, h);
end

l = num_h + 1;
for i in 1:N
    M[l, i] = Mult[i].pos[1];
    M[l, i + N] = Mult[i].pos[2];
    M[l, i + 2N] = Mult[i].vel[1];
    M[l, i + 3N] = Mult[i].vel[2];
end

return M
end

```

```
end
```

En el próximo programa se escriben las variables necesarias para realizar la simulación de los peatones, tales como, largo y ancho de la calle, velocidad inicial de cada peatón, número de peatones que se mueven a la izquierda y a la derecha, parámetros de las ecuaciones descritas por Helbing y Molnár, etc. El programa escribe como resultado en un archivo csv(`datos.csv`) una matriz

$$M = \begin{pmatrix} x_1 \dots x_N & y_1 \dots y_N & v_{x_1} \dots v_{x_N} & v_{y_1} \dots v_{y_N} \\ \vdots & \vdots & \vdots & \vdots \\ x_1 \dots x_N & y_1 \dots y_N & v_{x_1} \dots v_{x_N} & v_{y_1} \dots v_{y_N} \end{pmatrix}$$

donde cada fila representa un intervalo temporal. También crea una animación gif de las posiciones de todos los peatones, según los intervalos de filas que se den como entrada de la matriz  $M$ .

```

#=#
# Para correr el programa se necesita una
# estructura de directorios
OscPeatonal/Calle/camino-h

# Además de los archivos
# Definición de las funciones
OscPeatonal/Calle/MFScalle.jl
# Dirección de éste archivo
OscPeatonal/Calle/Ejec-calle-h.jl

#####
# Modelo de Fuerza Social por
# Helbing y Molnár
#####

```

Autor: René Mora Maya

Paquetes a utilizar

```

Plots-----para gráficos
LaTeXStrings--para insertar texto tex en gráficos
Random-----funciones con variables aleatorias
Distributions--distribuciones de probabilidad
LinearAlgebra--operaciones de algebra lineal
FTW-----series de Fourier
=#
using Plots, LaTeXStrings, Random, Distributions

include("OscPeatonal/Calle/MFScale.jl");

#####
##### Parámetros del entorno #####
#####
c_a = 50.0; # Largo de la calle
c_b = 20.0; # Ancho de la calle
orientacion = "horizontal";
cc = 9.0;# Distancia de interacción entre peatones
pe = MFS.ParEnt(c_a, c_b, orientacion, cc);
#####
##### Parámetros de los peatones #####
#####
# Número total de peatones
# Se necesita una densidad de 0.3/m^2
rho = 0.3;
peats = c_a * c_b * rho;
if (peats)%2 == 0.0
    # Peatones a la izquierda
    N_pi = Int64((peats)/2);
    # Peatones a la derecha
    N_pd = N_pi;
else
    N_pi = Int64(ceil(peats/2.0));
    N_pd = Int64(floor(peats/2.0));
end
v_0 = 1.0; #*****
v_max = 1.3*v_0; # Magnitud de vel. máxima
# Valores para los intervalos
l1 = 0.125;
l = (1.0/2.0)^4;
l2 = 1.125
Multitud = [MFS.Peaton(
    [rand(l1:l:pe.largo-l2) + rand(), # Posición
     rand(l1:l:pe.ancho-l2) + rand()],
    [0.0, 0.0], # Velocidad
    [pe.largo + 1.5, rand(l1:l:pe.ancho)],# Punto de destino
    v_0, # Magnitud de vel. deseada
    v_max) # Magnitud de vel. máxima

```

```

    for i in 1:N_pi];
Multitud_2 = [MFS.Peaton(
    [rand(1:1:pe.largo-12) + rand(), # Posición
    rand(1:1:pe.ancho-12) + rand()],
    [0.0, 0.0],
    [-1.5, rand(1:1:pe.ancho)],
    0.0,
    v_max)
    for i in 1:N_pd];
append!(Multitud, Multitud_2);
#####
##### Parámetros de las funciones #####
#####
tau = 0.5; # Tiempo de relajación *****
delta_t = (1.0/2.0)^4; # Tiempo asociado al paso de beta
V_0 = 2.1; # Inter. entre peatones
sigma = 0.3; # Inter. entre peatones
U_0 = 10.0; # Inter. entre fronteras
R = 0.2; # Inter. entre fronteras
par = MFS.ParPeaton(tau, delta_t, V_0, sigma, U_0, R);
#####
### Ejecución de la matriz M #####
#####
# Lapso temporal para la ecuación de Euler
h = delta_t;
# Número de lapsos temporales
num_h = 4000;
M = MFS.fun_Matrix_Pos(Multitud, par, pe, h, num_h);
MFS.fun_arch_datos(M,
"OscPeatonal/Calle/camino-h/datos.csv");
#####
##### Parámetros de la animación #####
#####
graf_a = [3000, 5, 4000];
graf_colores = [:red, :blue, :black];
graf_puntos = [4, 4];
graf_tam = [600, 300];
graf_int_ejes = [(-1.5, pe.largo+1.5),
(-0.5, pe.ancho+0.5)];
graf_fps = 15;
graf_arch =
"OscPeatonal/Calle/camino-h/datos-$(graf_fps)fps";
graf_imag =
"OscPeatonal/Calle/camino-h/im-datos/";
graf = MFS.DatosGif(graf_a,
graf_colores,
graf_puntos,
graf_tam,

```

