

# MASTER UNIVERSITARIO EN FÍSICA DE SISTEMAS COMPLEJOS

---

## ESTUDIO DE UN FLUJO INCOMPRESIBLE Y VISCOSO ALREDEDOR DE UN OBSTÁCULO: RÉGIMEN ESTACIONARIO

---

Trabajo Fin de Máster

**Alumno: Jaime Rey Vidaurrázaga**

**Tutora: Mar Serrano Maestro**

Curso 2013-2014

Universidad Nacional de Educación a Distancia

“Estudio de un flujo incompresible y viscoso alrededor de un obstáculo: Régimen estacionario”

VºBº Mar Serrano Maestro

En Madrid a 12 de septiembre de 2014

## Resumen

Se describe el comportamiento estacionario de un fluido viscoso incompresible en cuyo camino se interpone un obstáculo. Para ello, se plantean las ecuaciones hidrodinámicas no lineales que describen el problema. Estas ecuaciones en derivadas parciales son elípticas y, debido a su complejidad, deben resolverse mediante el uso de métodos numéricos. Se investiga el efecto producido por un obstáculo ortoédrico de longitud infinita sobre un flujo laminar, lo que permite tratar al problema en dos dimensiones. El algoritmo de resolución desarrollado se fundamenta en la utilización de diferencias finitas para la discretización de las ecuaciones del problema, que se estudian en la representación vorticidad-función de corriente. El análisis de los resultados obtenidos permiten afirmar que el método de diferencias finitas empleado reproduce adecuadamente el comportamiento de un flujo laminar estacionario alrededor de un obstáculo ortoédrico para números de Reynolds bajos e inferiores al número de Reynolds crítico.

## Abstract

In this paper we describe the steady state of an incompressible viscous flow past an obstacle. To do this, the nonlinear hydrodynamic equations describing the problem are set out. These elliptic partial differential equations, due to their complexity, must be solved through the use of numerical computation techniques. These equations are studied in the vorticity-stream function representation. We consider the effect of an orthohedral obstacle of infinite length on a laminar flow, which can be studied as a two-dimensional problem. The developed method is based on the use of finite differences for the discretization of the problem's equations. The analysis of the results confirms that the finite differences method used properly reproduces the behavior of a steady state laminar flow around an orthohedral obstacle for low Reynolds numbers below the critical Reynolds number.

## Índice

Resumen.....	2
Abstract.....	2
1. Introducción.....	4
2. Ecuaciones de la dinámica de fluidos.....	6
2.1. La ecuación de continuidad.....	7
2.2. La ecuación del momento lineal.....	7
2.3. La ecuación de la energía.....	8
3. Descripción del problema y condiciones geométricas.....	9
3.1. Vorticidad y función de corriente.....	10
3.2. Condiciones de contorno.....	12
4. Resolución numérica del problema.....	19
4.1. Adimensionalización de las ecuaciones y las condiciones de contorno.....	19
4.2. Discretización de las ecuaciones y de las condiciones de contorno.....	24
5. Método de resolución.....	34
6. Resultados.....	38
6.1. Análisis del comportamiento físico del sistema para $Re=0.1, 1$ y $16$ .....	39
6.2. Cálculo de la presión.....	44
6.3. Efecto de la resolución.....	46
6.4. Cambio del tamaño de la malla.....	50
6.5. Efecto de la elección de distintas condiciones de contorno.....	56
6.6. Importancia de las condiciones de partida para el método iterativo.....	58
6.7. Comparativa del método implementado con el método de Lattice Boltzmann.....	59
6.8. Número de Reynolds crítico.....	61
6.9. Efecto de un segundo obstáculo en las simulaciones.....	65
7. Conclusiones.....	68
8. Bibliografía.....	69
9. Anexo.....	71
9.1. Código general del programa.....	71
9.2. Código del método de Boltzmann.....	87

## 1. Introducción

El estudio del comportamiento de un flujo alrededor un obstáculo es un problema físico muy importante tanto desde el punto de vista teórico como práctico. Su campo de aplicación es muy amplio y abarca desde el estudio de las propiedades mecánicas de las espumas [1] a la aerodinámica de los puentes [2] o a las ondas de gravedad generadas en el flujo atmosférico [3] entre otros.

Las ecuaciones diferenciales parciales tienen un papel fundamental en la formulación matemática de prácticamente la totalidad de los fenómenos físicos, cuando estos están distribuidos en el espacio y en el tiempo y, los fluidos, como se va a estudiar, no son una excepción. Sin embargo, estas ecuaciones no pueden resolverse analíticamente, a excepción de los casos más simples, por lo que deben utilizarse métodos numéricos para obtener resultados cuantitativos. En estos tratamientos numéricos es habitual describir las variables dependientes (en el caso de los fluidos, la presión, la temperatura o la velocidad) por su valor en determinados puntos discretos (habitualmente en una malla), en función de las variables independientes (como el espacio y el tiempo), de tal manera que, utilizando las discretizaciones adecuadas, se consigue reducir las ecuaciones diferenciales en derivadas parciales a resolver a sistemas de ecuaciones no lineales en diferencias finitas.

En el presente trabajo, para el caso estacionario, se van a resolver unas ecuaciones diferenciales parciales de segundo orden de tipo elíptico, es decir, unas ecuaciones diferenciales en las que existen derivadas de segundo orden en cada una de las variables independientes, teniendo todas las derivadas el mismo signo cuando todos los términos se agrupan en un lado de la ecuación. Estas ecuaciones elípticas se diferencian de las parabólicas e hiperbólicas en que no son ecuaciones de evolución, es decir, en el problema el tiempo no juega un papel.

Para poder resolver numéricamente estas ecuaciones se va a necesitar determinar previamente las condiciones particulares del problema, es decir, antes de aplicar cualquier método numérico se deben fijar las condiciones geométricas, las propiedades físicas y las condiciones de contorno que le afectan.

En el apartado 2 se describen las ecuaciones fundamentales de la dinámica de fluidos, que permiten describir el comportamiento de gases y líquidos en movimiento.

En el apartado 3 se plantea el problema estacionario concreto a resolver. Se imponen una serie de simplificaciones al fluido y se detallan las condiciones geométricas del problema. En el apartado 3.1 se detallan las ecuaciones que se van a resolver en la representación elegida (función de corriente-vorticidad) y en el apartado 3.2 las condiciones de contorno que se imponen.

En el apartado 4 se describe el método de resolución numérica elegido, mediante la adimensionalización (apartado 4.1) y discretización (apartado 4.2) de las ecuaciones y condiciones de contorno del problema por medio de diferencias finitas.

En el apartado 5 se detalla el método numérico iterativo empleado para la resolución de las ecuaciones discretas, imponiendo un criterio de convergencia de las aproximaciones a la solución del problema.

En el apartado 6 se exponen los resultados obtenidos. En el apartado 6.1 se detallan los resultados obtenidos para tres números de Reynolds distintos, elegidos como ejemplos prototípicos de los distintos equilibrios de fuerzas en el régimen estacionario. En el apartado 6.2 se realiza el cálculo del campo de presión que se añade a los resultados de los campos de función de corriente y vorticidad del apartado 6.1. En los apartados 6.3, 6.4 y 6.5 se analiza el efecto en el flujo de la variación de algunos parámetros de la simulación, como son la resolución y el tamaño de la malla, así como de las condiciones de contorno impuestas. En el apartado 6.6 se describe la importancia de las condiciones de la inicialización de los campos discretos en la convergencia del método iterativo. En el apartado 6.7 se realiza una comparación con el método de Lattice Boltzmann, que se completa en el apartado 6.8 con la definición del número de Reynolds crítico. Y, por último, en el apartado 6.9 se extiende el código de para analizar el comportamiento de un sistema para dos obstáculos.

En el apartado 7 se presentan las conclusiones más importantes de este trabajo.

En el apartado 8 se aportan las referencias bibliográficas utilizadas.

Y finalmente, en el apartado 9 se aporta el código elaborado en el lenguaje de programación *Python*.

## 2. Ecuaciones de la dinámica de fluidos

En la mecánica de fluidos se asume la hipótesis del medio continuo, según la cual se considera que el fluido es continuo a lo largo del espacio que ocupa, ignorando las posibles discontinuidades existentes a nivel atómico o molecular. En consecuencia, en el tratamiento matemático ideal de un medio continuo se admite que no hay discontinuidades entre las partículas y que la descripción matemática de este medio y de sus propiedades se puede realizar mediante funciones continuas de las coordenadas espaciales y del tiempo.

A partir de esta hipótesis, se asume que todo fluido debe cumplir los siguientes postulados físicos

- El principio de conservación de la masa,
- el principio de la conservación del momento lineal,
- el principio de la conservación de la energía.

En la mecánica de medios continuos existen dos maneras distintas, pero equivalentes, de describir el movimiento de un medio continuo. En la primera, denominada descripción lagrangiana, se hace una descripción del movimiento siguiendo una de las parcelas individuales del medio, a lo largo de su trayectoria a través del espacio y del tiempo. En la segunda, denominada descripción euleriana, el sistema de referencia se encuentra en reposo, por lo que la caracterización del medio se realiza a través de los campos del mismo en las posiciones espaciales y temporales vistas en dicho sistema de referencia.

Al utilizar la descripción euleriana para analizar los distintos postulados se considera un elemento infinitesimal del medio en movimiento con el flujo. El movimiento de este elemento fluido, que se puede apreciar con más detalle en la Figura 1, se realiza a través del espacio cartesiano. Los vectores unitarios a lo largo de los ejes  $x$ ,  $y$  y  $z$  son  $\vec{i}$ ,  $\vec{j}$  y  $\vec{k}$  respectivamente.

El campo vectorial de velocidades en este espacio cartesiano está dado por

$$\vec{v}(x, y, z, t) = u(x, y, z, t)\vec{i} + v(x, y, z, t)\vec{j} + w(x, y, z, t)\vec{k}$$

donde  $u$ ,  $v$  y  $w$  son los campos que describen las componentes de la velocidad.

De la misma manera, el campo escalar de densidad  $\rho$  puede expresarse como

$$\rho = \rho(x, y, z, t)$$

En la descripción euleriana, la derivada ordinaria  $\partial/\partial t$ , no representa toda la variación por unidad de tiempo de una determinada propiedad del fluido siguiendo a la parcela fluida, ya que no tiene en cuenta el movimiento del fluido. Para reflejar esta variación se utiliza la derivada material, definida como el operador

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \vec{v} \cdot \vec{\nabla}$$

donde  $\vec{\nabla} = (\partial_x, \partial_y, \partial_z)$  es el operador gradiente.

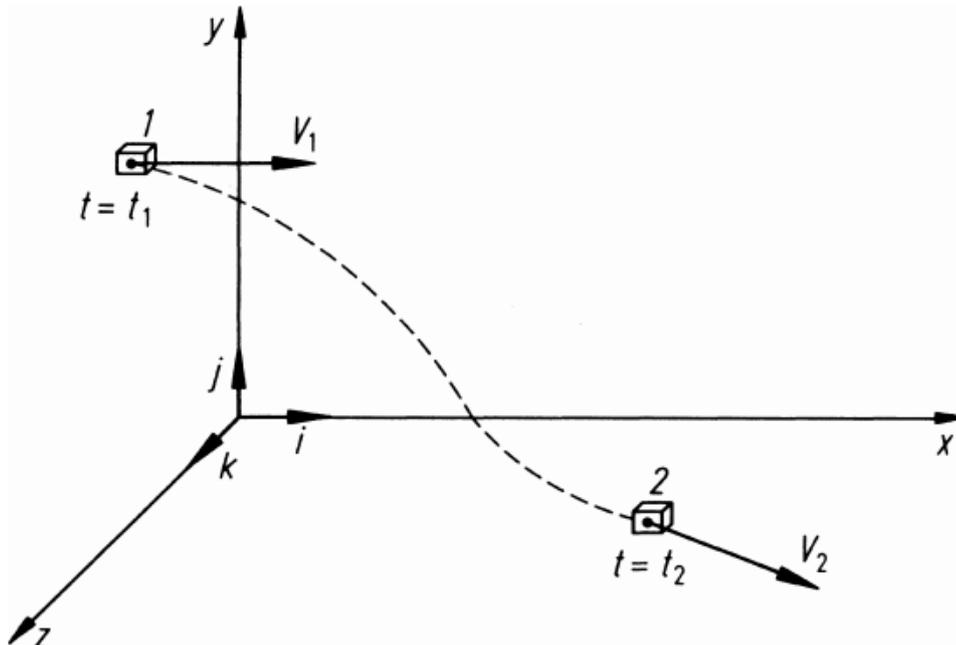


Figura 1. Translación de una parcela infinitesimal entre dos instantes de tiempo,  $t_1$  y  $t_2$ .

Llegados a este punto, se plantean matemáticamente los distintos postulados físicos en su forma euleriana.

### 2.1. La ecuación de continuidad

El principio de conservación de la masa establece que la masa no se puede crear ni destruir, no solo en el volumen total del fluido, sino también para cualquier subvolumen o parcela que se quiera del mismo. En la descripción euleriana este principio se expresa en forma de la denominada ecuación de continuidad en forma no-conservativa,

$$\frac{D\rho}{Dt} + \rho \vec{\nabla} \cdot \vec{v} = 0 \quad (1)$$

### 2.2. La ecuación del momento lineal

La conservación de momento lineal, que surge de aplicar la segunda ley de Newton al fluido como medio continuo, se expresa matemáticamente mediante las ecuaciones de Navier-Stokes, que en su expresión más general pueden escribirse como

$$\rho \frac{D\vec{v}}{Dt} = -\vec{\nabla}p + \vec{\nabla} \cdot \vec{\tau} + \vec{f}_{ext} \quad (2)$$

Donde  $p$  representa la presión y  $\vec{\tau} = \begin{pmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{xy} & \tau_{zz} \end{pmatrix}$  es el tensor de esfuerzos. Estas magnitudes recogen el efecto de las fuerzas internas que actúan sobre la superficie del elemento fluido, bien en forma de presión producida por el resto de elementos fluidos, bien en forma de fricción.

$\vec{f}_{ext}$  representa al conjunto de fuerzas externas (gravedad, Coriolis, propulsores, etc...) que pueden actuar sobre el elemento fluido.

### 2.3. La ecuación de la energía

La conservación de la energía se puede obtener al aplicar el primer principio de la termodinámica a un elemento del fluido. La variación de la energía en el elemento fluido se debe al flujo neto de calor hacia el elemento y al trabajo realizado sobre el elemento por las fuerzas internas y externas.

La ecuación de la energía, en función del campo de temperatura  $T = T(x, y, z, t)$  y de la capacidad calorífica volumétrica  $C_V$  puede escribirse como

$$\rho C_V \frac{DT}{Dt} = -(\vec{\nabla} \cdot \vec{q}) - T \left( \frac{\partial p}{\partial T} \right)_V (\vec{\nabla} \cdot \vec{v}) - (\vec{\tau} : \vec{\nabla} \vec{v}) \quad (3)$$

Donde  $\vec{q} = \vec{q}(x, y, z, t)$  representa la densidad de flujo de calor por unidad de masa.

### 3. Descripción del problema y condiciones geométricas

Se va a estudiar el comportamiento de un flujo libre sobre un obstáculo, con forma de ortoedro. Por simplicidad, se considerará que el obstáculo es infinito en una dirección (eje  $z$ ). Sus dimensiones serán  $L_x$  en la dirección  $x$  y  $2L_y$  en la dirección  $y$ . El flujo corriente arriba del obstáculo será constante y su velocidad será  $V_0$  en la dirección del eje  $x$ . Este flujo incidirá perpendicularmente sobre una de las caras del ortoedro.

En la Figura 2 puede observarse como las condiciones geométricas del problema hacen que el comportamiento del flujo sea el mismo en todos los planos perpendiculares al eje  $z$ . Por ello, solo será necesario estudiar el comportamiento del flujo en uno de dichos planos, pudiendo reducir el problema a la descripción de campos bidimensionales.

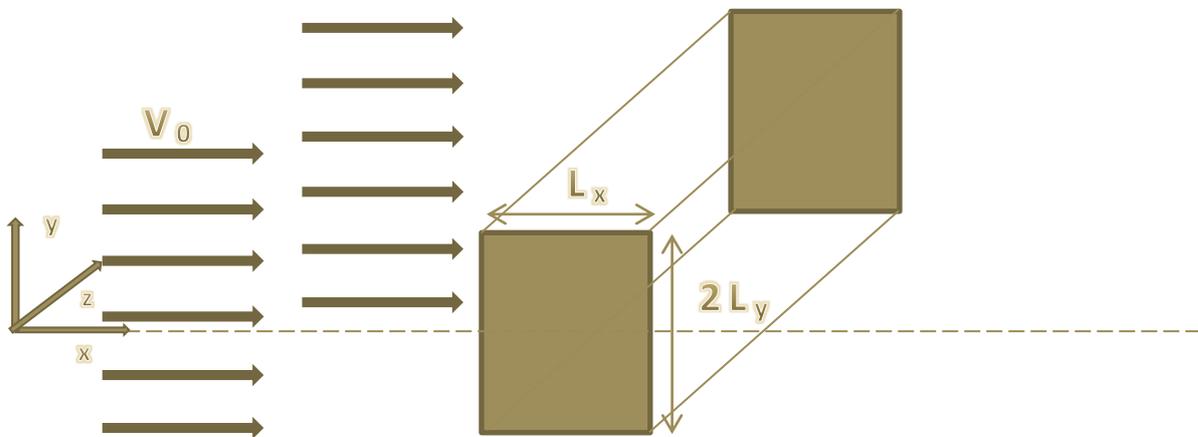


Figura 2. Representación de las condiciones geométricas impuestas en el problema.  $V_0$  representa la velocidad del flujo incidente.  $L_x$  y  $2L_y$ , son las dimensiones del obstáculo que tiene una longitud  $L_z \rightarrow \infty$  en la dirección del eje  $z$ .

Se va a considerar un fluido homogéneo, newtoniano, incompresible e isoterma en ausencia de fuerzas externas. En particular, estamos interesados en el estudio del fluido en un régimen estacionario, es decir, en un régimen en el que la velocidad en cada punto del espacio permanece constante en el tiempo y, por tanto, todas sus propiedades van a ser independientes del tiempo.

La primera consecuencia de utilizar estas aproximaciones es que la temperatura del sistema no va a variar y, por ello, no será necesario resolver la ecuación de la energía, (3).

La ecuación de continuidad, (1), se simplifica a

$$\nabla \cdot \vec{v} = 0 \tag{4}$$

La ecuación de Navier-Stokes se simplifica a tres términos, que representan las fuerzas inerciales, las fuerzas de presión y las fuerzas viscosas

$$(\vec{v} \cdot \nabla) \vec{v} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \vec{v} \tag{5}$$

donde  $\nu$  representa la viscosidad cinemática del fluido.

Teniendo en cuenta tanto las condiciones geométricas del problema como las aproximaciones físicas efectuadas, se pueden describir los campos únicamente en función de dos variables espaciales, siendo la velocidad

$$\vec{v}(x, y) = u(x, y)\vec{i} + v(x, y)\vec{j}$$

Reescribiendo las ecuaciones (4) y (5) en componentes se obtienen las ecuaciones necesarias para resolver el problema

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (6)$$

$$u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \nabla^2 u \quad (7)$$

$$u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = -\frac{1}{\rho} \frac{\partial p}{\partial y} + \nu \nabla^2 v \quad (8)$$

### 3.1. Vorticidad y función de corriente

Las ecuaciones (6), (7) y (8) permiten resolver el problema, ya que forman un sistema de tres ecuaciones diferenciales con tres incógnitas (los campos de las componentes de la velocidad,  $u$  y  $v$ , y el campo de presión,  $p$ ). Sin embargo, se va a reemplazar el campo de velocidad por los campos escalares equivalentes de la vorticidad y la función de corriente, lo que, como se verá posteriormente, presenta importantes ventajas.

Bajo las condiciones impuestas sobre el fluido se puede definir un vector función de corriente como  $\vec{\Psi} = (0, 0, \Psi)$ , tal que  $\Psi = \Psi(x, y)$  está relacionado con la velocidad mediante la ecuación

$$\vec{v} = \vec{\nabla} \times \vec{\Psi} \quad (9)$$

que en coordenadas cartesianas puede escribirse como

$$u(x, y) = \frac{\partial \Psi}{\partial y}, \quad v(x, y) = -\frac{\partial \Psi}{\partial x} \quad (10)$$

La función de corriente va a permitir representar las líneas de corriente, que a su vez representan las trayectorias de las partículas fluidas en un flujo estacionario.