```
graf_int_ejes,  
graf_arch,  
graf_fps,  
graf_imag);
```

```
MFS.fun_animacion_color(M, N_pi, N_pd, graf, pe)
```

La forma general de operar del programa es la siguiente: se colocan de forma aleatoria  $N$  peatones y a cada intervalo de tiempo  $h$  se guarda los valores de sus posiciones y velocidades. El calculo se realiza resolviendo las ecuaciones diferenciales del capítulo 1.3 por el método de Euler.

## Programa en Julia (Peatones cruzando una puerta).

A continuación se muestra un programa creado con Julia[16], llamado `MFSpuerta.jl`, donde se implementan las ecuaciones necesarias para representar el movimiento de los peatones que atraviesan una puerta en ambas direcciones, y ésta se encuentra a la mitad de una calle de largo y ancho definidos. El programa se basa en el modelo descrito por Helbing y Molnar[2].

En la primera parte del programa se implementan conjuntos de datos `Peaton`, `ParPeaton`, `ParEnt`, `DatosAlpha`, `DatosGenerales`, `DatosGif`, que determinan como se van a agrupar los valores necesarios correspondientes al inicio del experimento, es decir, las variables involucradas para cada peatón.

La segunda parte consiste en diseñar las ecuaciones del modelo de Helbing y Molnár y como van a operar con los conjuntos descritos en el párrafo anterior. También se crean las funciones para graficar y crear gifs.

Por último se desarrolla una función general `fun_Matrix_Pos` que utiliza todas las funciones de la segunda parte y crea una matriz con las posiciones y velocidades de todos los peatones para cada intervalo de tiempo. Y finalmente se guarda esta matriz como un archivo de datos con extensión CSV.

```
#####
# Funciones Modelo de Fuerza Social
#####

#=
En este módulo se desarrollan las fórmulas para el Modelo de Fuerza Social.

René Mora Maya

Lunes 03-02-23
=#
# La siguiente instrucción sirve para *precompilar* el módulo
__precompile__(true)

module MFS
    import Base: +, -, *, /, ^, ==, &, |
    import Plots: @animate
    using Plots, DataFrames, CSV

    mutable struct Peaton
        pos::Vector{Float64} # Posición
```

```

    vel::Vector{Float64} # Velocidad
    dir::Vector{Float64} # Punto de destino
    v_0::Float64         # Magnitud de vel. deseada
    v_max::Float64      # Magnitud de vel. máxima
end
mutable struct ParPeaton
    tau::Float64        # Tiempo de relajación
    delta_t::Float64    # Tiempo asociado al paso de beta
    V_0::Float64        # Inter. entre peatones
    sigma::Float64      # Inter. entre peatones
    U_0::Float64        # Inter. entre fronteras
    R::Float64          # Inter. entre fronteras
end
mutable struct ParEnt
    largo::Float64      # Largo de la calle
    ancho::Float64      # Ancho de la calle
    c::Float64          # Distancia a la que reacciona el peatón con la frontera
    puerta::Vector{Float64} # Mitad del ancho de la puerta
    lpuerta::Float64    # Longitud de la puerta
    cc::Float64         # Distancia de interacción entre peatones
end
mutable struct DatosAlpha
    # Tiempo de relajación
    tau::Float64
    # Magnitud de vel. deseada
    v_0::Float64
    # Longitud de la puerta
    lpuerta::Float64
end
mutable struct DatosGenerales
    # Largo de la calle
    calle_a::Float64
    # Ancho de la calle
    calle_b::Float64
    # Distancia interacción peatón y frontera
    dfront::Float64
    # Localización de la puerta
    ppuerta::Vector{Float64}
    # Distancia de interacción entre peatones
    dpeat::Float64
    # Peatones a la izquierda
    Npi::Int64
    # Peatones a la derecha
    Npd::Int64
    # Tiempo asociado al paso de beta
    delta_t::Float64
    # Interacción entre peatones
    V_0::Float64

```

```

sigma::Float64
# Interacción entre fronteras
U_0::Float64
R::Float64
# Lapso temporal para la ecuación de Euler
h::Float64
# Número de lapsos temporales
num_h::Int64
end
mutable struct DatosGif
# Intervalos del gif (a[1]=inicio, a[2]=paso, a[3]=fin)
a::Vector{Int64}
# Colores de las gráficas (:red,:blue)
colores::Vector{Symbol}
# Tamaño de los puntos graficados ([3,4])
puntos::Vector{Int64}
# Tamaño de la gráfica ([500, 500])
tam::Vector{Int64}
# Intervalo de los ejes ([[0.0, 50.0), (0.0, 10.0]])
int_ejes::Vector{Tuple{Float64, Float64}}
# Archivo con terminación "datos.gif"
arch::String
# Número de imágenes por segundo
fps::Int64
# Directorio con imágenes
imag::String
end
function fun_arch_datos(M::Matrix{Float64},
    arch_nombre::String)

    ARC = open( arch_nombre,"w" );

    for i in 1:length(M[1:end, 1])
        for j in 1:length(M[1, 1:end])
            write(ARC,"$(M[i, j]) ")
        end
        write(ARC,"\n")
    end
    close(ARC)
end
function fun_trayectoria(M::Matrix{Float64},
    Ni::Int64,
    Nd::Int64,
    graf::MFS.DatosGif,
    j::Int64) # Peatón j

    scatter(M[1:end, j], M[1:end, Ni + Nd + j], color = graf.colores[1],
        legend =:false, marker = graf.puntos[1], size = graf.tam)

```

```

xlims!(graf.int_ejes[1])
ylims!(graf.int_ejes[2])
end
function fun_animacion_color(M::Matrix{Float64},
    Ni::Int64,          # Peatones del lado izquierdo
    Nd::Int64,          # Peatones del lado derecho
    graf::MFS.DatosGif,
    pe::MFS.ParEnt)

    N = Ni + Nd;
    anim = @animate for i in graf.a[1]:graf.a[2]:graf.a[3]
        scatter(M[i, 1:Ni], M[i, N+1:N+Ni], color = graf.colores[1],
            legend =:false, marker = graf.puntos[1], size = graf.tam)
        scatter!(M[i, Ni + 1:N], M[i, N+Ni+1:2N ], color = graf.colores[2],
            legend =:false, marker = graf.puntos[2])
        plot!([pe.puerta[1],pe.puerta[1]],
            [0.0, pe.puerta[2] - pe.lpuerta/2.0],
            color = graf.colores[3], legend =:false, line = 2)
        plot!([pe.puerta[1],pe.puerta[1]],
            [pe.puerta[2] + pe.lpuerta/2.0, pe.ancho],
            color = graf.colores[3], legend =:false, line = 2)
        xlims!(graf.int_ejes[1])
        ylims!(graf.int_ejes[2])
        savefig(graf.imag*"i")
    end

    return gif(anim, graf.arch*".gif", fps = graf.fps)
end
function fun_animacion_color_inv(M::Matrix{Float64},
    Ni::Int64,          # Peatones del lado izquierdo
    Nd::Int64,          # Peatones del lado derecho
    graf::MFS.DatosGif,
    pe::MFS.ParEnt)

    N = Ni + Nd;
    anim = @animate for i in graf.a[1]:graf.a[2]:graf.a[3]
        scatter(M[i, N+1:N+Ni], M[i, 1:Ni], color = graf.colores[1],
            legend =:false, marker = graf.puntos[1], size = graf.tam)
        scatter!(M[i, N+Ni+1:2N], M[i, Ni + 1:N], color = graf.colores[2],
            legend =:false, marker = graf.puntos[2])
        plot!([0.0, pe.puerta[2] - pe.lpuerta/2.0],
            [pe.puerta[1],pe.puerta[1]],
            color = graf.colores[3], legend =:false, line = 2)
        plot!([pe.puerta[2] + pe.lpuerta/2.0, pe.ancho],
            [pe.puerta[1],pe.puerta[1]],
            color = graf.colores[3], legend =:false, line = 2)
        ylims!(graf.int_ejes[1])
        xlims!(graf.int_ejes[2])
    end

```