La vorticidad es un campo vectorial que describe el movimiento de rotación local de un fluido cerca de un punto, tal como sería vista por un observador situado en el punto y que viaja con el fluido. La vorticidad se expresa como:

$$\vec{\xi}(x, y) = -\vec{\nabla} \times \vec{v} \quad (11)$$

que teniendo en cuenta las condiciones geométricas del problema, toma la forma

$$\vec{\xi}(x, y) = \left( \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} \right) \vec{k} \quad (12)$$

En el problema el vector vorticidad es siempre paralelo al eje  $z$  y, por tanto, de ahora en adelante, tendrá el tratamiento de un campo escalar:

$$\xi(x, y) = \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} \quad (13)$$

Teniendo en cuenta las ecuaciones (10) y (13) ambas magnitudes pueden relacionarse mediante la ecuación

$$\xi = \nabla^2 \Psi \quad (14)$$

El siguiente paso es obtener la denominada ecuación de la vorticidad. Para ello, se deriva la ecuación (7) respecto a  $y$  y la ecuación (8) respecto a  $x$ ,

$$\frac{\partial u}{\partial y} \frac{\partial u}{\partial x} + u \frac{\partial^2 u}{\partial x \partial y} + \frac{\partial v}{\partial y} \frac{\partial u}{\partial y} + v \frac{\partial^2 u}{\partial y^2} = -\frac{1}{\rho} \frac{\partial^2 p}{\partial x \partial y} + \nu \nabla^2 \left( \frac{\partial u}{\partial y} \right) \quad (15)$$

$$\frac{\partial u}{\partial x} \frac{\partial v}{\partial x} + u \frac{\partial^2 v}{\partial x^2} + \frac{\partial v}{\partial x} \frac{\partial v}{\partial y} + v \frac{\partial^2 v}{\partial x \partial y} = -\frac{1}{\rho} \frac{\partial^2 p}{\partial x \partial y} + \nu \nabla^2 \left( \frac{\partial v}{\partial x} \right) \quad (16)$$

se restan

$$\left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \left( \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} \right) + u \frac{\partial^2 u}{\partial x \partial y} + v \frac{\partial^2 u}{\partial y^2} - u \frac{\partial^2 v}{\partial x^2} - v \frac{\partial^2 v}{\partial x \partial y} = \nu \nabla^2 \left( \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} \right) \quad (17)$$

y haciendo uso de las ecuaciones (10) y (13), que definen la función de corriente y la vorticidad, se llega a

$$\nu \nabla^2 \xi = \left( \frac{\partial \Psi}{\partial y} \frac{\partial \xi}{\partial x} - \frac{\partial \Psi}{\partial x} \frac{\partial \xi}{\partial y} \right) \quad (18)$$

Para determinar la presión se deriva la ecuación (7) respecto a  $x$  y la ecuación (8) respecto a  $y$

$$\left( \frac{\partial u}{\partial x} \right)^2 + u \frac{\partial^2 u}{\partial x^2} + \frac{\partial v}{\partial x} \frac{\partial u}{\partial y} + v \frac{\partial^2 u}{\partial x \partial y} = -\frac{1}{\rho} \frac{\partial^2 p}{\partial x^2} + \nu \nabla^2 \left( \frac{\partial u}{\partial x} \right) \quad (19)$$

$$\frac{\partial u}{\partial y} \frac{\partial v}{\partial x} + u \frac{\partial^2 v}{\partial x \partial y} + \left( \frac{\partial v}{\partial y} \right)^2 + v \frac{\partial^2 v}{\partial y^2} = -\frac{1}{\rho} \frac{\partial^2 p}{\partial y^2} + \nu \nabla^2 \left( \frac{\partial v}{\partial y} \right) \quad (20)$$

y se suman, quedando

$$\begin{aligned} \left( \frac{\partial u}{\partial x} \right)^2 + \left( \frac{\partial v}{\partial y} \right)^2 + u \frac{\partial^2 u}{\partial x^2} + v \frac{\partial^2 v}{\partial y^2} + 2 \frac{\partial v}{\partial x} \frac{\partial u}{\partial y} + v \frac{\partial^2 u}{\partial x \partial y} + u \frac{\partial^2 v}{\partial x \partial y} \\ = -\frac{1}{\rho} \frac{\partial^2 p}{\partial x^2} - \frac{1}{\rho} \frac{\partial^2 p}{\partial y^2} + \nu \nabla^2 \left( \frac{\partial u}{\partial x} \right) + \nu \nabla^2 \left( \frac{\partial v}{\partial y} \right) \end{aligned} \quad (21)$$

Sustituyendo la definición de la función corriente, dada en la ecuación (10), y reordenando los términos se llega a

$$\nabla^2 p = 2\rho \left[ \left( \frac{\partial^2 \Psi}{\partial x^2} \right) \left( \frac{\partial^2 \Psi}{\partial y^2} \right) - \left( \frac{\partial^2 \Psi}{\partial x \partial y} \right)^2 \right] \quad (22)$$

Las ecuaciones (14), (18) y (22) forman el conjunto de ecuaciones a resolver. Este sistema de ecuaciones está definido en función de los campos estacionarios  $\Psi(x, y)$ ,  $\xi(x, y)$  y  $p(x, y)$ , y tiene la ventaja, respecto al sistema formado por las ecuaciones (6), (7) y (8), de tener desacoplada la presión, es decir, para calcular tanto la vorticidad como la función de corriente no será necesario conocer la presión. Este hecho es importante, ya que ya no será necesario resolver las tres ecuaciones simultáneamente, pudiendo resolver primero las ecuaciones (14) y (18) y, si posteriormente interesa, resolver la ecuación (22).

### 3.2. Condiciones de contorno

En este apartado, se va a restringir el dominio del problema, de manera que el sistema físico sea finito. Además, se van a plantear las condiciones de contorno necesarias para resolverlo.

Existen múltiples posibilidades en cuanto a la elección de las condiciones de contorno. Sin embargo, lo más habitual es utilizar condiciones de contorno de Dirichlet, de Von Neumann o mixtas. En las primeras, se fija el valor de la función en el contorno. En las segundas se fija el valor de la derivada de la función con respecto al vector normal en cada punto de la frontera. En las mixtas se fija el valor de una combinación lineal de la función y de su derivada respecto de la normal en la frontera.

Se van a definir las condiciones de contorno del problema bidimensional haciendo uso de la simetría. Al considerar que el obstáculo rectangular tiene una de sus caras perpendicular al flujo incidente, la línea central del rectángulo será una línea de simetría que, por conveniencia, se sitúa en  $y = 0$ . En la Figura 3 la línea punteada representa dicho eje de simetría. La principal ventaja de tenerla en cuenta es que solamente se necesita realizar la simulación numérica de la mitad del problema, reduciendo notablemente el tiempo de computación.

A la hora de establecer las condiciones de contorno, se divide el contorno en distintas partes claramente diferenciadas: el contorno del fluido en la línea central de simetría del obstáculo, tanto anterior como posterior a este, los contornos del obstáculo contiguos al fluido, y los distintos contornos externos del dominio. Para cada uno de estos contornos se va a realizar un análisis en profundidad de las condiciones de contorno, utilizando como referencia la Figura 4.

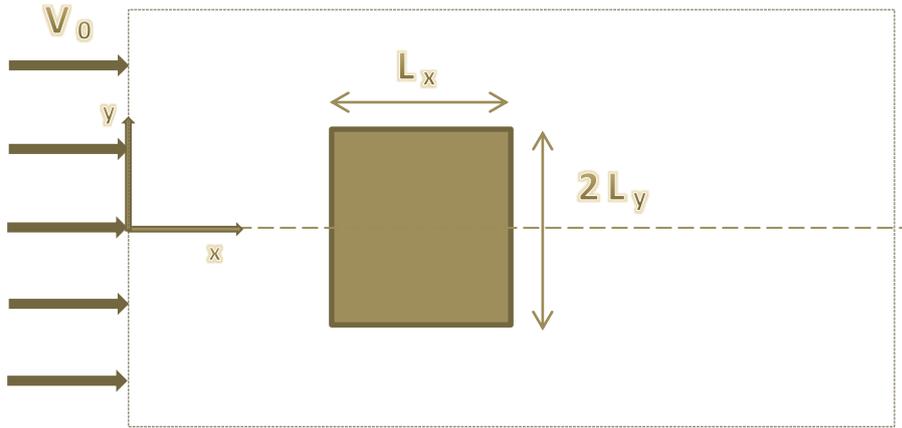


Figura 3. Sección transversal de la figura 2, en la que se impone un dominio de resolución finito. La línea punteada central representa la simetría del problema respecto al eje x.

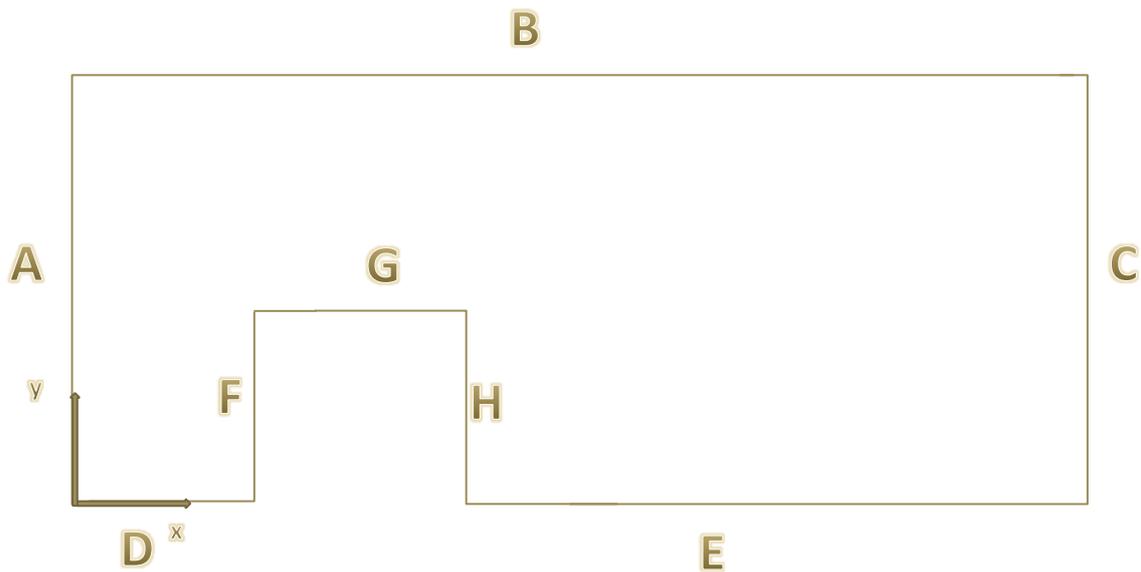


Figura 4. Representación del dominio de resolución del problema con los distintos contornos marcados con letras de la A a la H.

Antes de analizar los distintos contornos es necesario reescribir las ecuaciones (7) y (8) en función de la función de corriente y la vorticidad, ecuaciones (10) y (13), obteniendo

$$\frac{\partial \Psi}{\partial y} \frac{\partial^2 \Psi}{\partial x \partial y} + \frac{\partial \Psi}{\partial x} \frac{\partial^2 \Psi}{\partial y^2} = -\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \frac{\partial \xi}{\partial y} \quad (23)$$

$$-\frac{\partial \Psi}{\partial y} \frac{\partial^2 \Psi}{\partial x^2} + \frac{\partial \Psi}{\partial x} \frac{\partial^2 \Psi}{\partial x \partial y} = -\frac{1}{\rho} \frac{\partial p}{\partial y} - \nu \frac{\partial \xi}{\partial x} \quad (24)$$

Estas ecuaciones van a servir para definir las condiciones de contorno de tipo Neumann para el campo de presión en los distintos contornos.

A. Contorno A. 

El contorno A representa el contorno “corriente arriba” del obstáculo.

En el caso de no existir el obstáculo el flujo sería libre, y la solución del problema sería  $u^L(x, y) = V_0$ ,  $v^L(x, y) = 0$ , por lo que la función de corriente y la vorticidad tomarían la forma  $\Psi^L(x, y) = V_0 y$  y  $\xi^L(x, y) = 0$  en todo el espacio. La presión sería  $p^L(x, y) = p_0$ . Donde  $L$  hace referencia al flujo libre.

Teniendo en cuenta lo anterior parece razonable imponer la condición de flujo libre en el contorno A. Para la función de corriente se obtienen dos posibles condiciones de contorno,

$$u|_A = V_0 \rightarrow \frac{\partial \Psi}{\partial y} \Big|_A = V_0 \quad (25)$$

$$v|_A = 0 \rightarrow \frac{\partial \Psi}{\partial x} \Big|_A = 0 \quad (26)$$

La vorticidad será nula en el contorno,

$$\xi|_A = 0 \quad (27)$$

Además, al tener la misma vorticidad todos los puntos del contorno

$$\frac{\partial \xi}{\partial y} \Big|_A = 0 \quad (28)$$

Teniendo en cuenta estas condiciones se pueden analizar las ecuaciones (23) y (24) para obtener las condiciones de contorno para la presión:

$$(23)|_A \rightarrow \frac{\partial p}{\partial x} \Big|_A = -\rho V_0 \frac{\partial^2 \Psi}{\partial x \partial y} \Big|_A \quad (29)$$

$$(24)|_A \rightarrow \frac{\partial p}{\partial y} \Big|_A = \rho V_0 \frac{\partial^2 \Psi}{\partial x^2} \Big|_A - \rho v \frac{\partial \xi}{\partial x} \Big|_A \quad (30)$$

Así, se han obtenido dos posibles condiciones de contorno tanto para la función de corriente como para la presión. Para resolver el problema solamente es necesario aplicar una de ellas, dejando su elección para el apartado 4.2, en el que se discretizarán las condiciones de contorno.

B. Contorno B. 

En este contorno también se puede considerar que el flujo ha de tener las características del flujo libre, por lo que

$$u|_B = V_0 \rightarrow \frac{\partial \Psi}{\partial y} \Big|_B = V_0 \quad (31)$$

$$v|_B = 0 \rightarrow \frac{\partial \Psi}{\partial x} \Big|_B = 0 \quad (32)$$

Además, al cumplirse la condición dada por la ecuación (32) en todos los puntos del contorno B, también se cumplirá,

$$\left. \frac{\partial v}{\partial x} \right|_B = 0 \rightarrow \left. \frac{\partial^2 \Psi}{\partial x^2} \right|_B = 0 \quad (33)$$

Y para la vorticidad,

$$\xi|_B = 0 \quad (34)$$

Teniendo en cuenta la ecuación (33), se pueden obtener las condiciones de contorno para la presión,

$$(23)|_B \rightarrow \left. \frac{\partial p}{\partial x} \right|_B = \rho v \left. \frac{\partial \xi}{\partial y} \right|_B - \rho V_0 \left. \frac{\partial^2 \Psi}{\partial x \partial y} \right|_B \quad (35)$$

$$(24)|_B \rightarrow \left. \frac{\partial p}{\partial y} \right|_B = \rho V_0 \left. \frac{\partial^2 \Psi}{\partial x^2} \right|_B \equiv 0 \quad (36)$$

C. Contorno C. 

En este caso, la elección de las condiciones de contorno a aplicar no es intuitiva, ya que el efecto principal del obstáculo es el de perturbar el flujo “corriente abajo”. Por ello, no es adecuado imponer la condición de flujo libre. En este caso es importante situar el contorno lo suficientemente alejado del obstáculo, para poder observar claramente los efectos del obstáculo sobre el flujo.

Una elección razonable sería imponer que en el contorno C el flujo volverá a ser horizontal, sin imponer si se ha restaurado el flujo libre. Así, se obtiene una condición de contorno para la función de corriente,

$$v|_C = - \left. \frac{\partial \Psi}{\partial x} \right|_C = 0 \quad (37)$$

Además, teniendo en cuenta que  $v$  será la misma en todo el contorno, en particular nula, también se cumplirá,

$$\left. \frac{\partial v}{\partial y} \right|_C = 0 \rightarrow \left. \frac{\partial^2 \Psi}{\partial x \partial y} \right|_C = 0 \quad (38)$$

La elección de la condición de contorno para la vorticidad es más complicada que en los contornos A y B, ya que en este caso no se puede imponer una condición de Dirichlet. Una opción es intentar imponer una condición de Von Neumann, en la que consideramos que la vorticidad no variará en la horizontal en el contorno C,

$$\left. \frac{\partial \xi}{\partial x} \right|_C = 0 \quad (39)$$

Una segunda opción es imponer la condición de Von Neumann sobre la derivada segunda de la vorticidad en la dirección normal al contorno,

$$\left. \frac{\partial^2 \xi}{\partial x^2} \right|_c = 0 \quad (40)$$

Para la presión, teniendo en cuenta la ecuación (38) en las ecuaciones (23) y (24) se obtienen las siguientes condiciones:

$$(23)|_c \rightarrow \left. \frac{\partial p}{\partial x} \right|_c = \rho v \left. \frac{\partial \xi}{\partial y} \right|_c \quad (41)$$

$$(24)|_c \rightarrow \left. \frac{\partial p}{\partial y} \right|_c = \rho \left. \frac{\partial \Psi}{\partial y} \right|_c \left. \frac{\partial^2 \Psi}{\partial x^2} \right|_c - \rho v \left. \frac{\partial \xi}{\partial x} \right|_c \quad (42)$$

D. Contorno D. 

En este contorno, dado que es una línea de simetría, la componente  $y$  de la velocidad,  $v$ , debe de hacerse cero en todos los puntos, lo que implica que la derivada respecto a  $x$  de la función de corriente también lo sea

$$v|_D = 0 \rightarrow \left. \frac{\partial \Psi}{\partial x} \right|_D = 0 \quad (43)$$

Este hecho implica que D es una línea de corriente  $\Psi$ , por tanto, la función de corriente será constante,  $\Psi|_D = cte$ . El valor de dicha constante se fijará posteriormente.

Al ser  $v|_D = 0$  para todos los puntos del contorno, se tiene que

$$\left. \frac{\partial v}{\partial x} \right|_D = 0 \quad (44)$$

Además, debido a la simetría, en cualquier punto de este contorno la derivada normal de la componente tangencial de la velocidad será cero, es decir, la velocidad horizontal no va a depender de  $y$

$$\left. \frac{\partial u}{\partial y} \right|_D = 0 \quad (45)$$

Las ecuaciones (44) y (45) representan los dos términos de la vorticidad para el contorno D, tal como se definió en la ecuación (13). Por ello, la vorticidad en D debe ser cero,

$$\xi|_D = 0 \quad (46)$$

Para obtener en este contorno las condiciones de contorno para la presión es suficiente con tener en cuenta que por simetría se tiene que  $\left. \frac{\partial p}{\partial y} \right|_D = 0$ . Ecuación fácilmente obtenible cuando se analizan las ecuaciones para la presión en el contorno:

$$(23)|_D \rightarrow \left. \frac{\partial p}{\partial x} \right|_D = -\rho \left. \frac{\partial \Psi}{\partial y} \right|_D \left. \frac{\partial^2 \Psi}{\partial x \partial y} \right|_D + \rho v \left. \frac{\partial \xi}{\partial y} \right|_D \quad (47)$$

$$(24)|_D \rightarrow \left. \frac{\partial p}{\partial y} \right|_D = 0 \quad (48)$$

E. Contorno E. 

Las condiciones de contorno responden a la misma justificación física que las del contorno D. El contorno E está en la línea de simetría, por lo que también será una línea de corriente, con  $\Psi|_E = cte$ . Además, también por simetría, se tiene que

$$\xi|_E = 0 \quad (49)$$

$$\left. \frac{\partial p}{\partial y} \right|_E = 0 \quad (50)$$

Las posibles condiciones de contorno para la presión serán

$$(23)|_E \rightarrow \left. \frac{\partial p}{\partial x} \right|_E = -\rho \left. \frac{\partial \Psi}{\partial y} \right|_E \left. \frac{\partial^2 \Psi}{\partial x \partial y} \right|_E + \rho v \left. \frac{\partial \xi}{\partial y} \right|_E \quad (51)$$

$$(24)|_E \rightarrow \left. \frac{\partial p}{\partial y} \right|_E = 0 \quad (52)$$

F. Contorno F. 

En el contorno del obstáculo se debe tener en cuenta la condición de contorno “no-slip”, que establece que cuando un fluido viscoso fluye sobre un obstáculo sólido, no existe un movimiento relativo entre el fluido y el sólido en la superficie frontera del sólido [4]. Esta condición puede expresarse como

$$v_{normal} = 0 \quad (53)$$

$$v_{tangencial} = v_{pared\ obstáculo} \quad (54)$$

En el problema que se está estudiando, el obstáculo se encuentra en reposo, por lo que en todos los puntos de los contornos del obstáculo tanto la velocidad normal como la tangencial deben de ser nulas. Esto hace que la función de corriente en la superficie F no dependa ni de  $x$  ni de  $y$ , es decir, sea constante. Así, se tiene que  $\Psi|_F = cte$  es una línea de corriente.

Esta condición se va a cumplir también para los contornos superior y posterior del obstáculo y, por ello, también se tiene que  $\Psi|_G = cte$  y  $\Psi|_H = cte$ . Pero además, anteriormente se vio que los contornos D y E eran también líneas de corriente. Por ello, se puede considerar que toda la superficie DFGHE es una línea de corriente, a la que se le puede asignar arbitrariamente un valor. Así pues, por conveniencia se tomará

$$\Psi|_D = \Psi|_F = \Psi|_G = \Psi|_H = \Psi|_E = 0 \quad (55)$$

Volviendo al caso particular del contorno F, en la dirección normal a la superficie la velocidad se anula, lo que conduce a obtener la condición de contorno para la vorticidad

$$\xi|_F = \left. \frac{\partial u}{\partial y} \right|_F - \left. \frac{\partial v}{\partial x} \right|_F = - \left. \frac{\partial v}{\partial x} \right|_F = \left. \frac{\partial^2 \Psi}{\partial x^2} \right|_F \quad (56)$$

Para obtener la condición de contorno para la presión se sustituyen en las ecuaciones (23) y (24) las condiciones ya mencionadas, quedando

$$(23)|_F \rightarrow \frac{\partial p}{\partial x}|_F = \rho v \frac{\partial \xi}{\partial y}|_F \quad (57)$$

$$(24)|_F \rightarrow \frac{\partial p}{\partial y}|_F = -\rho v \frac{\partial \xi}{\partial x}|_F \quad (58)$$

G. Contorno G. 

En este contorno la situación es la misma que en el contorno F, salvo que en este caso la dirección normal a la superficie es la del vector unitario  $\vec{j}$ .

La condición de contorno para la función de corriente ya fue definida en la ecuación (55),  $\Psi|_G = 0$ .

Aplicando la condición “no slip” y recordando que

$$\frac{\partial v}{\partial x}|_G = -\frac{\partial^2 \Psi}{\partial x^2}|_G = 0 \quad (59)$$

se obtiene la condición de contorno para la vorticidad

$$\xi|_G = \frac{\partial u}{\partial y}|_G - \frac{\partial v}{\partial x}|_G = \frac{\partial u}{\partial y}|_G = \frac{\partial^2 \Psi}{\partial y^2}|_G \quad (60)$$

Para la presión se tiene que

$$(23)|_G \rightarrow \frac{\partial p}{\partial x}|_G = \rho v \frac{\partial \xi}{\partial y}|_G \quad (61)$$

$$(24)|_G \rightarrow \frac{\partial p}{\partial y}|_G = -\rho v \frac{\partial \xi}{\partial x}|_G \quad (62)$$

H. Contorno H. 

La condición de contorno para la función de corriente ya fue definida en la ecuación (55),  $\Psi|_H = 0$ .

En la dirección normal a la superficie la velocidad se anula, lo que conduce a obtener la condición de contorno para la vorticidad

$$\xi|_H = \frac{\partial u}{\partial y}|_H - \frac{\partial v}{\partial x}|_H = -\frac{\partial v}{\partial x}|_H = \frac{\partial^2 \Psi}{\partial x^2}|_H \quad (63)$$

Y para la presión se tiene que

$$(23)|_H \rightarrow \frac{\partial p}{\partial x}|_H = \rho v \frac{\partial \xi}{\partial y}|_H \quad (64)$$

$$(24)|_H \rightarrow \frac{\partial p}{\partial y}|_H = -\rho v \frac{\partial \xi}{\partial x}|_H \quad (65)$$

## 4. Resolución numérica del problema

En el apartado anterior se completó la modelización del problema a resolver, estableciendo las ecuaciones físicas que lo describen y las condiciones de contorno necesarias para resolverlo. En este apartado se busca obtener una serie de algoritmos que permitan obtener los resultados numéricos.

Para ello, en el apartado 4.1 se adimensionalizarán las ecuaciones y las condiciones de contorno, para posteriormente, en el apartado 4.2, discretizarlas.

### 4.1. Adimensionalización de las ecuaciones y las condiciones de contorno

Con la adimensionalización de un problema se busca reducir el número de parámetros necesarios para resolverlo. Para ello, mediante un cambio de variables, se busca un problema numérico equivalente y adimensional que requiera un menor número de parámetros. Esto permite realizar un análisis del comportamiento del problema para diferentes valores de los parámetros adimensionales.

Para adimensionalizar el problema se aplica el teorema  $\pi$  de Buckingham, que establece que cualquier problema con  $m$  parámetros independientes y  $n$  magnitudes fundamentales puede ser adimensionalizado, reduciéndose a un sistema con  $(m - n)$  parámetros adimensionales.

Los parámetros que se pueden variar independientemente son las constantes físicas del fluido y las magnitudes características del problema. De las ecuaciones (14), (18) y (22) se obtiene que las constantes físicas del fluido son la densidad y la viscosidad dinámica. Como magnitudes características se eligen la velocidad y la presión del flujo libre,  $V_0$  y  $p_0$ , y la longitud del obstáculo,  $L_x$ . Así, se tienen los siguientes parámetros independientes

$$m = \{\rho, \nu, V_0, p_0, L_x\} = 5$$

Las magnitudes fundamentales, por otro lado, son longitud, masa y tiempo, ya que la temperatura se considera constante.

$$n = \{L, M, t\} = 3$$

De manera que se deben obtener dos parámetros adimensionales, como se verá más adelante.

$$\pi = \{\pi_1, \pi_2\} = (m - n) = 2$$

Teniendo esto en cuenta se pueden adimensionalizar las distintas variables con su respectiva unidad-magnitud. La forma adimensional se notará mediante un asterisco:

$$\begin{aligned} u^* &= \frac{u}{V_0} & v^* &= \frac{v}{V_0} & p^* &= \frac{p}{p_0} \\ x^* &= \frac{x}{L_x} & y^* &= \frac{y}{L_x} \end{aligned} \tag{66}$$

Teniendo en cuenta las ecuaciones (10) y (13), se pueden adimensionalizar la vorticidad y la función de corriente

$$\xi^* = \xi \frac{L_x}{V_0} \quad (67)$$

$$\Psi^* = \frac{\Psi}{V_0 L_x} \quad (68)$$

Además, se define la forma adimensional de los operadores gradiente y nabra como

$$\begin{aligned} \nabla^* &= \left( \frac{\partial}{\partial x^*}, \frac{\partial}{\partial y^*} \right) = L_x \left( \frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right) = L_x \nabla \\ \Delta^* &= \left( \frac{\partial^2}{\partial x^{*2}} + \frac{\partial^2}{\partial y^{*2}} \right) = L_x^2 \left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) = L_x^2 \Delta \end{aligned} \quad (69)$$

Si ahora se sustituyen estas variables adimensionales en las ecuaciones problema, (14), (18) y (22), se tiene que:

$$\xi^* = \nabla^{*2} \Psi^* \quad (70)$$

$$\nabla^{*2} \xi^* = \frac{V_0 L_x}{\nu} \left( \frac{\partial \Psi^*}{\partial y^*} \frac{\partial \xi^*}{\partial x^*} - \frac{\partial \Psi^*}{\partial x^*} \frac{\partial \xi^*}{\partial y^*} \right) \quad (71)$$

$$\nabla^{*2} p^* = \frac{2\rho V_0^2}{p_0} \left[ \left( \frac{\partial^2 \Psi^*}{\partial x^{*2}} \right) \left( \frac{\partial^2 \Psi^*}{\partial y^{*2}} \right) - \left( \frac{\partial^2 \Psi^*}{\partial x^* \partial y^*} \right)^2 \right] \quad (72)$$

Se observa como las ecuaciones (71) y (72) contienen además de las variables adimensionales dos parámetros adimensionales, en consonancia con el teorema  $\pi$  de Buckingham. Estos parámetros adimensionales se corresponden con las definiciones de los números de Reynolds y de Euler, definidas como

$$Re = \frac{V_0 L_x}{\nu} \quad (73)$$

$$Eu = \frac{p_0}{\rho V_0^2} \quad (74)$$

El número de Reynolds expresa la relación entre las fuerzas inerciales y las fuerzas viscosas a partir de la velocidad característica y la longitud característica,  $V_0$  y  $L_x$ , en la macroescala.

El número de Euler relaciona las fuerzas de presión con las fuerzas inerciales.

Una vez adimensionalizadas las ecuaciones que rigen el comportamiento del problema, es necesario adimensionalizar las condiciones de contorno. Para ello, se utilizará la notación definida en la Figura 4.

A. Contorno A. 

Para la función de corriente se tiene que las ecuaciones (25) y (26) adimensionalizadas toman la forma

$$\left. \frac{\partial \Psi^*}{\partial y^*} \right|_A = 1 \quad (75)$$

$$\left. \frac{\partial \Psi^*}{\partial x^*} \right|_A = 0 \quad (76)$$

La ecuación de la vorticidad en el contorno A, ecuación (27), tiene como forma adimensionalizada

$$\xi^*|_A = 0 \quad (77)$$

Para la presión existen dos posibles condiciones de contorno, dadas por las ecuaciones (29) y (30), que adimensionalizadas se convierten en

$$\left. \frac{\partial p^*}{\partial x^*} \right|_A = - \frac{1}{Eu} \left. \frac{\partial^2 \Psi^*}{\partial x^* \partial y^*} \right|_A \quad (78)$$

$$\left. \frac{\partial p^*}{\partial y^*} \right|_A = \frac{1}{Eu} \left. \frac{\partial^2 \Psi^*}{\partial x^{*2}} \right|_A - \frac{1}{ReEu} \left. \frac{\partial \xi^*}{\partial x^*} \right|_A \quad (79)$$

B. Contorno B. 

Para la función de corriente se tiene que las ecuaciones (31) y (32) toman la forma

$$\left. \frac{\partial \Psi^*}{\partial y^*} \right|_B = 1 \quad (80)$$

$$\left. \frac{\partial \Psi^*}{\partial x^*} \right|_B = 0 \quad (81)$$

La ecuación de la vorticidad, definida por (34), toma la forma

$$\xi^*|_B = 0 \quad (82)$$

Para la presión, la adimensionalización de las ecuaciones (35) y (36) conduce a

$$\left. \frac{\partial p^*}{\partial x^*} \right|_B = \frac{1}{ReEu} \left. \frac{\partial \xi^*}{\partial y^*} \right|_B - \frac{1}{Eu} \left. \frac{\partial^2 \Psi^*}{\partial x^* \partial y^*} \right|_B \quad (83)$$

$$\left. \frac{\partial p^*}{\partial y^*} \right|_B = 0 \quad (84)$$

C. Contorno C. 

La adimensionalización para la función de corriente de la ecuación (37) conduce a

$$\left. \frac{\partial \Psi^*}{\partial x^*} \right|_C = 0 \quad (85)$$

Para la vorticidad las dos posibles condiciones de contorno, dadas por las ecuaciones (39) y (40), toman como forma adimensional las ecuaciones

$$\left. \frac{\partial \xi^*}{\partial x^*} \right|_C = 0 \quad (86)$$

$$\left. \frac{\partial^2 \xi^*}{\partial x^{*2}} \right|_C = 0 \quad (87)$$

Para la presión la adimensionalización de las ecuaciones (41) y (42) conduce a

$$\left. \frac{\partial p^*}{\partial x^*} \right|_C = \frac{1}{ReEu} \left. \frac{\partial \xi^*}{\partial y^*} \right|_C \quad (88)$$

$$\left. \frac{\partial p^*}{\partial y^*} \right|_C = \frac{1}{Eu} \left. \frac{\partial \Psi^*}{\partial y^*} \right|_C - \frac{1}{ReEu} \left. \frac{\partial \xi^*}{\partial x^*} \right|_C \quad (89)$$

D. Contorno D. 

Para la función de corriente y la vorticidad la adimensionalización de las ecuaciones (55) y (46) es simplemente

$$\Psi^*|_D = 0 \quad (90)$$

$$\xi^*|_D = 0 \quad (91)$$

Para la presión las condiciones de contorno obtenidas a partir de la adimensionalización de las ecuaciones (47) y (48) son

$$\left. \frac{\partial p^*}{\partial x^*} \right|_D = - \frac{1}{Eu} \left. \frac{\partial \Psi^*}{\partial y^*} \right|_D - \left. \frac{\partial^2 \Psi^*}{\partial x^* \partial y^*} \right|_D + \frac{1}{ReEu} \left. \frac{\partial \xi^*}{\partial y^*} \right|_D \quad (92)$$

$$\left. \frac{\partial p^*}{\partial y^*} \right|_D = 0 \quad (93)$$

E. Contorno E. 

Para el contorno E, la adimensionalización de las condiciones de contorno para la función de corriente y la vorticidad, dadas por (49) y (50), es

$$\Psi^*|_E = 0 \quad (94)$$

$$\xi^*|_E = 0 \quad (95)$$

Para la presión, la adimensionalización de las ecuaciones (51) y (52) da como resultado

$$\left. \frac{\partial p^*}{\partial x^*} \right|_E = - \frac{1}{Eu} \left. \frac{\partial \Psi^*}{\partial y^*} \right|_E - \left. \frac{\partial^2 \Psi^*}{\partial x^* \partial y^*} \right|_E + \frac{1}{ReEu} \left. \frac{\partial \xi^*}{\partial y^*} \right|_E \quad (96)$$

$$\left. \frac{\partial p^*}{\partial y^*} \right|_E = 0 \quad (97)$$

F. Contorno F 

La adimensionalización de la ecuación de contorno para la función de corriente, (55), es

$$\Psi^*|_F = 0 \quad (98)$$

Para la vorticidad en el contorno F es necesario adimensionalizar la ecuación (56), lo que conduce a

$$\xi^*|_F = \frac{\partial^2 \Psi^*}{\partial x^{*2}} \Big|_F \quad (99)$$

Las condiciones de contorno para la presión vienen descritas por las ecuaciones (57) y (58), que adimensionalizadas se expresan como

$$\frac{\partial p^*}{\partial x^*} \Big|_F = \frac{1}{ReEu} \frac{\partial \xi^*}{\partial y^*} \Big|_F \quad (100)$$

$$\frac{\partial p^*}{\partial y^*} \Big|_F = -\frac{1}{ReEu} \frac{\partial \xi^*}{\partial x^*} \Big|_F \quad (101)$$

G. Contorno G. 

Las condiciones son muy parecidas a las del contorno F, por lo que partiendo de las ecuaciones (55) y (60) se tiene

$$\Psi^*|_G = 0 \quad (102)$$

$$\xi^*|_G = \frac{\partial^2 \Psi^*}{\partial y^{*2}} \Big|_G \quad (103)$$

Para la presión se adimensionalizan las ecuaciones (61) y (62),

$$\frac{\partial p^*}{\partial x^*} \Big|_G = \frac{1}{ReEu} \frac{\partial \xi^*}{\partial y^*} \Big|_G \quad (104)$$

$$\frac{\partial p^*}{\partial y^*} \Big|_G = -\frac{1}{ReEu} \frac{\partial \xi^*}{\partial x^*} \Big|_G \quad (105)$$

H. Contorno H. 

La adimensionalización de la ecuación de contorno para la función de corriente, (55), es simplemente,

$$\Psi^*|_H = 0 \quad (106)$$

Para la vorticidad en el contorno H la adimensionalización de la ecuación (63), da como resultado

$$\xi^*|_H = \frac{\partial^2 \Psi^*}{\partial x^{*2}} \Big|_H \quad (107)$$

Las condiciones de contorno para la presión vienen descritas por las ecuaciones (64) y (65), que adimensionalizadas se expresan como

$$\frac{\partial p^*}{\partial x^*} \Big|_H = \frac{1}{ReEu} \frac{\partial \xi^*}{\partial y^*} \Big|_H \quad (108)$$

$$\frac{\partial p^*}{\partial y^*} \Big|_H = -\frac{1}{ReEu} \frac{\partial \xi^*}{\partial x^*} \Big|_H \quad (109)$$

## 4.2. Discretización de las ecuaciones y de las condiciones de contorno

La discretización del problema tiene como objetivo transformar el sistema de ecuaciones en derivadas parciales, donde las variables son continuas, en un sistema de ecuaciones algebraicas en un número finito de puntos. Este sistema de ecuaciones algebraicas podrá resolverse para obtener unos resultados cuantitativos de las magnitudes relevantes del problema.

Para la discretización se va a utilizar el método de las diferencias finitas.

La principal ventaja de haber escogido la descripción euleriana en el apartado 2, es que permite seleccionar, de manera sencilla, los puntos en el espacio donde se van a discretizar las ecuaciones. Su principal desventaja es que, al tratar con puntos fijos en el espacio, no tiene en cuenta posibles inhomogeneidades en el flujo. Además, dependiendo de la malla empleada, la discretización de las condiciones de contorno puede ser complicada [5].

La forma más sencilla de seleccionar los puntos en donde se van a aplicar las ecuaciones discretizadas es crear una malla regular. Se ha escogido seleccionar las coordenadas  $x$  e  $y$  en intervalos regulares de longitud,  $h$ , de tal manera que  $\Delta x = \Delta y = h$ , por lo que la malla será cuadrada. A esta longitud  $h$  se la conoce como tamaño de la red. En la Figura 5 se puede observar la malla utilizada para realizar la simulación del flujo del problema.

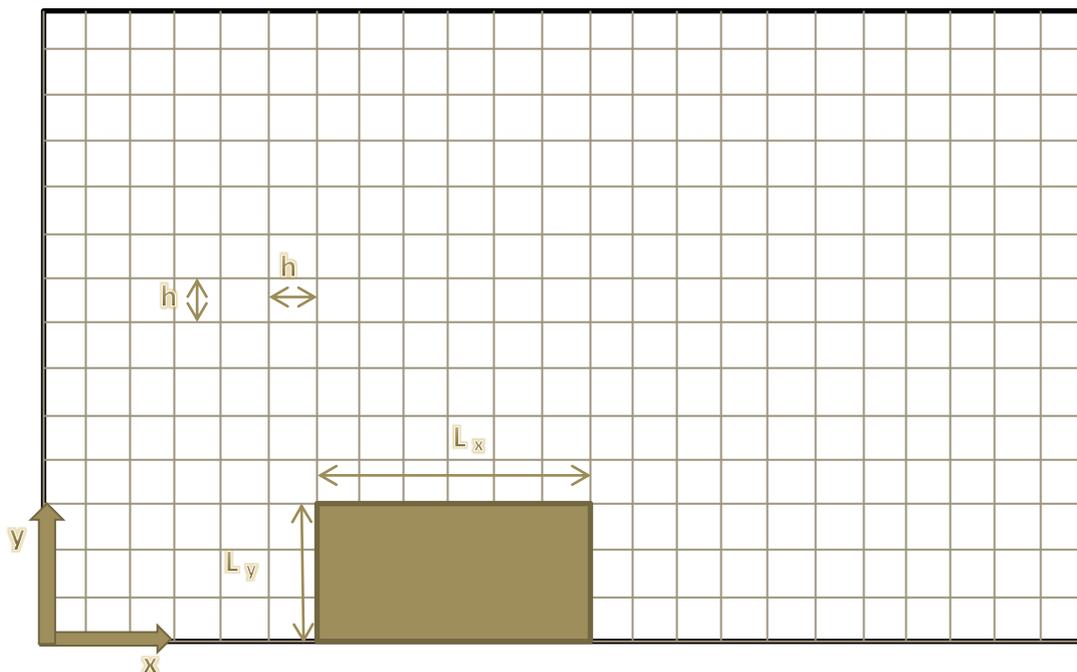


Figura 5. Malla de resolución del problema.

El tamaño de red adimensional será:

$$h^* = \frac{h}{L_x} \quad (110)$$

Y, lógicamente, la longitud del obstáculo pasará a ser la unidad en la dirección  $x$ ,

$$L_x^* = \frac{L_x}{L_x} = 1 \quad (111)$$

$$L_y^* = \frac{L_y}{L_x} \quad (112)$$

De tal manera que la vorticidad y las líneas de corriente van a ser calculadas en

$$\left. \begin{aligned} \xi^*(x^*, y^*) &= \xi^*(ih^*, jh^*) = \xi_{i,j}^* \\ \Psi^*(x^*, y^*) &= \Psi^*(ih^*, jh^*) = \Psi_{i,j}^* \end{aligned} \right\} \quad i = 1, \dots, N_x \quad j = 1, \dots, N_y \quad (113)$$

donde  $N_x$  es el número de puntos en el eje  $x$  y  $N_y$  es el número de puntos en el eje  $y$ .

Además, es conveniente definir el tamaño de la malla en función de dos magnitudes adimensionalizadas, su largo,  $L^*$ , y su alto,  $H^*$ .

$$L^* = N_x h^* \quad (114)$$

$$H^* = N_y h^* \quad (115)$$

Para discretizar las ecuaciones adimensionales del problema, (70), (71) y (72), se va a utilizar el esquema de diferencias finitas centradas para las variables espaciales  $(x^*, y^*)$ . Para los contornos, en los que no se tendrán puntos centrados no va a ser siempre posible, por lo que en ocasiones será necesario utilizar diferencias finitas adelantadas y atrasadas.

Para una función genérica,  $f(x^*, y^*)$ , que puede representar tanto la vorticidad como la función de corriente o la presión adimensionalizadas, se han utilizado las siguientes fórmulas para las derivadas.

- Diferencias finitas centradas:

Para construir las formula de diferencias centradas para  $\frac{\partial f(x^*, y^*)}{\partial x^*}$ , se toman los desarrollos en serie de Taylor de  $f(x^* + h^*, y^*)$  y  $f(x^* - h^*, y^*)$  en torno al punto  $(x^*, y^*)$

$$f(x^* + h^*, y^*) = f(x^*, y^*) + h^* \frac{\partial f(x^*, y^*)}{\partial x^*} + \dots + \frac{h^{*n}}{n!} \frac{\partial^n f(x^*, y^*)}{\partial y^{*n}} + \dots \quad (116)$$

$$f(x^* - h^*, y^*) = f(x^*, y^*) - h^* \frac{\partial f(x^*, y^*)}{\partial x^*} + \dots + (-1)^n \frac{h^{*n}}{n!} \frac{\partial^n f(x^*, y^*)}{\partial y^{*n}} + \dots \quad (117)$$

Que si se restan permiten obtener una expresión para  $\frac{\partial f(x^*, y^*)}{\partial x^*}$

$$\frac{\partial f(x^*, y^*)}{\partial x^*} = \frac{f(x^* + h^*, y^*) - f(x^* - h^*, y^*)}{2h^*} + O(h^{*2}) \quad (118)$$

Donde  $O(h^{*2})$ , es el error cometido al truncar la serie, despreciados los términos de orden mayor o igual a  $h^{*2}$ .