```

        savefig(graf.imag*"i")
    end

    return gif(anim, graf.arch*".gif", fps = graf.fps)
end
##### DataFrames #####
function fun_trayectoria(M::Matrix{Union{Missing, Float64}},
    Ni::Int64,
    Nd::Int64,
    graf::MFS.DatosGif,
    j::Int64) # Peatón j

    scatter(M[1:end, j], M[1:end, Ni + Nd + j], color = graf.colores[1],
        legend =:false, marker = graf.puntos[1], size = graf.tam)
    xlims!(graf.int_ejes[1])
    ylims!(graf.int_ejes[2])
end
function fun_animacion_color(M::Matrix{Union{Missing, Float64}},
    Ni::Int64,          # Peatones del lado izquierdo
    Nd::Int64,          # Peatones del lado derecho
    graf::MFS.DatosGif,
    pe::MFS.ParEnt)

    N = Ni + Nd;
    anim = @animate for i in graf.a[1]:graf.a[2]:graf.a[3]
        scatter(M[i, 1:Ni], M[i, N+1:N+Ni], color = graf.colores[1],
            legend =:false, marker = graf.puntos[1], size = graf.tam)
        scatter!(M[i, Ni + 1:N], M[i, N+Ni+1:2N ], color = graf.colores[2],
            legend =:false, marker = graf.puntos[2])
        plot!([pe.puerta[1],pe.puerta[1]],
            [0.0, pe.puerta[2] - pe.lpuerta/2.0],
            color = graf.colores[3], legend =:false, line = 2)
        plot!([pe.puerta[1],pe.puerta[1]],
            [pe.puerta[2] + pe.lpuerta/2.0, pe.ancho],
            color = graf.colores[3], legend =:false, line = 2)
        xlims!(graf.int_ejes[1])
        ylims!(graf.int_ejes[2])
        savefig(graf.imag*"i")
    end

    return gif(anim, graf.arch*".gif", fps = graf.fps)
end
function fun_animacion_color_inv(M::Matrix{Union{Missing, Float64}},
    Ni::Int64,          # Peatones del lado izquierdo
    Nd::Int64,          # Peatones del lado derecho
    graf::MFS.DatosGif,
    pe::MFS.ParEnt)

```

```

N = Ni + Nd;
anim = @animate for i in graf.a[1]:graf.a[2]:graf.a[3]
    scatter(M[i, N+1:N+Ni], M[i, 1:Ni], color = graf.colores[1],
            legend =:false, marker = graf.puntos[1], size = graf.tam)
    scatter!(M[i, N+Ni+1:2N], M[i, Ni + 1:N], color = graf.colores[2],
            legend =:false, marker = graf.puntos[2])
    plot!([0.0, pe.puerta[2] - pe.lpuerta/2.0],
          [pe.puerta[1],pe.puerta[1]],
          color = graf.colores[3], legend =:false, line = 2)
    plot!([pe.puerta[2] + pe.lpuerta/2.0, pe.anch],
          [pe.puerta[1],pe.puerta[1]],
          color = graf.colores[3], legend =:false, line = 2)
    ylims!(graf.int_ejes[1])
    xlims!(graf.int_ejes[2])
    savefig(graf.imag*"i")
end

return gif(anim, graf.arch*".gif", fps = graf.fps)
end

fun_norma(x::Vector{Float64}) = (x[1]^2 + x[2]^2)^(1/2);

# Fuerza de aceleración
function fun_e_a(x::Peaton,
                pe::ParEnt)

    if ( fun_norma(pe.puerta - x.pos) <= pe.lpuerta/2.0 ||
        abs(x.dir[1] - x.pos[1]) < abs(x.dir[1] - pe.puerta[1]) )
        D = x.dir;
    else
        D = pe.puerta;
    end

    return (D - x.pos)/fun_norma(D - x.pos);
end

fun_Fa(x::Peaton, y::ParPeaton, pe::ParEnt) =
(x.v_0 * fun_e_a(x, pe) - x.vel)/y.tau;

# Interacción con los peatones
function fun_fab(x::Peaton,
                y::Peaton,
                par::ParPeaton,
                pe::ParEnt)

    r_ab = x.pos - y.pos;
    v_b = fun_norma(y.vel);
    e_b = fun_e_a(y, pe);
    dt = par.delta_t;

```