El resto de fórmulas pueden obtenerse de manera similar utilizando desarrollos de Taylor

$$\frac{\partial f(x^*, y^*)}{\partial y^*} = \frac{f(x^*, y^* + h^*) - f(x^*, y^* - h^*)}{2h^*} + O(h^{*2}) \quad (119)$$

$$\frac{\partial^2 f(x^*, y^*)}{\partial x^{*2}} = \frac{f(x^* + h^*, y^*) - 2f(x^*, y^*) + f(x^* - h^*, y^*)}{h^{*2}} + O(h^{*2}) \quad (120)$$

$$\frac{\partial^2 f(x^*, y^*)}{\partial y^{*2}} = \frac{f(x^*, y^* + h^*) - 2f(x^*, y^*) + f(x^*, y^* - h^*)}{h^{*2}} + O(h^{*2}) \quad (121)$$

$$\frac{\partial^2 f(x^*, y^*)}{\partial x^* \partial y^*} = \frac{f(x^* + h^*, y^* + h^*) - f(x^* + h^*, y^* - h^*) - f(x^* - h^*, y^* + h^*) + f(x^* - h^*, y^* - h^*)}{4h^{*2}} + O(h^{*2}) \quad (122)$$

- Diferencias finitas adelantadas:

$$\frac{\partial f(x^*, y^*)}{\partial x^*} = \frac{f(x^* + h^*, y^*) - f(x^*, y^*)}{h^*} + O(h^*) \quad (123)$$

$$\frac{\partial f(x^*, y^*)}{\partial y^*} = \frac{f(x^*, y^* + h^*) - f(x^*, y^*)}{h^*} + O(h^*) \quad (124)$$

$$\frac{\partial^2 f(x^*, y^*)}{\partial x^{*2}} = \frac{f(x^* + 2h^*, y^*) - 2f(x^* + h^*, y^*) + f(x^*, y^*)}{h^{*2}} + O(h^*) \quad (125)$$

$$\frac{\partial^2 f(x^*, y^*)}{\partial y^{*2}} = \frac{f(x^*, y^* + 2h^*) - 2f(x^*, y^* + h^*) + f(x^*, y^*)}{h^{*2}} + O(h^*) \quad (126)$$

$$\frac{\partial^2 f(x^*, y^*)}{\partial x^* \partial y^*} = \frac{f(x^* + h^*, y^* + h^*) - f(x^* + h^*, y^*) - f(x^*, y^* + h^*) + f(x^*, y^*)}{h^{*2}} + O(h^*) \quad (127)$$

- Diferencias finitas atrasadas:

$$\frac{\partial f(x^*, y^*)}{\partial x^*} = \frac{f(x^*, y^*) - f(x^* - h^*, y^*)}{h^*} + O(h^*) \quad (128)$$

$$\frac{\partial f(x^*, y^*)}{\partial y^*} = \frac{f(x^*, y^*) - f(x^*, y^* - h^*)}{h^*} + O(h^*) \quad (129)$$

$$\frac{\partial^2 f(x^*, y^*)}{\partial x^{*2}} = \frac{f(x^*, y^*) - 2f(x^* - h^*, y^*) + f(x^* - 2h^*, y^*)}{h^{*2}} + O(h^*) \quad (130)$$

$$\frac{\partial^2 f(x^*, y^*)}{\partial y^{*2}} = \frac{f(x^*, y^*) - 2f(x^*, y^* - h^*) + f(x^*, y^* - 2h^*)}{h^{*2}} + O(h^*) \quad (131)$$

$$\frac{\partial^2 f(x^*, y^*)}{\partial x^* \partial y^*} = \frac{f(x^* - h^*, y^* - h^*) - f(x^*, y^* - h^*) - f(x^* - h^*, y^*) + f(x^*, y^*)}{h^{*2}} + O(h^*) \quad (132)$$

Utilizando las fórmulas (120) y (121), la discretización de la ecuación (70) da como resultado

$$\xi_{i,j}^* = \frac{\Psi_{i+1,j}^* + \Psi_{i-1,j}^* + \Psi_{i,j+1}^* + \Psi_{i,j-1}^* - 4\Psi_{i,j}^*}{h^{*2}} \quad (133)$$

Para la discretización de la ecuación (71) se utilizan las fórmulas (118), (119), (120) y (121),

$$= Re \left[ \frac{\xi_{i+1,j}^* + \xi_{i-1,j}^* + \xi_{i,j+1}^* + \xi_{i,j-1}^* - 4\xi_{i,j}^*}{h^{*2}} \left( \frac{(\Psi_{i,j+1}^* - \Psi_{i,j-1}^*)(\xi_{i+1,j}^* + \xi_{i-1,j}^*)}{2h^*} - \frac{(\Psi_{i+1,j}^* - \Psi_{i-1,j}^*)(\xi_{i,j+1}^* - \xi_{i,j-1}^*)}{2h^*} \right) \right] \quad (134)$$

Así, se han obtenido dos ecuaciones que se pueden reordenar para obtener sendas ecuaciones para la función de corriente y la vorticidad en el punto de malla  $(i, j)$ .

En la ecuación (134) no aparece la función de corriente en el punto de malla  $(i, j)$ , por lo que para obtener la función de corriente en dicho punto se ha de reordenar la ecuación (133),

$$\Psi_{i,j}^* = \frac{\Psi_{i+1,j}^* + \Psi_{i-1,j}^* + \Psi_{i,j+1}^* + \Psi_{i,j-1}^* - h^{*2}\xi_{i,j}^*}{4} \quad (135)$$

De la ecuación (134) se puede obtener una expresión para la vorticidad en el punto de malla  $(i, j)$ :

$$\xi_{i,j}^* = \frac{\xi_{i+1,j}^* + \xi_{i-1,j}^* + \xi_{i,j+1}^* + \xi_{i,j-1}^*}{4} - \frac{Re}{16} \left[ (\Psi_{i,j+1}^* - \Psi_{i,j-1}^*)(\xi_{i+1,j}^* + \xi_{i-1,j}^*) - (\Psi_{i+1,j}^* - \Psi_{i-1,j}^*)(\xi_{i,j+1}^* - \xi_{i,j-1}^*) \right] \quad (136)$$

Por último, falta por discretizar la ecuación (72) utilizando las fórmulas (120), (121) y (122)

$$p_{i+1,j}^* + p_{i-1,j}^* + p_{i,j+1}^* + p_{i,j-1}^* - 4p_{i,j}^* = \frac{2}{h^{*2}Eu} \left[ (\Psi_{i+1,j}^* + \Psi_{i-1,j}^* - 2\Psi_{i,j}^*)(\Psi_{i,j+1}^* + \Psi_{i,j-1}^* - 2\Psi_{i,j}^*) - \frac{1}{16} (\Psi_{i+1,j+1}^* - \Psi_{i+1,j-1}^* - \Psi_{i-1,j+1}^* + \Psi_{i-1,j-1}^*)^2 \right] \quad (137)$$

que reordenada conduce a la expresión de la presión en el punto de malla  $(i, j)$ :

$$p_{i,j}^* = \frac{p_{i+1,j}^* + p_{i-1,j}^* + p_{i,j+1}^* + p_{i,j-1}^*}{4} - \frac{1}{2h^{*2}Eu} \left[ (\Psi_{i+1,j}^* + \Psi_{i-1,j}^* - 2\Psi_{i,j}^*)(\Psi_{i,j+1}^* + \Psi_{i,j-1}^* - 2\Psi_{i,j}^*) - \frac{1}{16} (\Psi_{i+1,j+1}^* - \Psi_{i+1,j-1}^* - \Psi_{i-1,j+1}^* + \Psi_{i-1,j-1}^*)^2 \right] \quad (138)$$

Una vez discretizadas las ecuaciones se discretizan las condiciones de contorno. Como criterio general, en los casos en que existan dos posibles condiciones de contorno, se escogerá la condición de Dirichlet, ya que su aplicación es inmediata y, en el caso de no existir una condición de Dirichlet, se escogerá la condición de Neumann. En el apartado 3.2 se definieron ambos tipos de condiciones de contorno.

A. Contorno A. 

Para la función de corriente las dos posibles condiciones de contorno vienen dadas por las ecuaciones (75) y (76). Sea  $(i, j)$  un punto del contorno A, la ecuación (75) se puede discretizar utilizando diferencias finitas centradas, pero no se obtendría un valor de la función en el punto de malla  $(i, j)$ , por lo que es conveniente utilizar diferencias finitas adelantadas o atrasadas. Para discretizar la ecuación (76) solo es posible aplicar diferencias finitas adelantadas, ya que el punto  $(i - 1, j)$  se encontraría fuera de la malla y se desconoce su valor. Por ello, utilizando la fórmula adelantada (124) para discretizar la ecuación (75) y la fórmula adelantada (123) para discretizar la ecuación (76) se tiene que

$$\Psi_{i,j}^*|_A = \Psi_{i,j+1}^*|_A - h^* \quad (139)$$

$$\Psi_{i,j}^*|_A = \Psi_{i+1,j}^*|_A \quad (140)$$

Así, se han obtenido dos ecuaciones válidas para la función de corriente del contorno A. Sin embargo, solo es necesario aplicar una.

En el contorno A, no existe una condición de Dirichlet para la función de corriente adimensionalizada, pero la ecuación de contorno (76) es de Neumann, por lo que en este caso, se utilizará la ecuación (140). Hay que tener en cuenta que si se aplicase la ecuación (139), la condición de contorno no se extendería al interior del dominio, ya que los puntos  $(i, j)$  e  $(i, j + 1)$  pertenecen al contorno A.

Para la vorticidad la discretización de la ecuación (77) es simplemente

$$\xi_{i,j}^*|_A = 0 \quad (141)$$

Para la presión se discretiza la ecuación (78), al ser esta una condición de contorno de Neumann. Para ello, se deben utilizar las fórmulas (123) y (127) en diferencias finitas adelantadas, lo que conduce a

$$p_{i,j}^*|_A = p_{i+1,j}^*|_A + \frac{1}{h^* Eu} (\Psi_{i+1,j+1}^*|_A - \Psi_{i+1,j}^*|_A - \Psi_{i,j+1}^*|_A + \Psi_{i,j}^*|_A) \quad (142)$$

B. Contorno B. 

Para el contorno B las dos posibles ecuaciones de contorno para la función de corriente vienen determinadas por las ecuaciones (80) y (81). Para discretizar la ecuación (80) es necesario utilizar diferencias finitas atrasadas, para lo que se emplea la fórmula (129). Para discretizar la ecuación (81) es conveniente utilizar diferencias finitas adelantadas o atrasadas, con el fin de obtener la función de corriente en el punto de malla  $(i, j)$ . La ecuación (144) se obtiene utilizando la fórmula (123).

$$\Psi_{i,j}^*|_B = \Psi_{i,j-1}^*|_B + h^* \quad (143)$$

$$\Psi_{i,j}^*|_B = \Psi_{i+1,j}^*|_B \quad (144)$$

En este caso, la condición que se selecciona es la (143), al ser una condición de contorno de Neumann, ya que mide la variación en la dirección normal a la superficie. La ecuación (144) no

aporta información del contorno al interior del dominio de resolución, por lo que no sería adecuada.

Para la vorticidad, la discretización de la ecuación (82) es simplemente

$$\xi_{i,j}^*|_B = 0 \quad (145)$$

Para la presión basta con discretizar la ecuación (84), que utilizando la fórmula (129) conduce a

$$p_{i,j}^*|_B = p_{i,j-1}^*|_B \quad (146)$$

C. Contorno C. 

Siguiendo el mismo tipo de razonamiento, para la función de corriente la discretización de la ecuación (85) debe realizarse con diferencias finitas atrasadas, por lo que utilizando la fórmula (128) se tiene que

$$\Psi_{i,j}^*|_C = \Psi_{i-1,j}^*|_C \quad (147)$$

Para la vorticidad se eligieron dos posibles condiciones de contorno, definidas por las ecuaciones (86) y (87), que discretizadas mediante las fórmulas (128) y (130) se transforman en

$$\xi_{i,j}^*|_C = \xi_{i-1,j}^*|_C \quad (148)$$

$$\xi_{i,j}^*|_C = 2\xi_{i-1,j}^*|_C - \xi_{i-2,j}^*|_C \quad (149)$$

Para la presión, la discretización de la ecuación de contorno de tipo Neumann, ecuación (88), puede realizarse utilizando diferencias finitas atrasadas en la presión y diferencias finitas centradas en la vorticidad, mediante las fórmulas (128) y (119), obteniéndose

$$p_{i,j}^*|_C = p_{i-1,j}^*|_C + \frac{1}{ReEu} \frac{(\xi_{i,j+1}^*|_C - \xi_{i,j-1}^*|_C)}{2} \quad (150)$$

D. Contorno D. 

Para la función de corriente y la vorticidad la discretización de las ecuaciones (90) y (91) es simplemente

$$\Psi_{i,j}^*|_D = 0 \quad (151)$$

$$\xi_{i,j}^*|_D = 0 \quad (152)$$

Para la presión, la discretización de la ecuación (93) requiere el uso de la fórmula (124) en diferencias finitas adelantadas

$$p_{i,j}^*|_D = p_{i,j+1}^*|_D \quad (153)$$

E. Contorno E. 

La discretización de las ecuaciones (94), (95) y (97) requiere del uso de la fórmula (124) en diferencias finitas adelantadas para la presión, obteniéndose

$$\Psi_{i,j}^*|_E = 0 \quad (154)$$

$$\xi_{i,j}^*|_E = 0 \quad (155)$$

$$p_{i,j}^*|_E = p_{i,j+1}^*|_E \quad (156)$$

F. Contorno F. 

La discretización de la ecuación (98) conduce a,

$$\Psi_{i,j}^*|_F = 0 \quad (157)$$

Para la discretizar la ecuación (99) de la vorticidad se va a considerar una capa de fluido muy cercana al contorno ( $h^{*2} \rightarrow 0$ ). Mediante un desarrollo de Taylor en dos variables se puede obtener la función de corriente en dicha capa

$$\Psi_{i-1,j}^*|_F = \Psi_{i,j}^*|_F - h^* \frac{\partial \Psi^*}{\partial x^*}|_F + \frac{h^{*2}}{2} \frac{\partial^2 \Psi^*}{\partial x^{*2}}|_F + \dots \quad (158)$$

Pero, por la condición “no-slip”, en la pared del obstáculo la velocidad del fluido debe ser la misma que la del obstáculo y, como el obstáculo se encuentra en reposo, se tiene que

$$\frac{\partial \Psi^*}{\partial x^*}|_F = 0 \quad (159)$$

Por tanto, despreciando los términos de orden superior a dos, si se sustituye la vorticidad en la ecuación (158) y teniendo en cuenta la ecuación (157) se tiene que

$$\xi_{i,j}^*|_F = 2 \frac{\Psi_{i-1,j}^*|_F}{h^{*2}} \quad (160)$$

Para la presión se discretiza la ecuación (104), mediante la fórmula (128) en diferencias finitas atrasadas para la presión y la fórmula (119) en diferencias finitas centradas para la vorticidad

$$p_{i,j}^*|_F = p_{i-1,j}^*|_F + \frac{1}{ReEu} \frac{(\xi_{i,j+1}^*|_F - \xi_{i,j-1}^*|_F)}{2} \quad (161)$$

G. Contorno G. 

Para la función de corriente la discretización de la ecuación (106) es

$$\Psi_{i,j}^*|_G = 0 \quad (162)$$

Para discretizar la ecuación (103) de la vorticidad se vuelve a considerar una capa fluida muy cercana al obstáculo, de tal manera que en dicha capa el desarrollo de Taylor será

$$\Psi_{i,j+1}^*|_G = \Psi_{i,j}^*|_G + h^* \frac{\partial \Psi^*}{\partial y^*}|_G + \frac{h^{*2}}{2} \frac{\partial^2 \Psi^*}{\partial y^{*2}}|_G + \dots \quad (163)$$

que teniendo en cuenta la condición “no slip” y la ecuación (162), conduce a obtener la vorticidad

$$\xi_{i,j}^*|_G = 2 \frac{\Psi_{i,j+1}^*|_G}{h^{*2}} \quad (164)$$

Para la presión se discretiza la ecuación (105), mediante la fórmula (124) en diferencias finitas adelantadas para la presión y la fórmula (118) en diferencias finitas centradas para la vorticidad

$$p_{i,j}^*|_G = p_{i,j+1}^*|_G + \frac{1}{ReEu} \frac{(\xi_{i+1,j}^*|_G - \xi_{i-1,j}^*|_G)}{2} \quad (165)$$

H. Contorno H. 

Para la función de corriente la discretización de la ecuación (106) es

$$\Psi_{i,j}^*|_H = 0 \quad (166)$$

Para discretizar la ecuación (107) de la vorticidad se vuelve a considerar una capa fluida muy cercana al obstáculo, de tal manera que en dicha capa

$$\Psi_{i+1,j}^*|_H = \Psi_{i,j}^*|_H + h^* \frac{\partial \Psi^*}{\partial x^*}|_H + \frac{h^{*2}}{2} \frac{\partial^2 \Psi^*}{\partial x^{*2}}|_H + \dots \quad (167)$$

Que teniendo en cuenta la condición “no slip” y la ecuación (166), conduce a obtener la vorticidad

$$\xi_{i,j}^*|_H = 2 \frac{\Psi_{i+1,j}^*|_H}{h^{*2}} \quad (168)$$

Para la presión se discretiza la ecuación (108), mediante la fórmula (123) en diferencias finitas adelantadas para la presión y la fórmula (119) en diferencias finitas centradas para la vorticidad

$$p_{i,j}^*|_H = p_{i+1,j}^*|_H - \frac{1}{ReEu} \frac{(\xi_{i,j+1}^*|_H - \xi_{i,j-1}^*|_H)}{2} \quad (169)$$

La discretización de las condiciones de contorno es algo más complicada. En el apartado 4.1 se detallaron las condiciones de contorno a aplicar en los distintos contornos. A la hora de discretizar dichas ecuaciones de contorno surge la duda de que ecuación escoger en los puntos de intersección entre los contornos. En lo que sigue se va a detallar su elección, detallando las ecuaciones discretizadas a utilizar en cada intersección.

I. Intersección de los contornos A y B. 

En el punto de intersección de los contornos A y B se podrían aplicar las condiciones del contorno A o las del contorno B.

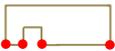
Para la función de corriente, en el punto de intersección,  $A \cap B$ , las ecuaciones (75) y (76) son las mismas que las ecuaciones (80) y (81). Al elegir la condición de contorno a imponer se escoge la más sencilla, que en este caso es la ecuación (76), cuya forma discretizada está dada por la ecuación (140).

La vorticidad en ambos contornos es cero, por lo que en la intersección de los contornos también lo será. Por ello, se impondrá la ecuación discretizada (141).

Para la presión la elección es más complicada, ya que en cada contorno se cumplen unas condiciones de contorno distintas. Sin embargo, al comparar la ecuación (142) del contorno A con la ecuación (146) del contorno B, se ve que ésta última tiene menos términos, por lo que se cometerá un menor error numérico en su cálculo. Por ello, será la ecuación (146) la que se imponga.

J. Intersección de los contornos B y C. 

Para la función de corriente se impondrá la ecuación (147). Para la vorticidad la condición impuesta en el contorno B es de Dirichlet, por lo que será la ecuación (145) que se impondrá en la intersección. Para la presión se aplicará la ecuación (146) del contorno B.

K. Intersección de los contornos en el eje de simetría. ( $C \cap E$ ,  $D \cap A$ ,  $D \cap F$  y  $E \cap H$ ) 

Todos estos puntos de intersección pertenecen al eje de simetría y, por tanto, a la línea de corriente. En este caso, tanto la función de corriente como la vorticidad deben anularse. Por ello, en estos puntos se impondrá

$$\Psi_{i,j}^*|_{C \cap E} = \Psi_{i,j}^*|_{D \cap A} = \Psi_{i,j}^*|_{D \cap F} = \Psi_{i,j}^*|_{E \cap H} = 0 \quad (170)$$

$$\xi_{i,j}^*|_{C \cap E} = \xi_{i,j}^*|_{D \cap A} = \xi_{i,j}^*|_{D \cap F} = \xi_{i,j}^*|_{E \cap H} = 0 \quad (171)$$

Para la presión aplicaremos la ecuación (153).

L. Intersección de los contornos F y G. 

Todo el contorno del obstáculo es una línea de corriente. Por ello, la función de corriente se anula en este punto y se impondrá la ecuación (157).

Para la vorticidad la situación es bastante más complicada, ya que los contornos F y G cumplen dos condiciones “no-slip” distintas. Se ha optado por imponer una condición de contorno combinación lineal de las condiciones definidas por las ecuaciones (160) y (164),

$$\xi_{i,j}^*|_{F \cap G} = \frac{\Psi_{i-1,j}^*|_{F \cap G} + \Psi_{i,j+1}^*|_{F \cap G}}{h^{*2}} \quad (172)$$

Para la presión se pueden imponer las ecuaciones (161) y (165). Introducir una condición mezcla sería demasiado complicado, por lo que se ha optado por imponer la condición del contorno F, dada por la ecuación (161).

M. Intersección de los contornos G y H. 

Para esta intersección de los contornos G y H se procederá de manera análoga a la empleada en la intersección de los contornos F y G. La discretización de la función de corriente estará dada por la ecuación (162).

Para la vorticidad se impondrá una condición que será mezcla entre las ecuaciones discretizadas (164) y (168), en particular

$$\xi_{i,j}^*|_{G \cap H} = \frac{\Psi_{i+1,j}^*|_{G \cap H} + \Psi_{i,j+1}^*|_{G \cap H}}{h^{*2}} \quad (173)$$

Y para la presión la condición impuesta será la del contorno H, (169).

## 5. Método de resolución

Una vez realizadas las discretizaciones de las distintas ecuaciones diferenciales y condiciones de contorno es preciso describir el método de resolución del problema. En el presente trabajo se ha optado por utilizar un método iterativo, basado en la aplicación de un algoritmo que, a partir de una aproximación a la solución del problema busca converger a la solución numérica de este.

Estos métodos son comunes en la resolución de problemas físicos con un gran número de variables y están asociados frecuentemente a la resolución de ecuaciones diferenciales en derivadas parciales. El mayor problema que tienen es que no garantizan la convergencia.

Para simplificar la notación, a partir de este momento se va a prescindir de los asteriscos en las magnitudes adimensionalizadas, pero es importante recordar que se está trabajando con ellas.

Como criterio de convergencia para detener el método iterativo se define un criterio de tolerancia. Así, se considerará que se ha alcanzado una solución estacionaria satisfactoria cuando se cumplan a la vez las dos siguientes condiciones

$$TOL > \frac{\max |\Psi_{i,j}^{(k+1)} - \Psi_{i,j}^{(k)}|}{\max |\Psi_{i,j}^{(k+1)}|} \quad (174)$$

$$TOL > \frac{\max |\xi_{i,j}^{(k+1)} - \xi_{i,j}^{(k)}|}{\max |\xi_{i,j}^{(k+1)}|} \quad (175)$$

donde  $k$  y  $k + 1$  representan dos iteraciones consecutivas.

Para el cálculo del campo de presión se utiliza el siguiente criterio de convergencia,

$$TOL > \frac{\max |p_{i,j}^{(k+1)} - p_{i,j}^{(k)}|}{\max |p_{i,j}^{(k+1)}|} \quad (176)$$

Como se vio en el apartado 3.1, las ecuaciones de la función corriente y la vorticidad están acopladas, y ello hace que la estrategia de resolución deba incluir el cálculo de ambas magnitudes, si no simultáneamente, si en la misma iteración. En cambio, la presión se encuentra desacoplada y, por ello, su cálculo puede realizarse una vez se han obtenido la función de corriente y la vorticidad. Así, el algoritmo secuencial utilizado para la resolución conjunta de las ecuaciones discretizadas (135) y (136) es el siguiente:

1. Se escogen unos valores de inicialización o partida para la función corriente y la vorticidad discretizadas. Una primera opción es escoger, para cada punto de malla  $(i, j)$ , los valores correspondientes al flujo libre

$$\Psi_{i,j}^{(0)} = y_j \quad \xi_{i,j}^{(0)} = 0 \quad (177)$$

donde el índice 0 indica las condiciones de arranque o partida del algoritmo secuencial.

Estos valores de partida son razonables para realizar las simulaciones de números de Reynolds bajos, pero como se verá se puede mejorar y acelerar la convergencia utilizando como valores de partida las soluciones numéricas obtenidas con números de Reynolds cercanos ( $Re \approx Re'$ ) al de interés.

$$\Psi_{i,j}^{(0)} = \Psi_{i,j}^{Re'} \quad \xi_{i,j}^{(0)} = \xi_{i,j}^{Re'} \quad (178)$$

donde  $\Psi_{i,j}^{Re'}$ ,  $\xi_{i,j}^{Re'}$  son las soluciones obtenidas con un número de Reynolds  $Re'$ , próximo al número de Reynolds  $Re$  de interés.

2. Se realiza una primera iteración de la función corriente para obtener un mejor aproximación de esta,  $\Psi_{i,j}^{(1)}$ , en todos los nodos de la malla, a partir de la ecuación (135):<sup>1</sup>

$$\Psi_{i,j}^{(1)} = \frac{\Psi_{i+1,j}^{(0)} + \Psi_{i-1,j}^{(0)} + \Psi_{i,j+1}^{(0)} + \Psi_{i,j-1}^{(0)} - h^2 \xi_{i,j}^{(0)}}{4} \quad (179)$$

Esta aproximación se calcula mediante el método de Jacobi, es decir, se utilizarán los valores de la función de corriente en la anterior iteración para obtener una nueva aproximación en cada punto de malla. En un principio parece más interesante utilizar Gauss-Seidel, ya que al ir utilizando los valores aproximados permitiría acelerar la convergencia. Sin embargo, en el problema que se está tratando este método no converge. Esto puede deberse a que en el método de Gauss-Seidel las condiciones para garantizar la convergencia son más restrictivas que en el método de Jacobi [6].

Además, con el fin de facilitar la convergencia se introducirá un método de relajación, donde  $w_\Psi$  será el parámetro de relajación para la función de corriente. Así, en la primera iteración se tendrá que

$$\Psi_{i,j}^{(1)'} = w_\Psi \Psi_{i,j}^{(1)} + (1 - w_\Psi) \Psi_{i,j}^{(0)} \quad (180)$$

3. Se realiza el cálculo de la función vorticidad utilizando  $\Psi_{i,j}^{(1)'}$  y  $\xi_{i,j}^{(0)}$  para obtener  $\xi_{i,j}^{(1)}$ , a partir de la ecuación (136):<sup>1</sup>

$$\xi_{i,j}^{(1)} = \frac{\xi_{i+1,j}^{(0)} + \xi_{i-1,j}^{(0)} + \xi_{i,j+1}^{(0)} + \xi_{i,j-1}^{(0)}}{4} \quad (181)$$

$$- \frac{Re}{16} \left[ \left( \Psi_{i,j+1}^{(1)'} - \Psi_{i,j-1}^{(1)'} \right) \left( \xi_{i+1,j}^{(0)} + \xi_{i-1,j}^{(0)} \right) - \left( \Psi_{i+1,j}^{(1)'} - \Psi_{i-1,j}^{(1)'} \right) \left( \xi_{i,j+1}^{(0)} - \xi_{i,j-1}^{(0)} \right) \right]$$

En este caso se utilizan los valores de la función de corriente discretizada calculados en la iteración actual.

Para la vorticidad también se utiliza en el cálculo un método de relajación, de manera que

<sup>1</sup> Es importante recordar que, a pesar de prescindir en la notación de los asteriscos en las magnitudes adimensionalizadas, se está trabajando con ellas.

$$\xi_{i,j}^{(1)'} = w_\xi \xi_{i,j}^{(1)} + (1 - w_\xi) \xi_{i,j}^{(0)} \quad (182)$$

donde  $w_\xi$  será el parámetro de relajación para la vorticidad.

4. Se comprueba si se cumple el criterio de convergencia establecido. Si no es así se vuelven a realizar los pasos 2 y 3 las iteraciones necesarias hasta que se cumple el criterio de convergencia. Una vez cumplido se habrán alcanzado las soluciones para la función de corriente y la vorticidad,  $\Psi_{i,j}^{(sol)}$  y  $\xi_{i,j}^{(sol)}$ , y se podrá realizar el cálculo de la presión.

Para calcular la presión se realiza el siguiente algoritmo secuencial usando la ecuación discretizada (138).

1. El primer paso consiste en elegir un valor de partida para la presión, que podrá ser el mismo en todos los puntos de malla

$$p_{i,j}^{(0)} = 0 \quad (183)$$

También podrá ser la presión previamente calculada para un número de Reynolds cercano ( $Re \approx Re'$ ).

$$p_{i,j}^{(0)} = p_{i,j}^{Re'} \quad (184)$$

2. Se realiza una primera iteración para la presión a partir de la ecuación discretizada (138)

$$p_{i,j}^{(1)} = \frac{p_{i+1,j}^{(0)} + p_{i-1,j}^{(0)} + p_{i,j+1}^{(0)} + p_{i,j-1}^{(0)}}{4} - \frac{1}{2h^2 Eu} \left[ (\Psi_{i+1,j}^{(sol)} - \Psi_{i-1,j}^{(sol)} - 2\Psi_{i,j}^{(sol)}) (\Psi_{i,j+1}^{(sol)} - \Psi_{i,j-1}^{(sol)} - 2\Psi_{i,j}^{(sol)}) - \frac{1}{16} (\Psi_{i+1,j+1}^{(sol)} - \Psi_{i-1,j+1}^{(sol)} - \Psi_{i-1,j+1}^{(sol)} + \Psi_{i-1,j-1}^{(sol)})^2 \right] \quad (185)$$

3. Se realiza una relajación de la presión,

$$p_{i,j}^{(1)'} = w_p p_{i,j}^{(1)} + (1 - w_p) p_{i,j}^{(0)} \quad (186)$$

donde  $w_p$  representa es el parámetro de relajación de la presión.

4. Se comprueba el criterio de convergencia para la presión y, si no se cumple, se itera las veces necesarias hasta cumplirlo, para obtener la solución numérica de la presión  $p_{i,j}^{(sol)}$ .

En un principio se ha explicado el algoritmo con tres parámetros de relajación independientes, pero con el fin de simplificar el problema, se considerará un único parámetro de relajación, de manera que  $w_\Psi = w_\xi = w_p = w$ . Este parámetro es un factor ponderado que tiene un valor entre 0 y 2.

Habitualmente la relajación se emplea como mejora al método de Gauss-Seidel, ya que tomar valores de  $w$  entre 1 y 2 (sobrerrelajación) permite acelerar en gran medida la convergencia.

Por otro lado, en casos en que la convergencia es complicada, la utilización del método de relajación sigue siendo muy importante, ya que la utilización de valores de  $w$  entre 0 y 1 (subrelajación) puede también facilitar la convergencia. El efecto negativo de la subrelajación es que puede producir una ralentización de la convergencia [6].

## 6. Resultados

Para presentar los resultados obtenidos en la resolución numérica es necesario previamente realizar un análisis de los distintos parámetros que van a afectar a la resolución del problema.

El tamaño del obstáculo viene definido por su largo y su alto, como reflejan las ecuaciones (111) y (112). El ancho del obstáculo, es fijo y de valor 1, ya que las unidades del sistema están referidas a él. El alto de objeto se va a fijar en  $L_y^* = 1/2$ , con el fin de utilizar un mismo obstáculo para todas las simulaciones. Con esta elección, el obstáculo será cuadrado.

El tamaño de la malla conviene escogerlo a partir de su largo,  $L^*$ , y su alto,  $H^*$ , de tal manera que pueda modificarse el paso de red, sin por ello modificar las dimensiones del problema. Además, por simplicidad se elegirán valores de  $L^*$  y  $H^*$  enteros.

Una vez seleccionado el tamaño de la malla queda por fijar la posición del obstáculo. Es interesante situarlo en una posición más cercana al contorno A que al contorno C, con el fin de observar el comportamiento del flujo corriente abajo del obstáculo. Se ha escogido colocar el contorno F del obstáculo en la posición

$$x^*_{obstáculo} = \frac{L^*}{3} \quad (187)$$

El paso de red será el parámetro fundamental a la hora de establecer un compromiso entre el tiempo de computación necesario para la convergencia requerida y el detalle de resolución del problema. A partir de las ecuaciones (114) y (115) se puede obtener una relación entre el número de puntos donde se calcularán los distintos campos y los parámetros de la red.

$$N_{total} = N_x N_y = \frac{L^* H^*}{h^{*2}} \quad (188)$$

Con el fin de poder comparar las distintas simulaciones que se van a realizar en los siguientes apartados, se utilizarán en todas las mismas condiciones de contorno. Así, para la vorticidad en el contorno C se utilizará la ecuación de contorno (149).

En el apartado 6.1 se va a analizar el comportamiento físico del sistema para tres números de Reynolds, 0.1, 1 y 16, que representan tres regímenes distintos.

El análisis del campo de presión se realizará en el apartado 6.2.

En el apartado 6.3 se analizará el efecto de la resolución, para un número de Reynolds fijo, en los resultados de las simulaciones y en el tiempo de computación.

Para ver el efecto del ancho y el alto de la malla, en el apartado 6.4 se compararán los resultados obtenidos para distintas mallas.

Para ver el efecto de las condiciones de contorno, en el apartado 6.5 se analizará el efecto de seleccionar distintas condiciones de contorno.

En el apartado 6.6 se analizará la importancia de la inicialización de los campos discretos en la convergencia del método iterativo.

Para comparar el método de resolución empleado con otro método numérico más avanzado, en el apartado 6.7 se hace una comparación con los resultados obtenidos al aplicar un método de Lattice Boltzmann [7].

Finalmente, en el apartado 6.8 se define el número de Reynolds crítico y se relaciona con el régimen estacionario y los resultados numéricos obtenidos.

Como extensión al trabajo se ha adaptado el código para reproducir los efectos de dos obstáculos, tal como se detallará en el apartado 6.9.

### 6.1. Análisis del comportamiento físico del sistema para $Re=0.1, 1$ y $16$

Se van a analizar los resultados para un flujo estacionario a distintos números de Reynolds, representando las líneas de corriente sobre la función de corriente, los contornos de vorticidad y el campo de velocidad. Se han escogido tres números de Reynolds, 0.1, 1 y 16, representativos de tres regímenes de comportamiento distintos. Para  $Re = 0.1$ , el comportamiento del fluido está regido principalmente por las fuerzas viscosas. Para  $Re = 1$  existe un balance entre las fuerzas inerciales y las fuerzas viscosas. Y para  $Re = 16$  son más importantes las fuerzas inerciales.

Para representar el campo de velocidad es necesario calcular previamente la velocidad en cada punto de la malla. Para ello se adimensionaliza la ecuación (10),

$$u^* = \frac{\partial \Psi^*}{\partial y^*}, \quad v^* = -\frac{\partial \Psi^*}{\partial x^*} \quad (189)$$

Y esta se discretiza mediante las fórmulas (118) y (119)

$$u_{i,j}^* = \frac{\Psi_{i,j+1}^* - \Psi_{i,j-1}^*}{2h^*}, \quad v_{i,j}^* = -\frac{\Psi_{i+1,j}^* - \Psi_{i-1,j}^*}{2h^*} \quad (190)$$

La representación de las líneas de corriente se realiza mediante el método “*streamplot*” del módulo “*matplotlib.pyplot*” del lenguaje de programación *Python*, a partir del campo de velocidad.

#### A. $Re = 0.1$ .

Para flujos a bajo números de Reynolds, los efectos viscosos son los predominantes, siendo prácticamente despreciables los efectos inerciales. De esta manera, el comportamiento del flujo es muy cercano al obtenido como solución del flujo de Stokes [8]. En el flujo de Stokes los términos no lineales de las ecuaciones de Navier-Stokes se desprecian y, por tanto, se desprecian los efectos inerciales. Esto trae como consecuencia la llamada reversibilidad cinemática o simetría de las líneas de corriente. En la Figura 6 se puede observar como las líneas de corriente son prácticamente simétricas respecto al plano  $x^* = 5,5$ , que divide al obstáculo por la mitad. Lo mismo se puede afirmar de los contornos de vorticidad en la Figura 7 o del módulo de la velocidad en la Figura 8.

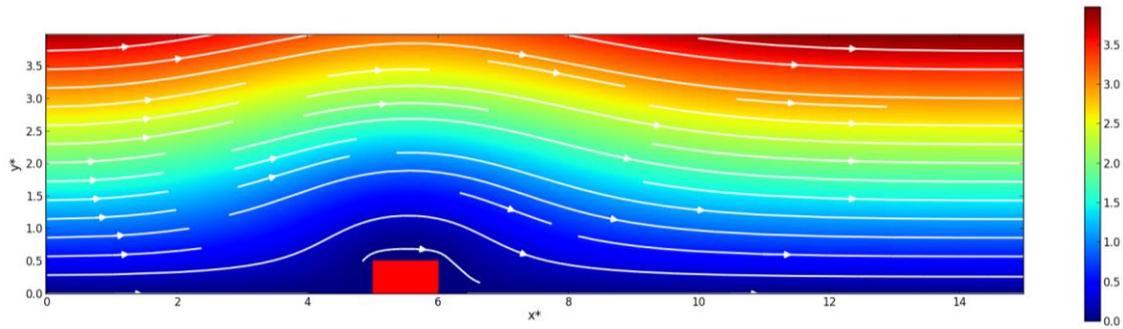


Figura 6. Función de corriente para  $Re = 0.1$ . En blanco se representan las líneas de corriente. Los parámetros de la simulación son  $L^* = 15$ ,  $H^* = 4$ , con un obstáculo de tamaño  $L_x^* = 1$ ,  $L_y^* = 1/2$ , un paso de red  $h^* = 1/32$  y una tolerancia  $TOL = 10^{-5}$ .

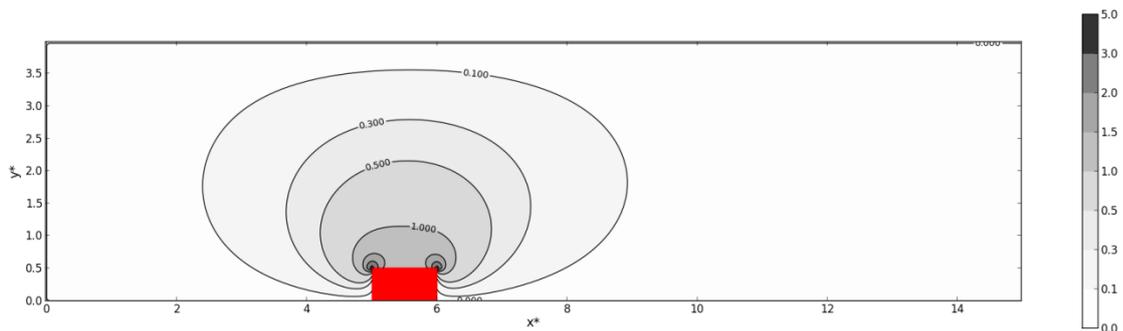


Figura 7. Líneas de contorno de la vorticidad para  $Re = 0.1$ . Los parámetros de la simulación son  $L^* = 15$ ,  $H^* = 4$ , con un obstáculo de tamaño  $L_x^* = 1$ ,  $L_y^* = 1/2$ , un paso de red  $h^* = 1/32$  y una tolerancia  $TOL = 10^{-5}$ .

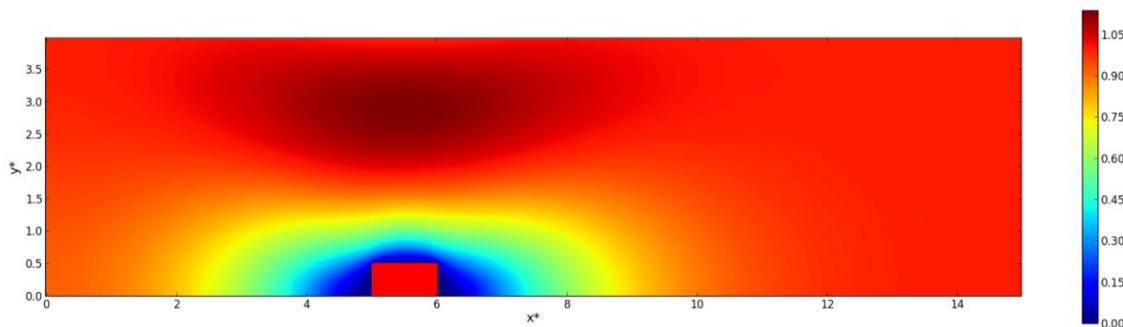


Figura 8. Módulo de la velocidad para  $Re = 0.1$ . Los parámetros de la simulación son  $L^* = 15$ ,  $H^* = 4$ , con un obstáculo de tamaño  $L_x^* = 1$ ,  $L_y^* = 1/2$ , un paso de red  $h^* = 1/32$  y una tolerancia  $TOL = 10^{-5}$ .

En la Figura 9 se puede apreciar como existen dos zonas del mismo tamaño, situadas antes y después del obstáculo, en las que el fluido se encuentra prácticamente en reposo.

Lógicamente en el flujo estudiado las fuerzas inerciales no han sido despreciadas, por lo que haciendo un análisis en detalle en la zona cercana al objeto, en este caso a través de los vectores velocidad (Figura 9), se puede observar cómo en realidad el flujo no es estrictamente simétrico.

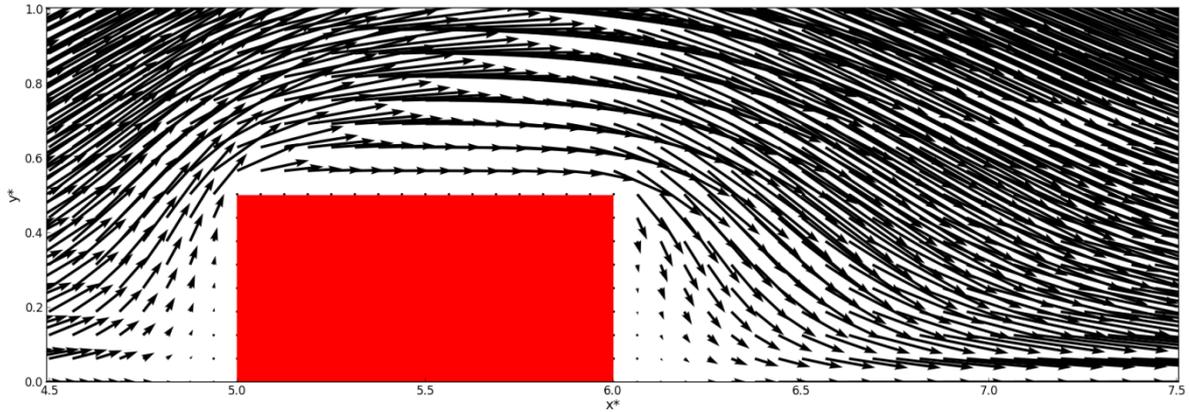


Figura 9. Vectores velocidad cerca del obstáculo para  $Re = 0.1$ . La longitud de los vectores representa su magnitud. Los parámetros de la simulación son  $L^* = 15$ ,  $H^* = 4$ , con un obstáculo de tamaño  $L_x^* = 1$ ,  $L_y^* = 1/2$ , un paso de red  $h^* = 1/32$  y una tolerancia  $TOL = 10^{-5}$ . Con el fin de mantener la figura inteligible solo se representa el campo de velocidades en la mitad de los puntos de la malla.

#### B. $Re = 1$ .

En este caso se observa en los campos de vorticidad (Figura 11) y de velocidad (Figura 12) la existencia de una asimetría con respecto del plano  $x^* = 5,5$ . En la Figura 11 se aprecia cómo se deforman las líneas de contorno de vorticidad corriente abajo del obstáculo. Así, la línea de contorno  $\xi^* = 0,1$  se extiende hasta pasado  $x^* = 10$ , mientras que para  $Re = 0.1$ , como se aprecia en la Figura 7, solo se extendía hasta  $x^* = 9$ . Esta asimetría se debe al efecto de las fuerzas inerciales. Para el campo función de corriente o para las líneas de corriente (Figura 10) dicha asimetría no es apreciable, lo que concuerda con los resultados obtenidos por Breuer et al. [9]

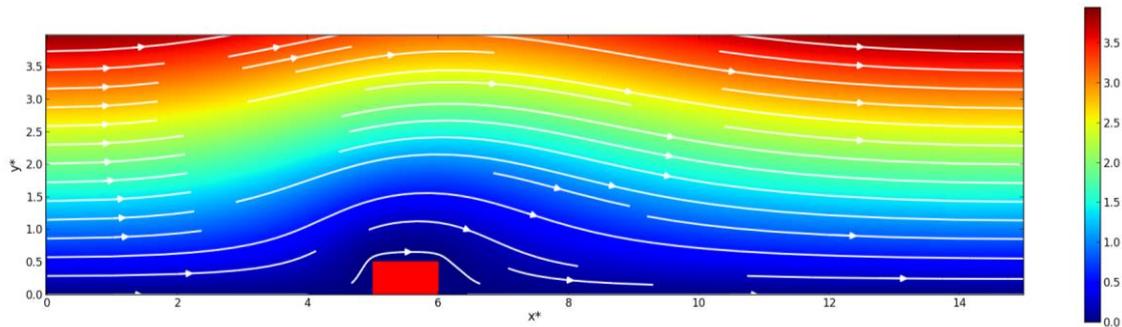


Figura 10. Función de corriente para  $Re = 1$ . En blanco se representan las líneas de corriente. Los parámetros de la simulación son  $L^* = 15$ ,  $H^* = 4$ , con un obstáculo de tamaño  $L_x^* = 1$ ,  $L_y^* = 1/2$ , un paso de red  $h^* = 1/32$  y una tolerancia  $TOL = 10^{-5}$ .