```

f = fun_norma(r_ab);
g = fun_norma(r_ab - v_b * dt * e_b);
b = 0.5 * ( (f + g)^2 - (v_b * dt)^2 )^(1/2);

grad_b = ( (f + g)^2 * r_ab - f * (f + g) * v_b * dt * e_b ) /
          (4.0 * b * f * g);
V_ab = par.V_0 * exp(-b/par.sigma);
der_V_ab = -V_ab/par.sigma;

return -der_V_ab * grad_b;
end
function fun_Sfab(Mult::Vector{Peaton},
                  par::ParPeaton,
                  pe::ParEnt,
                  i::Int64)

h = [0.0, 0.0];
N = length(Mult);

for j in 1:N
    if((i != j) && fun_norma(Mult[i].pos - Mult[j].pos) < pe.cc )
        h += fun_fab(Mult[i], Mult[j], par, pe);
    end
end

return h;
end

# Interacción con las fronteras
function fun_FaB(x::Peaton,
                 par::ParPeaton,
                 y::Vector{Float64})

r_aB = x.pos - y;
nr_aB = fun_norma(r_aB);
U_nraB = par.U_0 * exp(-nr_aB/par.R);
der_U_nraB = -U_nraB/par.R;

return -der_U_nraB * r_aB / nr_aB;

end
function fun_SFaB(x::Peaton,
                  par::ParPeaton,
                  pe::ParEnt)

esq_s = pe.puerta[2] + pe.lpuerta/2.0;
esq_i = pe.puerta[2] - pe.lpuerta/2.0;

```

```

if ( (abs(pe.puerta[1] - x.pos[1]) <= pe.c) &&
      (x.pos[2] <= esq_i || x.pos[2] >= esq_s) )

    return fun_FaB(x, par, [pe.puerta[1], x.pos[2]]);
elseif ( (abs(pe.puerta[1] - x.pos[1]) < pe.lpuerta) &&
          (esq_i < x.pos[2] < esq_s) )
    return 0.25*(
        fun_FaB(x, par, [pe.puerta[1], esq_s]) +
        fun_FaB(x, par, [pe.puerta[1], esq_i]) );
else
    return [0.0, 0.0];
end
end
end

# Funciones generales
function fun_euler(x::Peaton,
                  F::Vector{Float64},
                  h::Float64)

    x.pos = x.pos + h * x.vel;
    x.vel = x.vel + h * F;

    return x
end

function fun_g(x::Float64, y::Float64)
    if x > y
        return 1.0
    else
        return x/y
    end
end

function fun_condiciones(x::Peaton,
                        pe::ParEnt)

    # Evaluacion de la funcion g
    h_2 = fun_g(x.v_max, fun_norma(x.vel)); #constante auxiliar
    x.vel = h_2 * x.vel;                    # velocidad corregida

    if (x.pos[1] < 0.0)
        x.pos[1] = x.pos[1] + pe.largo;
        x.pos[2] = rand((1.0/2.0)^3:(1.0/2.0)^4:pe.anch-1.0) + rand();
    elseif (x.pos[1] > pe.largo)
        x.pos[1] = x.pos[1] - pe.largo;
        x.pos[2] = rand((1.0/2.0)^3:(1.0/2.0)^4:pe.anch-1.0) + rand();
    end

    return x
end
end

```

```

function fun_Posicion_sig(Mult::Vector{Peaton},
    par::ParPeaton,
    pe::ParEnt,
    h::Float64)

    N = length(Mult);
    copia = Array{Peaton}(undef, N);

    for i in 1:N

        A = fun_Fa(Mult[i], par, pe);
        B = fun_Sfab(Mult, par, pe, i);
        C = fun_SFAB(Mult[i], par, pe);

        F = A + B + C;

        # Función de Euler
        copia[i] = fun_euler(Mult[i], F, h);
        # Condiciones de frontera y velocidad maxima
        copia[i] = fun_condiciones(copia[i], pe);

    end

    return copia
end

function fun_Matrix_Pos(Mult::Vector{Peaton},
    par::ParPeaton,
    pe::ParEnt,
    h::Float64,
    num_h::Int64)

    N = length(Mult);
    M = zeros(Float64, num_h + 1, 4N);

    for l in 1:num_h
        for i in 1:N
            M[l, i] = Mult[i].pos[1];
            M[l, i + N] = Mult[i].pos[2];
            M[l, i + 2N] = Mult[i].vel[1];
            M[l, i + 3N] = Mult[i].vel[2];
        end

        Mult = fun_Posicion_sig(Mult, par, pe, h);
    end

    l = num_h + 1;
    for i in 1:N

```

```

    M[1, i] = Mult[i].pos[1];
    M[1, i + N] = Mult[i].pos[2];
    M[1, i + 2N] = Mult[i].vel[1];
    M[1, i + 3N] = Mult[i].vel[2];
end

    return M
end
function fun_flujo(M::Matrix{Float64},
    pe::ParEnt,
    N_pi::Int64,
    N_pd::Int64,
    num_h::Int64,
    div::Int64)

    li = pe.puerta[1] + pe.lpuerta;
    ld = pe.puerta[1] - pe.lpuerta;
    k1 = Float64[];
    k2 = Float64[];