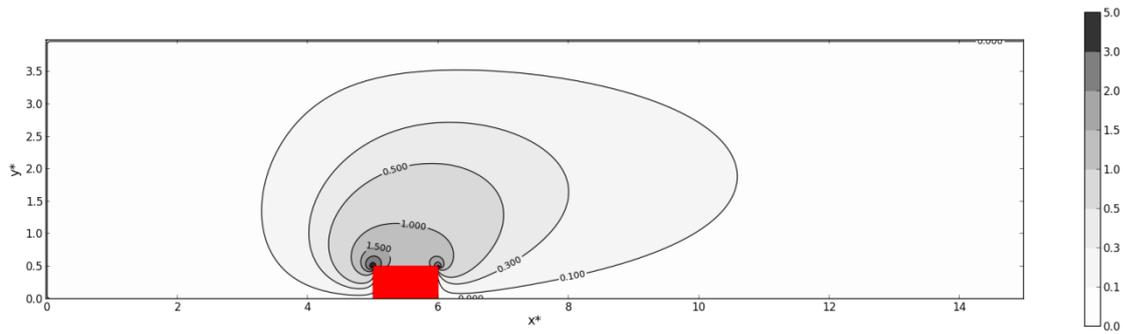


Figura 11. Líneas de contorno de la vorticidad para  $Re = 1$ . Los parámetros de la simulación son  $L^* = 15$ ,  $H^* = 4$ , con un obstáculo de tamaño  $L_x^* = 1$ ,  $L_y^* = 1/2$ , un paso de red  $h^* = 1/32$  y una tolerancia  $TOL = 10^{-5}$ .

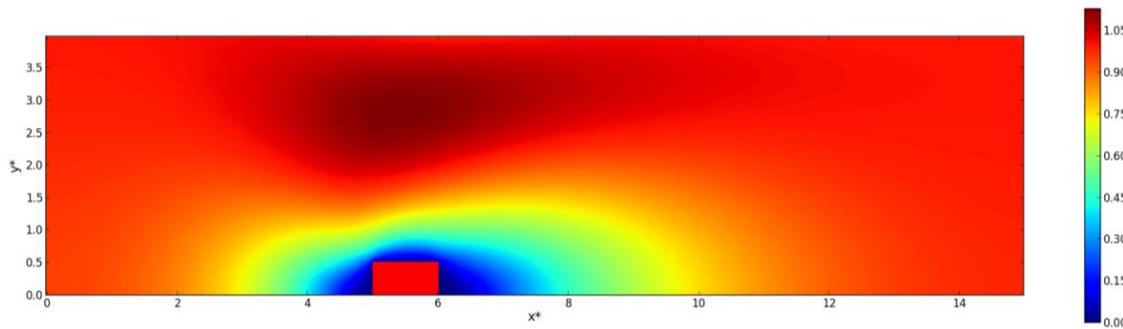


Figura 12. Módulo de la velocidad para  $Re = 1$ . Los parámetros de la simulación son  $L^* = 15$ ,  $H^* = 4$ , con un obstáculo de tamaño  $L_x^* = 1$ ,  $L_y^* = 1/2$ , un paso de red  $h^* = 1/32$  y una tolerancia  $TOL = 10^{-5}$ .

En la Figura 13 se puede apreciar que, al igual que para  $Re = 0.1$ , existen dos zonas en las que el fluido se encuentra prácticamente en reposo. Sin embargo, el tamaño de ambas zonas ya no es el mismo, siendo mayor la zona corriente abajo del obstáculo.

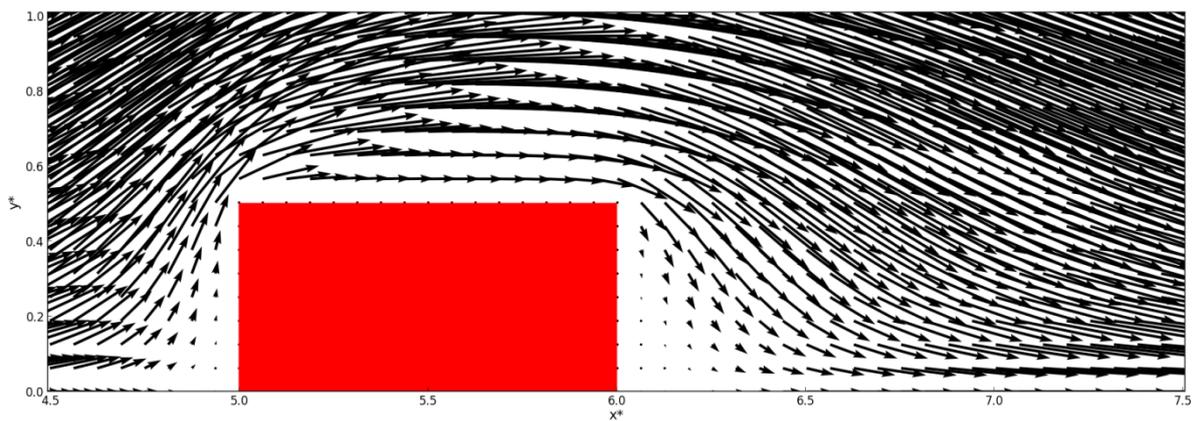


Figura 13. Vectores velocidad cerca del obstáculo para  $Re = 1$ . La longitud de los vectores representa su magnitud. Los parámetros de la simulación son  $L^* = 15$ ,  $H^* = 4$ , con un obstáculo de tamaño  $L_x^* = 1$ ,  $L_y^* = 1/2$ , un paso de red  $h^* = 1/32$  y una tolerancia  $TOL = 10^{-5}$ . Con el fin de mantener la figura inteligible solo se representa el campo de velocidades en la mitad de los puntos de la malla.

C.  $Re = 16.0$ .

En este caso las fuerzas inerciales tienen mayor importancia que las viscosas. Por ello, en las líneas de corriente de la Figura 14 se puede observar una circulación cerrada tras el obstáculo, en forma de vórtice.

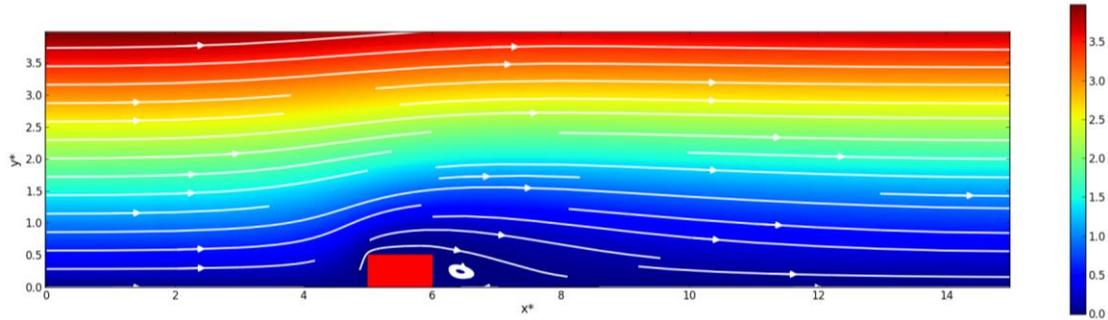


Figura 14. Función de corriente para  $Re = 16$ . En blanco se representan las líneas de corriente. Los parámetros de la simulación son  $L^* = 15$ ,  $H^* = 4$ , con un obstáculo de tamaño  $L_x^* = 1$ ,  $L_y^* = 1/2$ , un paso de red  $h^* = 1/32$  y una tolerancia  $TOL = 10^{-5}$ .

En la Figura 15 se aprecia que la vorticidad se propaga corriente abajo del obstáculo, no siendo posible determinar la extensión máxima del contorno de vorticidad  $\xi^* = 0,1$ . La forma de los contornos es consistente con los resultados obtenidos por Perumal et al. [10]

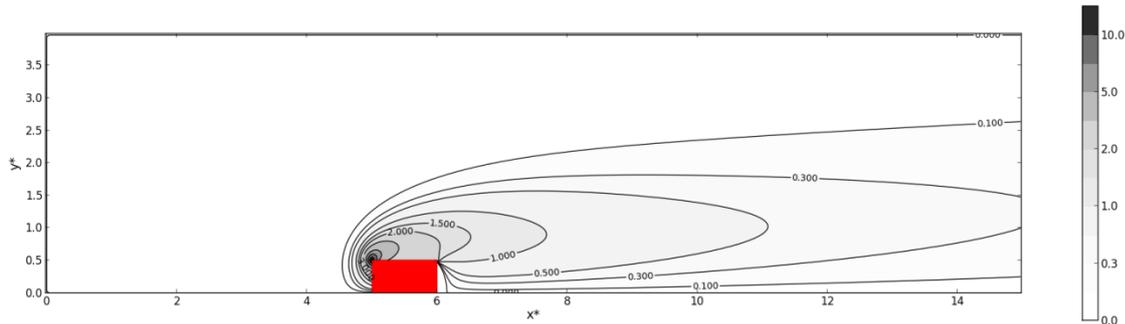


Figura 15. Líneas de contorno de la vorticidad para  $Re = 16$ . Los parámetros de la simulación son  $L^* = 15$ ,  $H^* = 4$ , con un obstáculo de tamaño  $L_x^* = 1$ ,  $L_y^* = 1/2$ , un paso de red  $h^* = 1/32$  y una tolerancia  $TOL = 10^{-5}$ .

En el campo del módulo de la velocidad, representado en la Figura 16, se aprecia la aparición de una estela corriente abajo del obstáculo, con velocidades claramente inferiores a las del flujo incidente.

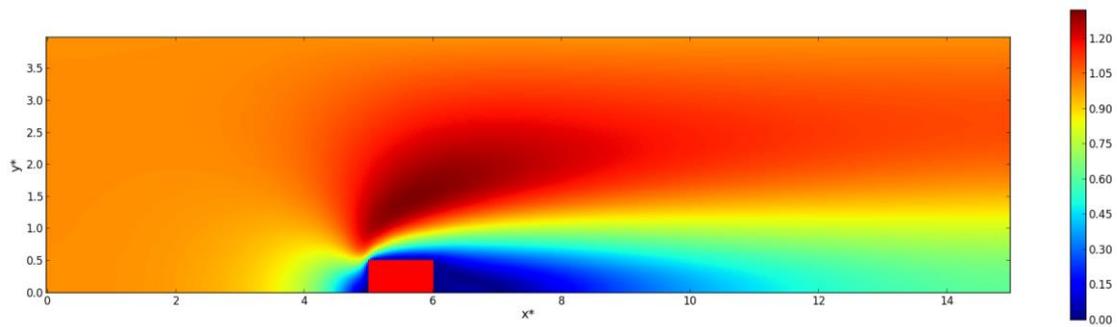


Figura 16. Módulo de la velocidad para  $Re = 16$ . Los parámetros de la simulación son  $L^* = 15$ ,  $H^* = 4$ , con un obstáculo de tamaño  $L_x^* = 1$ ,  $L_y^* = 1/2$ , un paso de red  $h^* = 1/32$  y una tolerancia  $TOL = 10^{-5}$ .

En la Figura 17 se aprecia claramente la circulación formada en la parte posterior al obstáculo. En este caso, el centro del vórtice se encuentra situado aproximadamente en la posición  $(x^* = 6,5, y^* = 0.25)$ , teniendo el vórtice aproximadamente el mismo largo que el obstáculo. Este largo se conoce como longitud de recirculación [9] y está relacionado con el número de Reynolds. El resultado obtenido está de acuerdo con los obtenidos en simulaciones más avanzadas por Breuer et al. [9] y Yoon et al. [11], si bien este último utiliza una definición algo distinta del parámetro de recirculación.

Las velocidades en éste vórtice tienen una magnitud mucho menor que la del flujo incidente sobre el obstáculo, como se puede apreciar en el módulo de la velocidad representado en la Figura 16 o en los vectores velocidad representados en la Figura 17.

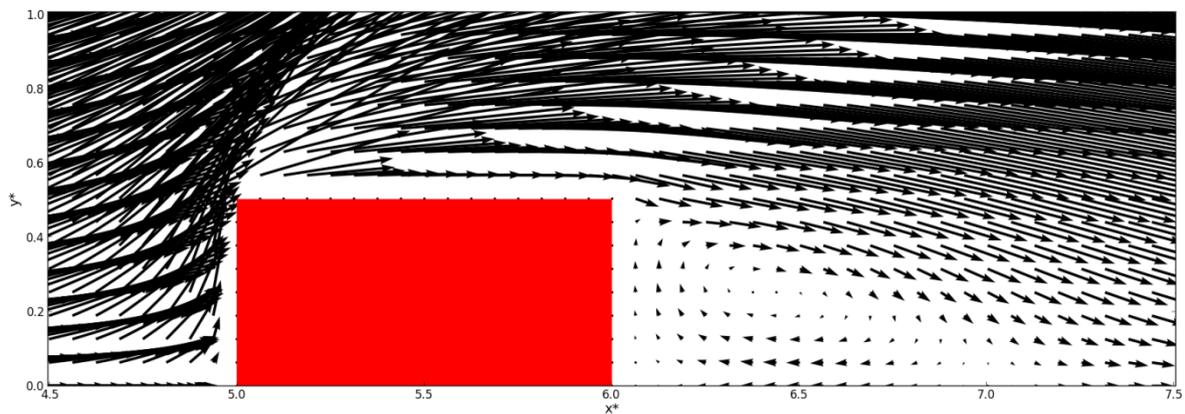


Figura 17. Vectores velocidad cerca del obstáculo para  $Re = 16$ . La longitud de los vectores representa su magnitud. Los parámetros de la simulación son  $L^* = 15$ ,  $H^* = 4$ , con un obstáculo de tamaño  $L_x^* = 1$ ,  $L_y^* = 1/2$ , un paso de red  $h^* = 1/32$  y una tolerancia  $TOL = 10^{-5}$ . Con el fin de mantener la figura inteligible solo se representa el campo de velocidades la mitad de los puntos de la malla.

## 6.2. Cálculo de la presión

En el apartado 5 ya se mencionó que no era necesario calcular el campo de presión para obtener el comportamiento del flujo. Sin embargo, no deja de ser interesante poder obtenerlo para complementar los resultados obtenidos.

El cálculo del campo de presión puede presentar un mayor error que el cálculo de los campos de función de corriente y vorticidad, ya que, como se puede ver en la ecuación (72), el campo de presión se calcula a partir de las derivadas segundas de la función de corriente, calculada previamente. Además, entre las condiciones de contorno de la presión no hay ninguna de tipo Dirichlet, por lo que no se puede imponer el valor de la presión en ningún contorno.

En este apartado se va a estudiar el campo de presión estacionario, que depende directamente del número de Euler, como indica la ecuación (74). El efecto de este número es cuantitativo, en el sentido de que va a modificar notablemente los valores de la presión que encontremos, pero no cualitativo, ya que los contornos de presión van a mantener la misma forma. Por ello, como ejemplo representativo, únicamente se van a representar las presiones obtenidas con el parámetro de Euler  $Eu = 1.0$ .

Además, la presión depende indirectamente del número de Reynolds, ya que depende de la función de corriente, como se puede ver en la ecuación (138). Por ello, se va a representar el campo de presión para los tres regímenes presentados en el apartado 6.1, es decir, para  $Re = 0.1, 1$  y  $16$ .

En la Figura 18 se representa el campo de presión para  $Eu = 1$  y  $Re = 0.1$ . Se puede apreciar que las variaciones de presión son muy importantes cerca del obstáculo. En concreto se observa un máximo en la esquina anterior del obstáculo y un mínimo, que tendrá un valor negativo, en la parte posterior del obstáculo. Además, en la parte posterior del obstáculo existe un pequeño máximo relativo de la presión, centrado aproximadamente en la posición  $(x^* = 8.0, y^* = 0.0)$ . Si no existiesen fuerzas inerciales (caso  $Re = 0$ ) existiría una simetría en los contornos de presión respecto al plano medio del obstáculo  $x^* = 5.5$ , tomando el campo de presión forma de dipolo. En la Figura 18 el campo de presión adopta una forma similar a dicho dipolo.

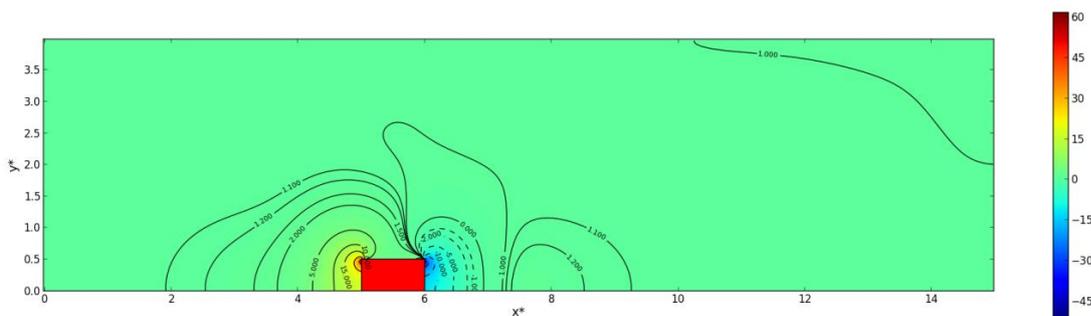


Figura 18. Campo de presión y contornos de presión para  $Eu = 1$  y  $Re = 0.1$ . Los parámetros de la simulación son  $L^* = 15, H^* = 4$ , con un obstáculo de tamaño  $L_x^* = 1, L_y^* = 1/2$ , un paso de red  $h^* = 1/32$  y una tolerancia  $TOL = 10^{-5}$ .

En la Figura 19 se puede apreciar que para  $Re = 1$  ya no existen valores de presión negativos, excepto en los puntos más cercanos al vértice posterior del obstáculo. Además, la forma del campo de presión ya no se asemeja a un dipolo, pero si se aprecia la formación de un máximo de presión relativo en la parte posterior del obstáculo.

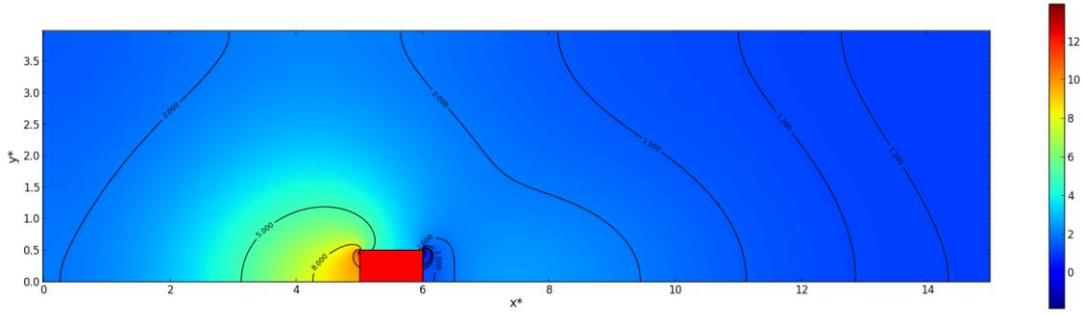


Figura 19. Campo de presión y contornos de presión para  $Eu = 1$  y  $Re = 1.0$ . Los parámetros de la simulación son  $L^* = 15$ ,  $H^* = 4$ , con un obstáculo de tamaño  $L_x^* = 1$ ,  $L_y^* = 1/2$ , un paso de red  $h^* = 1/32$  y una tolerancia  $TOL = 10^{-5}$ .

Por último, en la Figura 20, para un número de Reynolds  $Re = 16$ , se puede apreciar que el máximo de presión en toda la parte delantera del obstáculo es más acusado, mientras que en la parte trasera la presión ya no aparece un mínimo de presión ni un máximo relativo de presión.

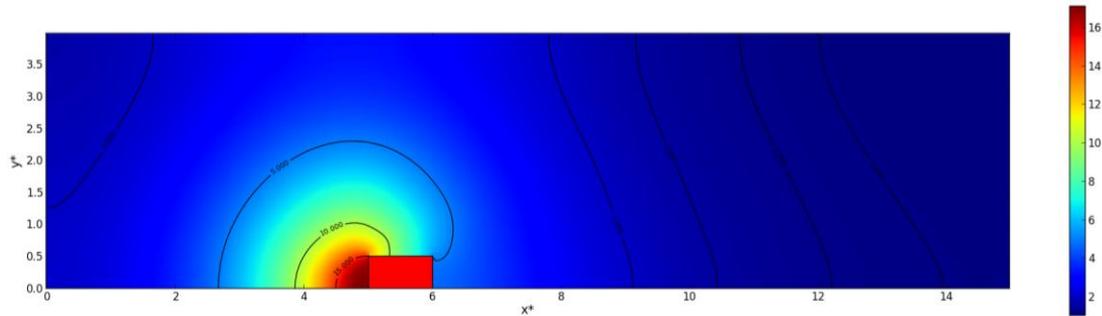


Figura 20. Campo de presión y contornos de presión para  $Eu = 1$  y  $Re = 16$ . Los parámetros de la simulación son  $L^* = 15$ ,  $H^* = 4$ , con un obstáculo de tamaño  $L_x^* = 1$ ,  $L_y^* = 1/2$ , un paso de red  $h^* = 1/32$  y una tolerancia  $TOL = 10^{-5}$ .

### 6.3. Efecto de la resolución

En los dos apartados anteriores se analizó el comportamiento del flujo para distintos números de Reynolds realizando distintas simulaciones con los mismos parámetros de la simulación. En este apartado se va a analizar el efecto producido por el aumento de la resolución, es decir, el efecto de disminuir el tamaño del paso de red para una misma caja física o problema.

Según la ecuación (188), el número de puntos de la simulación depende linealmente de  $L^*$  y  $H^*$ , y cuadráticamente e inversamente de  $h^*$ , de manera que la variación del paso de red afectará notablemente al tiempo de computación necesario para completar las simulaciones.

En la Tabla 1 se pueden observar los tiempos de computación necesarios para cumplir con el criterio de convergencia,  $TOL = 10^{-5}$ , utilizando una malla de dimensiones de  $L^* = 10$  y  $H^* = 4$ , un obstáculo de dimensiones  $L_x^* = 1$  y  $L_y^* = 1/2$ , y un número de Reynolds,  $Re = 16$ .

Paso de red ( $h^*$ )	Número de puntos ( $N_{total}$ )	Par. Relajación ( $w$ )	Iteraciones	Tiempo(s)
1/4	640	0,5	2819	48,60
1/8	2560	0,8	6160	463,29
1/10	4000	0,8	8574	1042,66
1/16	10240	0,9	14506	4417,87
1/20	16000	0,9	17919	8539,57
1/32	40960	0,8	16571	20279,89
1/40	64000	0,8	15193	28872,29

Tabla 1. Resultados numéricos para una malla fija de dimensiones  $10 \times 4$  variando el paso de red. Todas las simulaciones se han realizado con los mismos parámetros de simulación ( $L^* = 10, H^* = 4, L_x^* = 1, L_y^* = 1/2, Re = 16, w = 0,9, TOL = 10^{-5}$ ).

Con los datos de la Tabla 1 se puede realizar un ajuste lineal para obtener una ecuación que relacione el número de puntos de la simulación con el tiempo de computación en la máquina utilizada<sup>2</sup>, suponiendo que la convergencia se produce correctamente. En la Figura 21 se puede observar dicho ajuste lineal. Con esta ecuación, junto con la ecuación (188) se puede obtener una aproximación del tiempo de computación necesario en dicha máquina de usuario para una malla determinada.

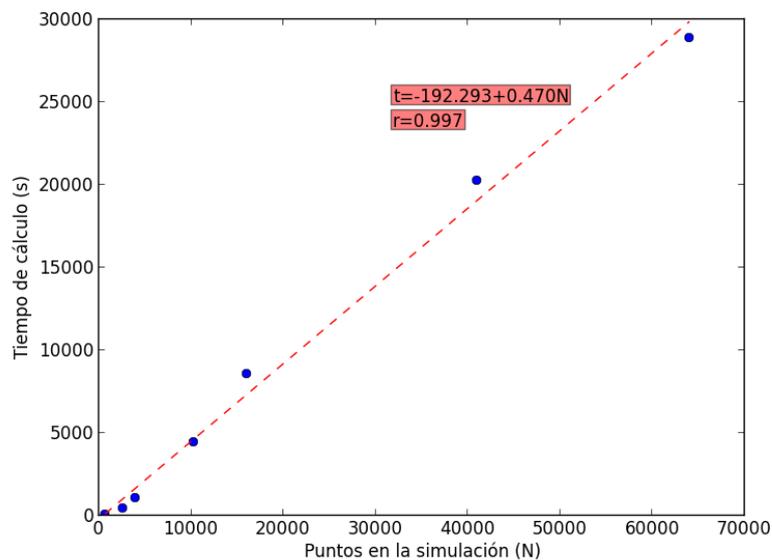


Figura 21. Tiempo de cálculo ( $t$ ) requerido para realizar una simulación en función del número de puntos ( $N$ ) en los que se va a realizar la simulación.

En la Figura 22 se representan la función de corriente y las líneas de corriente obtenidas en las distintas simulaciones realizadas, cuando se varía la resolución de la malla. Se observa que cuanto menor es el paso de red y, por tanto, mayor es la resolución, más horizontales son las líneas de corriente. También se observa que, excepto para la simulación de menor resolución  $h^* = 1/4$ , la aparición una circulación cerrada corriente abajo del obstáculo.

<sup>2</sup> Intel® Core™ i5-2300 CPU @ 3.00GHz, 8GB RAM, Windows 7 Home Premium

Para la simulación de menor resolución ,  $h^* = 1/4$ , como se puede ver en la Tabla 1, se ha necesitado utilizar un menor parámetro de relajación,  $w = 0.5$ , para alcanzar la convergencia. Además, como se observa en la Figura 23 en los contornos de vorticidad, para dicha resolución aparece un contorno de vorticidad anómalo situado cerca del vértice anterior del obstáculo.

Todos estas diferencias observadas en la simulación de menor resolución ,  $h^* = 1/4$ , se deben a que dicha resolución es o está cerca de ser insuficiente para resolver los contornos y las zonas de fuerte gradiente de la función de corriente.

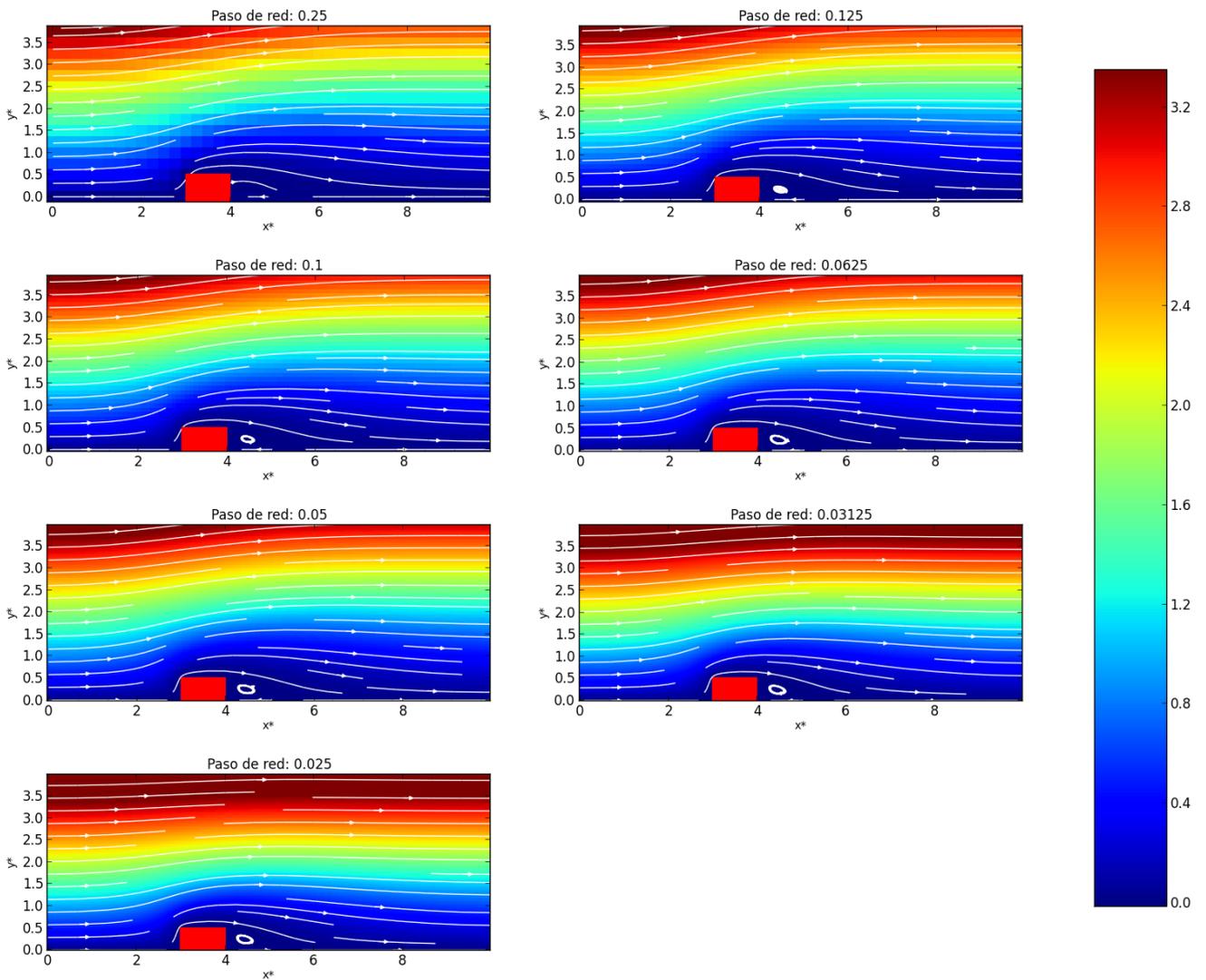


Figura 22. Función de corriente y líneas de corriente (líneas blancas con flechas) para distintas simulaciones con distinta resolución, pero con el mismo tamaño de malla. Todas las simulaciones se han realizado con los mismos parámetros de simulación ( $L^* = 10, H^* = 4, L_x^* = 1, L_y^* = 1/2, Re = 16, w = 0,9, TOL = 10^{-5}$ ).

En la Figura 23, para las mismas simulaciones, se representan los contornos de vorticidad. Se observa que, cuanto mayor es la resolución, un mismo contorno de vorticidad se extiende hasta más lejos corriente abajo del obstáculo. Además, conforme se aumenta la resolución, mayor es el valor que alcanza la vorticidad del flujo cerca de los vértices del obstáculo. Según Layton [12], esto se debe a que el imponer la condición “no-slip” en vértices puede provocar la aparición de resultados no físicos debidos a la acumulación de tensiones numéricas y, por

tanto, no físicas en dichos vértices. Por ello, como posible mejora a la resolución del problema, podría probarse a reemplazar las condiciones “no slip” por condiciones de “slip” con fricción cerca de los puntos con discontinuidad en las condiciones de contorno.

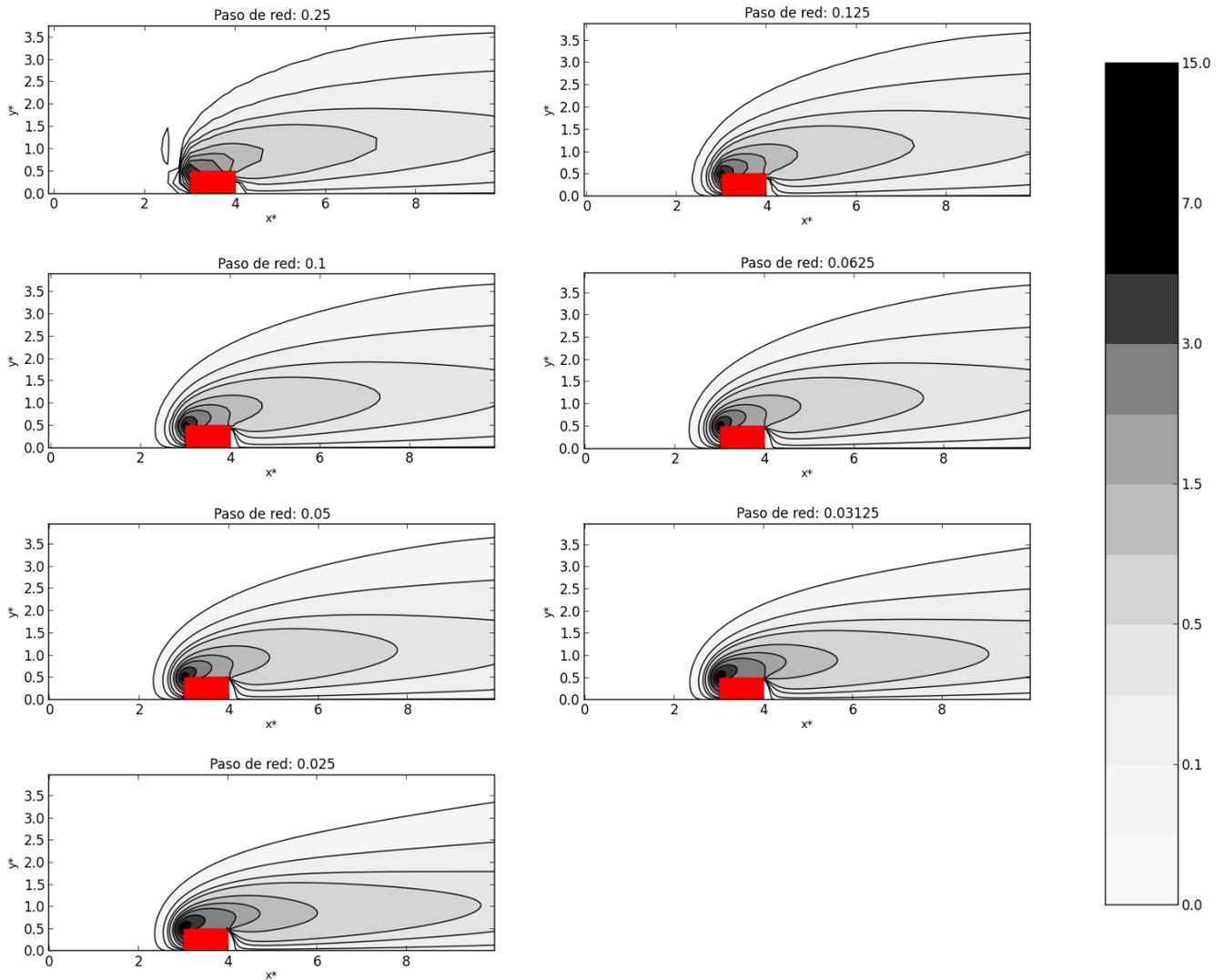


Figura 23. Contornos de vorticidad para distintas simulaciones con distinta resolución, pero con el mismo tamaño de malla. Todas las simulaciones se han realizado con los mismos parámetros de simulación ( $L^* = 10, H^* = 4, L_x^* = 1, L_y^* = 1/2, Re = 16, w = 0,9, TOL = 10^{-5}$ ).

A partir de ambas figuras se puede concluir que, al menos cualitativamente, las distintas resoluciones reproducen los mismos resultados. Sin embargo, es importante hacer, al menos, una comparación cuantitativa de lo previamente expuesto. Para ello, se han comparado las simulaciones de pasos de red  $h^* = 1/16$  y  $h^* = 1/32$ , en los puntos de malla en los que la simulación de paso de red  $h^* = 1/16$  toma valores. Los valores de las magnitudes para la simulación de paso de red  $h^* = 1/32$  se consideran como valores de referencia, notándose como  $S_{ij}^{ref}$ . Los valores de las magnitudes para la simulación de paso de red  $h^* = 1/16$  se notan como  $S_{ij}$ .

Un estimador sencillo para comparar las simulaciones puede ser la diferencia máxima puntual, tanto positiva ( $maxDif$ ) como negativa ( $minDif$ ), entre los resultados de las simulaciones.

$$maxDif = \max(S_{ij}^{ref} - S_{ij}) \quad (191)$$

$$minDif = \min(S_{ij}^{ref} - S_{ij}) \quad (192)$$

Otro estimador más avanzado podría ser obtener una desviación media,

$$\varepsilon = \frac{1}{N_x N_y} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} |S_{ij} - S_{ij}^{ref}| \quad (193)$$

o la desviación media relativa,

$$\varepsilon_r = \frac{1}{N_x N_y} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \left| \frac{S_{ij} - S_{ij}^{ref}}{S_{ij}^{ref}} \right| \quad (194)$$

En la Tabla 2 se pueden observar estos estimadores.

Parámetros			Función de corriente				Vorticidad			
Paso de red ( $h^*$ )	Iteraciones	Tiempo(s)	$maxDif$	$minDif$	$\varepsilon$	$\varepsilon_r$	$maxDif$	$minDif$	$\varepsilon$	$\varepsilon_r$
1/16	14506	4417,87	0,6374	-0,0018	0,2807	0,2003	9.6876	-0,2742	0,0509	3,9684
1/32	16571	20279,89	--	--	--	--	--	--	--	--

Tabla 2. Comparación de los resultados obtenidos para las simulaciones de pasos de red  $h^* = 1/16$  y  $h^* = 1/32$ . El resto de los parámetros de las simulaciones son los mismos ( $L^* = 10, H^* = 4, L_x^* = 1, L_y^* = 1/2, Re = 16, w = 0,9, TOL = 10^{-5}$ ).

Los estimadores de la Tabla 2 evidencian diferencias importantes entre las simulaciones. Para la función de corriente la desviación media es de casi el 0.3, siendo mayor la diferencia en los puntos cercanos al contorno de la malla. Para la vorticidad, la máxima diferencia se da en el vértice anterior del obstáculo, siendo claramente menor el error en el resto de los puntos. Sin embargo, el error relativo es muy alto, principalmente en los contornos de la malla.

Por ello, se puede concluir que la resolución es crucial para obtener unos buenos resultados numéricos, y su elección debe estar sujeta al tiempo de simulación disponible.

#### 6.4. Cambio del tamaño de la malla

En el apartado anterior se observó la gran importancia que tiene la resolución en los resultados de la simulación. En este apartado se va a tratar de analizar la importancia de los parámetros que fijan el tamaño de la malla, su largo  $L^*$  y su ancho  $H^*$ .

### 6.4.1. Aumento del largo de malla ( $L^*$ ) manteniendo fijos el alto y la resolución

Para analizar el efecto del largo de malla en la simulación se ha fijado el alto de malla en  $H^* = 4$  y el paso de red en  $h^* = 1/16$ . Los parámetros de la simulación serán  $Re = 16$ ,  $w = 0,9$  y una tolerancia  $TOL = 10^{-5}$ . Se han realizado cinco simulaciones con valores de  $L^*$  4, 7, 10, 12 y 15.

En la Figura 24 y en la Figura 25 se representan los campos de función de corriente y vorticidad obtenidos en estas simulaciones. Como el objetivo de este apartado es analizar el comportamiento alrededor del obstáculo se representa únicamente la parte de las simulaciones más cercana al obstáculo, concretamente una región de tamaño (4x4). En la Figura 24 puede apreciarse que, a medida que aumenta  $L^*$ , la inclinación de las líneas de corriente es mayor (apréciese el valor de la línea amarilla de valor 2.0 en la Figura 24). Esto se debe a que, aumentando  $L^*$ , las condiciones de contorno se imponen más lejos del obstáculo y, por tanto, influyen menos en el flujo alrededor del obstáculo. En la Figura 25, donde se representan los contornos de vorticidad, solamente se aprecian pequeñas diferencias cerca de la esquina posterior del obstáculo para la menor longitud.

Para hacer un análisis más detallado, es necesario establecer unas medidas cuantitativas con el fin de poder comparar estas diferencias. Si se consideran como referencia los resultados obtenidos con la malla de mayor largo, se pueden obtener una serie de estimadores que midan la diferencia, punto a punto, entre los resultados las simulaciones. Estas diferencias se calcularán solamente en el área de tamaño representada, con el fin de obtener unos resultados comparables. Para ello, se utilizan unas coordenadas definidas en la malla (4x4) y se aplicarán los estimadores definidos en el apartado 6.3. En la Tabla 3 se puede observar la evolución de estos estimadores.

Parámetros			Función de corriente				Vorticidad			
Malla	Iteraciones	Tiempo(s)	maxDif	minDif	$\varepsilon$	$\varepsilon_r$	maxDif	minDif	$\varepsilon$	$\varepsilon_r$
4x4	13906	1653,03	0,0669	-0,3152	0,0767	0,2385	0,6155	-2,5264	0,0477	0,4364
7x4	14690	3150,56	0,0112	-0,1854	0,0473	0,0963	0,1617	-1,1606	0,0189	0,1575
10x4	14506	4417,87	0,0016	-0,0779	0,0236	0,0332	0,0341	-0,4903	0,0076	0,0557
12x4	13916	5042,73	0,0001	-0,0247	0,0099	0,0096	0,0053	-0,1690	0,0027	0,0296
15x4	13360	6097,05	--	--	--	--	--	--	--	--

Tabla 3. Comparación cuantitativa de las distintas simulaciones en relación con la simulación de mayor largo de malla para la función de corriente y para la vorticidad. En la tabla se presenta el número de iteraciones y el tiempo necesario para realizar las simulaciones con parámetros ( $L_x^* = 1, L_y^* = 1/2, Re = 16, w = 0,9, TOL = 10^{-5}$ ).

Los datos de la tabla evidencian que, conforme se aumenta el largo de la malla, los datos se van acercando a la simulación de referencia, lo que da confianza en el método y en los resultados.

En la elección del largo de la malla, es también importante tener en cuenta el tiempo de computación para el cual se cumple el criterio de convergencia. Según los datos de la Tabla 3 la mejora de los resultados va aparejada a un aumento del tiempo de computación, lo que hace necesario alcanzar un compromiso entre la precisión a alcanzar y el coste computacional. Para

los siguientes apartados, en los que vamos a utilizar un largo de malla fijo, se va a utilizar un largo  $L^*=10$ , ya que los errores respecto a la simulación de referencia son, del orden de  $2 \times 10^{-2}$  para la función de corriente y del orden de  $8 \times 10^{-3}$  para la vorticidad, suficientemente razonables.