    for i in 1:Int64(floor(num_h/div))
        ki = 0.0;
        for j in 1:N_pi
            if( (M[(i-1)*div+1, j] < li) &&
                (M[i*div, j] > li)
            )
                ki += 1.0;
            else
                end
        end
        end
        append!(k1, ki/div)

        kd = 0.0;
        for j in 1:N_pd
            if ( (M[(i-1)*div+1, N_pi + j] > ld) &&
                (M[i*div, N_pi + j] < ld)
            )
                kd += 1.0;
            else
                end
        end
        end
        append!(k2, kd/div)
    end

    return k1, k2;
end
function fun_flujo(M::Matrix{Union{Missing, Float64}},
    pe::ParEnt,
```

```

N_pi::Int64,
N_pd::Int64,
num_h::Int64,
div::Int64)

li = pe.puerta[1] + pe.lpuerta;
ld = pe.puerta[1] - pe.lpuerta;
k1 = Float64[];
k2 = Float64[];

for i in 1:Int64(floor(num_h/div))
    ki = 0.0;
    for j in 1:N_pi
        if( (M[(i-1)*div+1, j] < li) &&
            (M[i*div, j] > li)
            )
            ki += 1.0;
        else
            end
        end
    end
    append!(k1, ki/div)

    kd = 0.0;
    for j in 1:N_pd
        if ( (M[(i-1)*div+1, N_pi + j] > ld) &&
            (M[i*div, N_pi + j] < ld)
            )
            kd += 1.0;
        else
            end
        end
    end
    append!(k2, kd/div)
end

return k1, k2;
end
function fun_Corr(x::Vector{Float64},
    y::Vector{Float64},
    m::Int64)

k = Float64[];
T = length(x);
for j in 1:m
    m_2 = T - j;
    h = 0.0;
    for i in 1:m_2
        h += x[i] * y[i + j]
    end
end

```

```

        append!(k, h);
    end

    return k;
end

end

```

En el próximo programa se escriben las variables necesarias para realizar la simulación de los peatones cruzando a través de una puerta, tales como, largo y ancho de la calle, tamaño de la puerta, velocidad inicial de cada peatón, número de peatones que se mueven a la izquierda y a la derecha, parámetros de las ecuaciones descritas por Helbing y Molnár, etc. El programa escribe como resultado un archivo (`datos.csv`) que contiene la información de todos los peatones con la forma de la matriz mostrada a continuación,

$$M = \begin{pmatrix} x_1 \dots x_N & y_1 \dots y_N & v_{x_1} \dots v_{x_N} & v_{y_1} \dots v_{y_N} \\ \vdots & \vdots & \vdots & \vdots \\ x_1 \dots x_N & y_1 \dots y_N & v_{x_1} \dots v_{x_N} & v_{y_1} \dots v_{y_N} \end{pmatrix}$$

donde cada fila representa un intervalo temporal.  $x_N, y_N$  son las coordenadas del peatón  $N$  y  $v_{x_N}, v_{y_N}$  las coordenadas de su velocidad. El programa también produce como salida una animación gif y una gráfica que muestra el flujo, la auto-correlación del flujo y la frecuencia para el movimiento de los peatones a través de la puerta.

`#=`

Para correr el programa se necesita una estructura de directorios

```

# Definición de las funciones
OscPeatonal/Puerta/MFScalle.jl

```

```

# Dirección de éste archivo
OscPeatonal/Puerta/Ejec-gif.jl

```

```

# Directorio para guardar los datos csv,
graficas y animación gif
OscPeatonal/Puerta/puertax

```

```

# Directorio para guardar las imágenes,
que forman la animación gif
OscPeatonal/Puerta/puertax/im-datos

```

Tesis

Paquetes a utilizar

Plots-----para gráficos

LaTeXStrings--para insertar texto tex en gráficos

Random-----funciones con variables aleatorias

Distributions--distribuciones de probabilidad

```

LinearAlgebra--operaciones de algebra lineal
SpecialFunctions: gamma ---funciones especiales
Polynomials.PolyCompat ---ajustes polinomiales a partir de datos
TaylorIntegration-----representación por el método de Taylor
FTW-----series de Fourier
=#
using Plots, LaTeXStrings, Random, Distributions, FFTW

include("OscPeatonal/Puerta/MFSpuerta.jl")

DA = MFS.DatosAlpha(
    0.5, # Tiempo de relajación
    1.0, # Magnitud de vel. deseada
    0.5 # Longitud de la puerta
);

DG = MFS.DatosGenerales(
    20.0, # Largo de la calle
    20.0, # Ancho de la calle
    2.0, # Distancia interacción peatón y frontera
    [10.0, 10.0], # Localización de la puerta
    # Distancia de interacción entre peatones
    9.0,
    125, # Peatones a la izquierda
    125, # Peatones a la derecha
    (1.0/2.0)^4, # Tiempo asociado al paso de beta
    # Interacción entre peatones
    2.1, # V_0
    0.3, # sigma
    # Interacción entre fronteras
    10.0, # U_0
    0.2, # R
    (1.0/2.0)^4, # h
    100 # num_h
);

#####
##### Parámetros del entorno #####
#####
pe = MFS.ParEnt(DG.calle_a,
    DG.calle_b,
    DG.dfront,
    DG.ppuerta,
    DA.lpuerta,
    DG.dpeat);
#####
##### Parámetros de los peatones #####
#####

```

```

par = MFS.ParPeaton(DA.tau,
  DG.delta_t,
  DG.V_0,
  DG.sigma,
  DG.U_0,
  DG.R);

v_max = 1.3*DA.v_0; # Magnitud de vel. máxima
# Valores para los intervalos
l1 = 0.125;
l = (1.0/2.0)^4;
l2 = 1.125
Multitud = [MFS.Peaton(
  [rand(l1:l:pe.puerta[1]-l2)+ rand(), # Posición
   rand(l1:l:pe.ancho-l2) + rand()],
  [0.0, 0.0], # Velocidad
  [pe.largo + 1.5, pe.ancho/2.0], # Punto de destino
  DA.v_0, # Magnitud de vel. deseada
  v_max) # Magnitud de vel. máxima
  for i in 1:DG.Npi];
Multitud_2 = [MFS.Peaton(
  [rand(pe.puerta[1]+0.5:l:pe.largo-l2)+ rand(),
   rand(l1:l:pe.ancho-l2) + rand()],
  [0.0, 0.0],
  [-1.5, pe.ancho/2.0],
  DA.v_0,
  v_max)
  for i in 1:DG.Npd];
append!(Multitud, Multitud_2);

M = MFS.fun_Matrix_Pos(Multitud, par, pe, DG.h, DG.num_h);
MFS.fun_arch_datos(M,
  "OscPeatonal/Puerta/puertax/datos.csv");

#####
##### Parámetros de la animación #####
#####
graf_a = [1, 1, 100];
graf_colores = [:skyblue, :orange, :black];
graf_puntos = [4, 4];
graf_tam = [500,500];
graf_int_ejes = [(5.0, 15.0), (5.0, 15.0)];
graf_fps = 15;
graf_arch =
  "OscPeatonal/Puerta/puertax/datos-$(graf_fps)fps";
graf_imag = "OscPeatonal/Puerta/puertax/im-datos/";
graf = MFS.DatosGif(
  graf_a,

```

```

graf_colores,
graf_puntos,
graf_tam,
graf_int_ejes,
graf_arch,
graf_fps,
graf_imag);

MFS.fun_animacion_color_inv(M, DG.Npi, DG.Npd, graf, pe);

#####
##### Datos de las gráficas #####
#####
div = 2;
# Vectores de los flujos a izquierda y derecha
flujo_i, flujo_d =
MFS.fun_flujo(M, pe, DG.Npi, DG.Npd, DG.num_h, div);

flujo = flujo_d - flujo_i;
m = 50;
Cflujo = MFS.fun_Corr(flujo, flujo, m);
Corr = Cflujo;
Tau = 1.0*[i for i in 1:m];
# División del lapsos temporales de una unidad de segundo
lt = 1.0/(DG.h*div);

# Transformada de Fourier
F = abs.(rfft(Corr));
freqs = rfftfreq(length(Tau), lt);
# valor máximo e índice de F
mm, ind = findmax(F[3:end]);

Flujo = plot(flujo, title = "Flujo", color=:blue, label=false);
tiempo = plot(Tau, Corr, title = "Correlación", legend=false);
frecuencia = plot(freqs, F, title = "Frecuencia", color=:red,
    label="$ (abs(freqs[ind+2])) Hz", legend=:right)
plot!([freqs[ind+2], freqs[ind+2]], [0.0, 500.0], line=0.8,
    color=:black, label=false);
plot!([freqs[ind+2], freqs[ind+2]], [0.0, 500.0], line=0.8,
    color=:black, label=false)

plot(Flujo, tiempo, frecuencia, layout = (3,1))
savefig("OscPeatonal/Puerta/puertax/analisis-final")

```

La forma general de operar del programa es la siguiente: se colocan de forma aleatoria  $N_i$  peatones del lado izquierdo de la puerta y  $N_d$  peatones del lado derecho, a cada intervalo de tiempo  $h$  se guardan los valores de sus posiciones y velocidades. Los peatones tratan de moverse hacia la puerta y después continúan hacia el final

de la calle. El cálculo se realiza resolviendo las ecuaciones diferenciales del capítulo 1.3 por el método de Euler.

## Apéndice E

### **Peatones con movimiento ortogonal.**

Se tiene el experimento de la intersección de dos calles perpendiculares (figura E.1), donde los peatones rojos se mueven hacia arriba y los azules se mueven a la derecha, se puede observar como se forman líneas de peatones inclinadas por los que el movimiento es el mismo que el que se presenta en una calle horizontal aplicando una rotación de los ejes.

Una situación similar se obtendría en el caso hidrodinámico si los fluidos de diferente densidad están sometidos a fuerzas de cuerpo en direcciones diferentes. No se profundizó en este problema en la realización de la Tesis. Dejaremos este problema para trabajo a futuro.

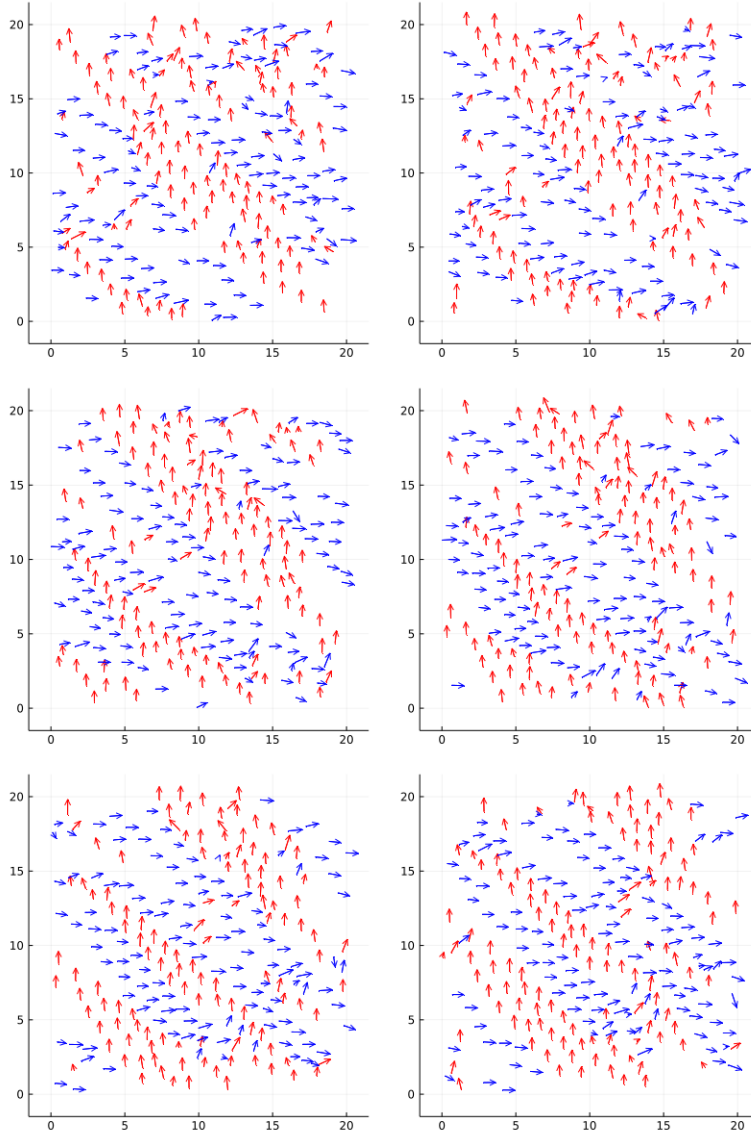


FIGURA E.1. Posiciones de los peatones para diferentes intervalos temporales de izquierda a derecha, de arriba a abajo,  $t = 3350(1/2)^4 s$ ,  $t = 3400(1/2)^4 s$ ,  $t = 3450(1/2)^4 s$ ,  $t = 3500(1/2)^4 s$ ,  $t = 3550(1/2)^4 s$  y  $t = 3600(1/2)^4 s$ . El ancho de la calle corresponde a  $c_b = 20.0$ , el largo a  $c_a = 20.0$  y los parámetros de la tabla 4.1. Los peatones rojos se mueven hacia arriba y los turquesa hacia la derecha. Para esta figura se tiene una densidad de peatones de  $0.6$  peatones/ $m^2$ . Todos los programas tienen condiciones periódicas a la frontera, tanto en el eje vertical como el horizontal.