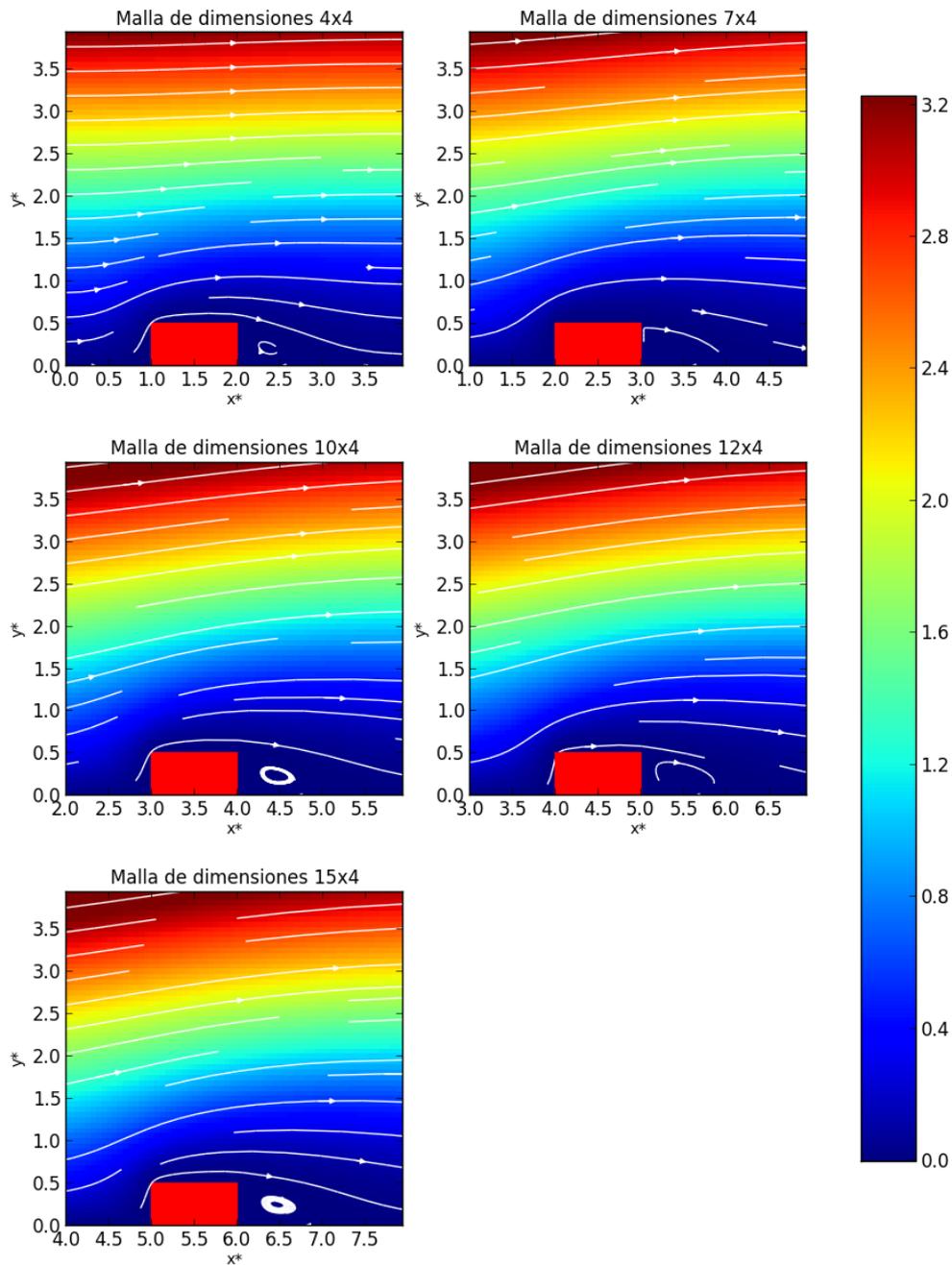


Figura 24. Función de corriente para distintas simulación con el mismo alto de malla ( $H^* = 4$ ), pero distinto largo,  $L^*$  (4, 7, 10, 12 y 15). Todas las simulaciones se han realizado con los mismos parámetros de simulación ( $L_x^* = 1, L_y^* = 1/2, Re = 16, w = 0, 9, h^* = 1/16, TOL = 10^{-5}$ ).

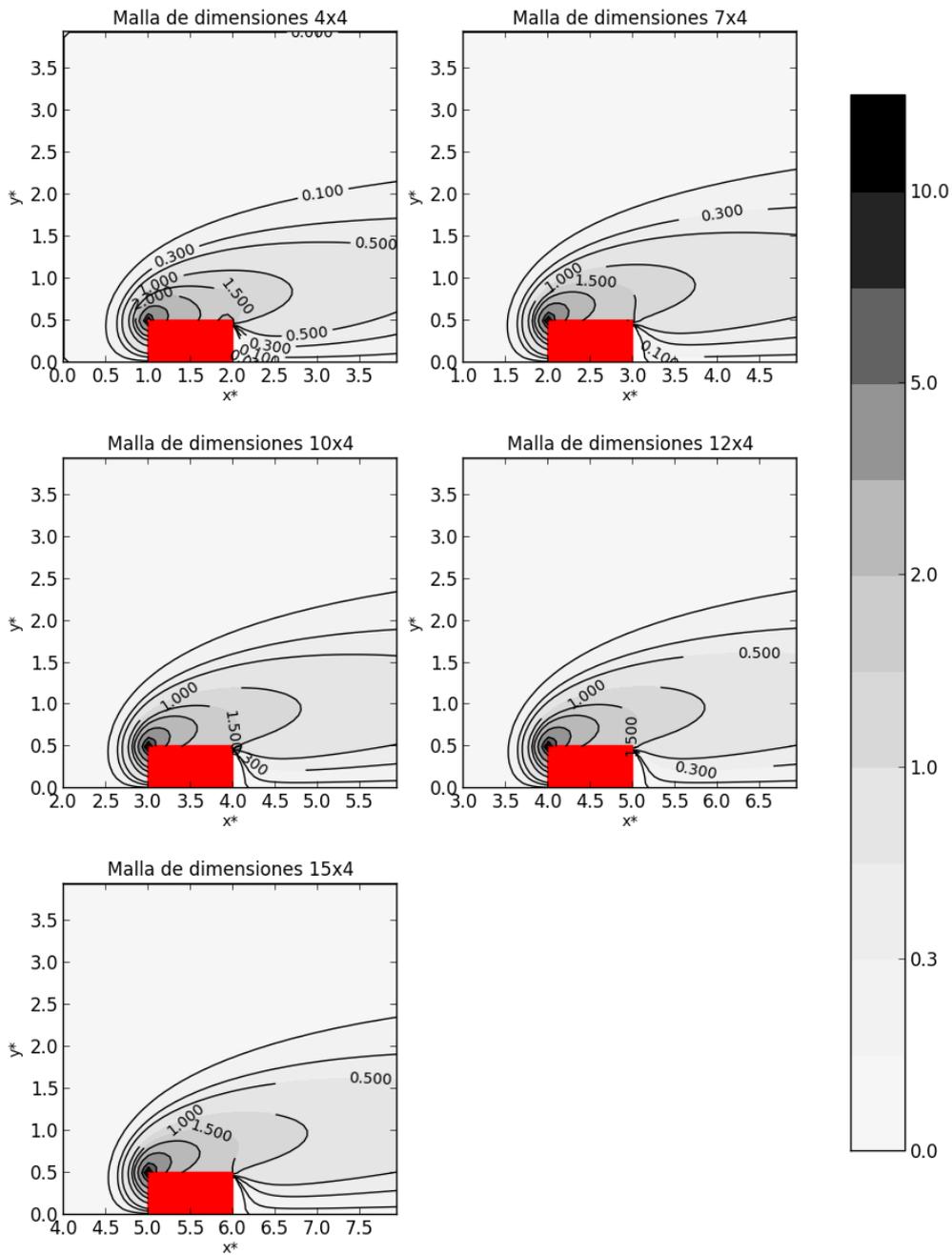


Figura 25. Contornos de vorticidad para distintas simulación con el mismo alto de malla ( $H^* = 4$ ), pero distinto largo,  $L^*$  (4, 7, 10, 12 y 15). Todas las simulaciones se han realizado con los mismos parámetros de simulación ( $L_x^* = 1, L_y^* = 1/2, Re = 16, w = 0, 9, h^* = 1/16, TOL = 10^{-5}$ ).

#### 6.4.2. Aumento del alto de malla ( $H^*$ ) manteniendo fijos el largo y la resolución

Para analizar el efecto del alto de la malla en la simulación se ha fijado el largo de malla en  $L^* = 10$  y el paso de red en  $h^* = 1/16$ . Los parámetros de la simulación serán  $Re = 16$ ,

$w = 0,9$  y una tolerancia  $TOL = 10^{-5}$ . Se han realizado cinco simulaciones con valores de  $H^*$  2, 2.5, 3, 3.5 y 4.

En la Figura 26 y en la Figura 27 se representan la función de corriente y la vorticidad para las distintas simulaciones, en el dominio de la malla de menor alto (10x2). En la Figura 26 es interesante observar que el aumento de  $H^*$  hace que la circulación cerrada que se produce detrás del obstáculo tenga mayor extensión. Esto se debe a que para las simulaciones con menor alto de malla,  $H^* = 2$  y  $H^* = 2.5$ , el contorno afecta al campo de función de corriente impidiéndole desarrollarse completamente.

Para la vorticidad se observa en la Figura 27 que para un ancho de malla pequeño los contornos de vorticidad se ven forzados a cerrarse. Para las simulaciones con  $H^* = 3$  o superior, este efecto no se observa, presentando todas el mismo.

En la Tabla 4 se pueden observar los valores que toman estimadores definidos en el apartado 6.3 para las distintas simulaciones. En este caso la simulación de referencia empleada es la que tiene un mayor alto.

Parámetros			Función de corriente				Vorticidad			
Malla	Iteraciones	Tiempo(s)	maxDif	minDif	$\varepsilon$	$\varepsilon_r$	maxDif	minDif	$\varepsilon$	$\varepsilon_r$
10x2	4144	613,21	0,5241	-0,0589	0,1479	0,5807	0,3342	-0,3012	0,0823	0,0823
10x2,5	7013	1301,27	0,2113	-0,0697	0,0491	0,1390	0,0541	-0,2394	0,0273	0,1931
10x3	10094	2261,14	0,0351	-0,0639	0,0152	0,0685	0,0356	-0,2029	0,0150	0,0836
10x3,5	9911	2603,48	0,0082	-0,0375	0,0117	0,0491	0,0164	-0,1488	0,0084	0,0331
10x4	14506	4417,87	--	--	--	--	--	--	--	--

Tabla 4. Comparación de las distintas simulaciones en relación con la simulación de mayor alto de malla  $H^* = 4$ , para la función de corriente y para la vorticidad. En la tabla se presenta el número de iteraciones y el tiempo necesario para realizar las simulaciones con parámetros ( $L_x^* = 1, L_y^* = 1/2, Re = 16, w = 0,9, TOL = 10^{-5}$ ).

Los datos de la tabla evidencian que, conforme se aumenta el alto de la malla, los datos se van acercando a la simulación de referencia, siendo el error más de ocho veces mayor para la malla de tamaño 10x2 que para la de tamaño 10x3, lo que da confianza en el método y en los resultados.

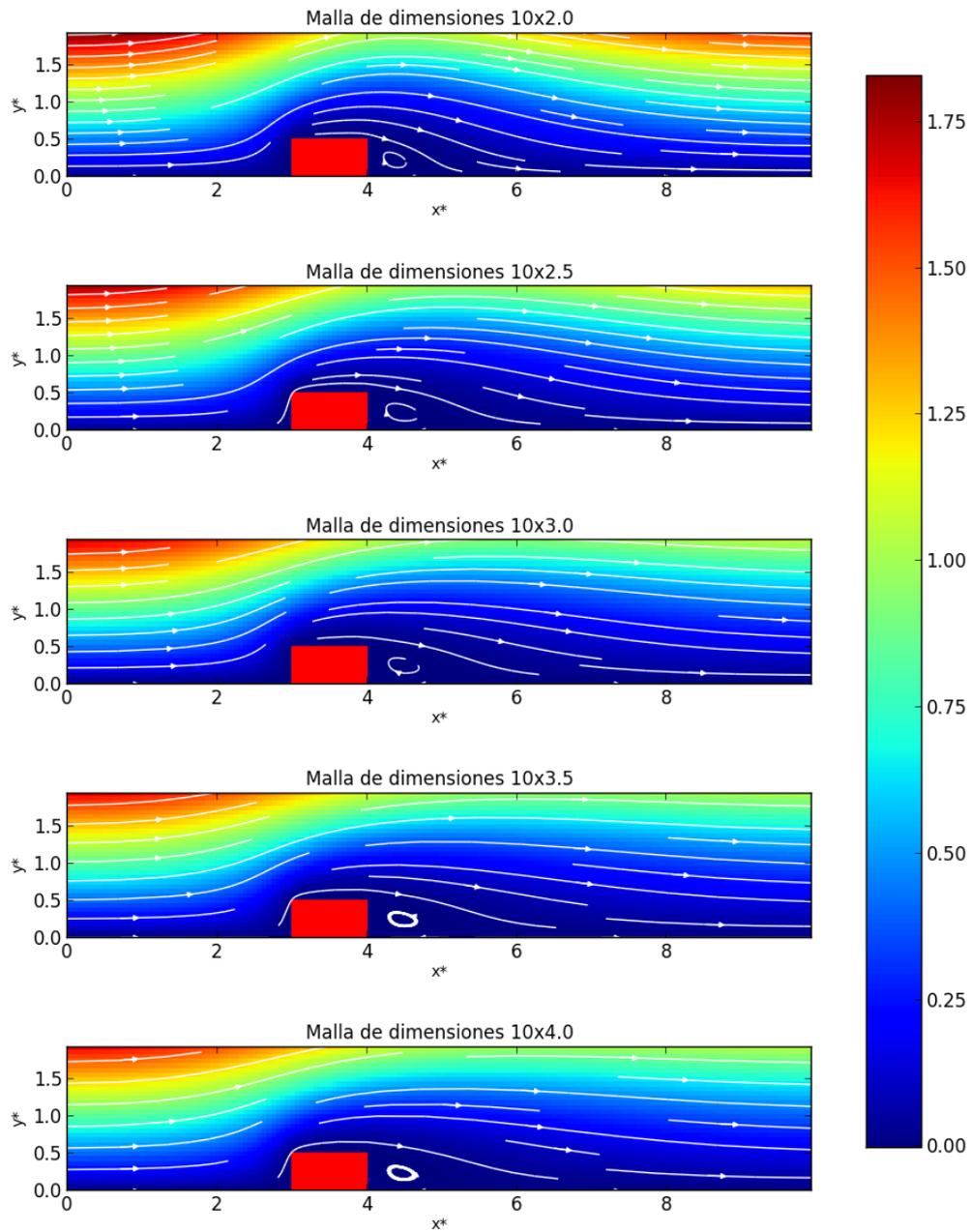


Figura 26. Función de corriente para distintas simulación con el mismo largo de malla ( $L^* = 10$ ), pero distinto alto,  $H^*$ . Todas las simulaciones se han realizado con los mismos parámetros de simulación ( $L_x^* = 1, L_y^* = 1/2, Re = 16, w = 0,9, h^* = 1/16, TOL = 10^{-5}$ ).

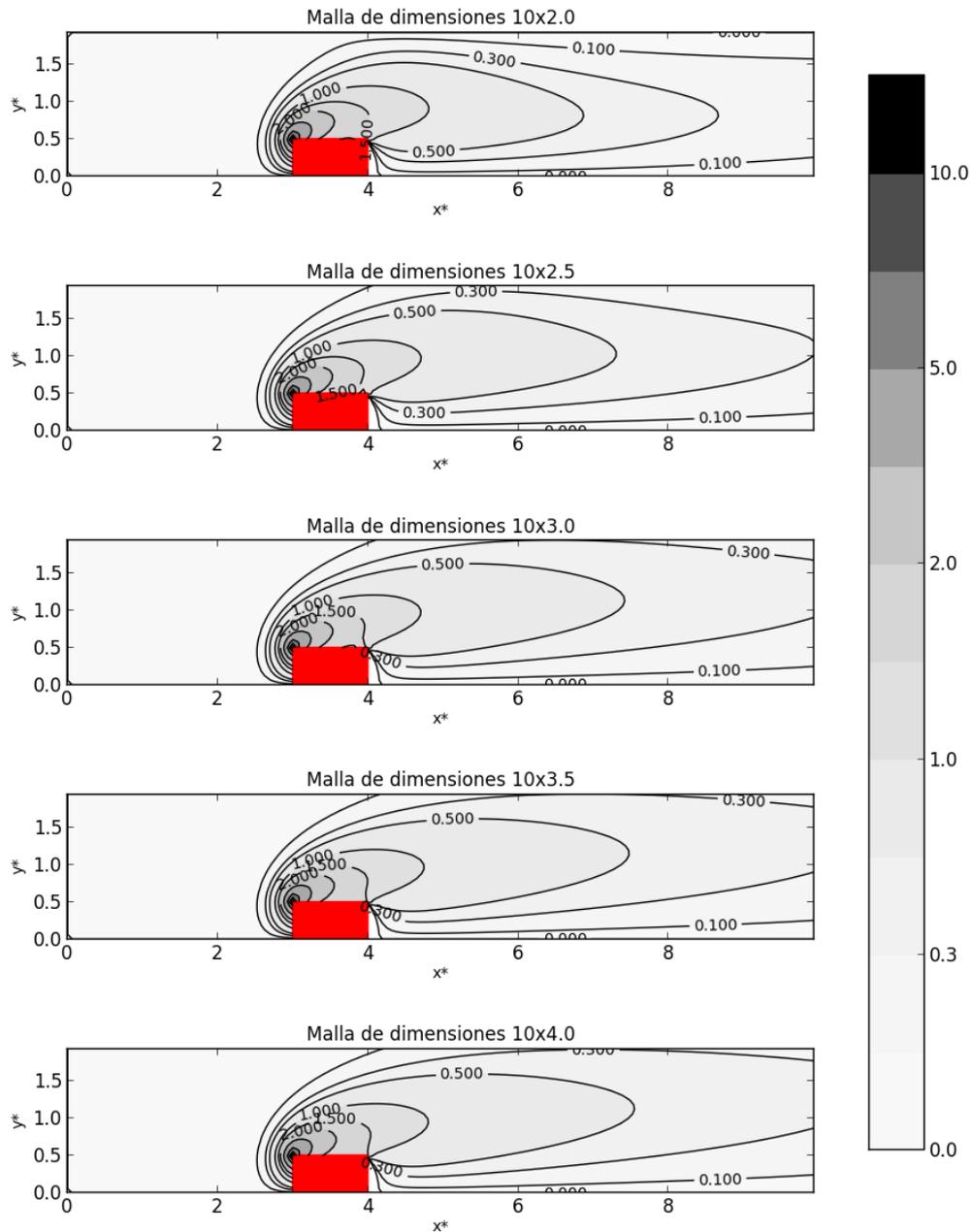


Figura 27. Contornos de vorticidad para distintas simulación con el mismo largo de malla ( $L^* = 10$ ), pero distinto alto,  $H^*$ . Todas las simulaciones se han realizado con los mismos parámetros de simulación ( $L_x^* = 1, L_y^* = 1/2, Re = 16, w = 0,9, h^* = 1/16, TOL = 10^{-5}$ ).

### 6.5. Efecto de la elección de distintas condiciones de contorno

En el apartado 4.2 se escogieron una serie de condiciones de contorno para los distintos contornos y se mencionó la existencia de posibles condiciones de contorno alternativas en varios de ellos. En este apartado se va a analizar el efecto que tendría sobre la simulación la elección de alguna de estas condiciones de contorno alternativas.

A. Contorno C. 

Cuando se discretizaron las condiciones de contorno en el contorno C se mencionaron dos posibles condiciones de contorno para la vorticidad, definidas por las ecuaciones (148) y (149).

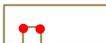
Se van a realizar dos simulaciones, manteniendo todos los parámetros iguales, variando solamente la condición de contorno para la vorticidad impuesta en C.

Con el fin de compararlas, se utilizarán los estimadores definidos en el apartado 6.1, considerando que la simulación que utiliza la ecuación de contorno (149) será la de referencia. En la Tabla 5 se pueden apreciar los resultados de estos estimadores.

Función de corriente				Vorticidad			
$maxDif$	$minDif$	$\varepsilon$	$\varepsilon_r$	$maxDif$	$minDif$	$\varepsilon$	$\varepsilon_r$
5,49E-08	-5,27E-04	2,00E-04	3,60E-04	9,03E-03	-1,00E-03	1,45E-04	1,17E-03

Tabla 5. Comparación de las simulaciones realizadas variando la condición de contorno en el Contorno C, para la función de corriente y para la vorticidad. Los parámetros de las simulaciones son  $L^* = 10$ ,  $H^* = 4$ , con un obstáculo de tamaño  $L_x^* = 1$ ,  $L_y^* = 1/2$ , un paso de red  $h^* = 1/16$ , una tolerancia  $TOL = 10^{-5}$  y un número de Reynolds  $Re = 16$ .

Se puede observar como el error cometido al variar la condición de contorno en C es muy pequeño. Además, los errores relativos son muy pequeños, del orden de  $10^{-3}$  para la vorticidad y de  $10^{-4}$  para la función de corriente, por lo que se puede considerar que los resultados obtenidos utilizando las ecuaciones (148) y (149) son muy parecidos.

B. Esquinas superiores del obstáculo. 

El interesante analizar la elección de las condiciones de contorno en los contornos del obstáculo. De la descripción realizada en el apartado 3.2, quedaba claro que la condición para todo el contorno del obstáculo es la condición de “no-slip”. Sin embargo, a la hora de discretizarla en diferencias finitas para la vorticidad surgía la duda de que condiciones aplicar en las esquinas del obstáculo. En el apartado 4.2 se decidió aplicar las ecuaciones discretizadas (172) y (173), que son una mezcla de los contornos laterales y superior.

En lo que sigue se van a probar otros tipos de condiciones de contorno en las esquinas, y en lugar de utilizar la condición (172) en la esquina anterior del obstáculo y la condición (173) en la esquina posterior, se utilizan las ecuaciones de contorno de los contornos laterales, o la ecuación de contorno del contorno superior. En la primera simulación, se utiliza la ecuación (160) para el punto de intersección  $F \cap G$  y la ecuación (168) para el punto de intersección  $G \cap H$ . En la segunda, se utiliza la ecuación (164) para los dos puntos de intersección,  $F \cap G$  y  $G \cap H$ .

Para compararlas, en la Tabla 6 se presentan los estimadores definidos en el apartado 6.1, considerando a la simulación realizada con las ecuaciones (172) y (173) como simulación de referencia.

Condición de contorno empleada	Función de corriente				Vorticidad			
	<i>maxDif</i>	<i>minDif</i>	$\varepsilon$	$\varepsilon_r$	<i>maxDif</i>	<i>minDif</i>	$\varepsilon$	$\varepsilon_r$
Ec. (160) (168) en esquinas	4,46E-04	-5,40E-03	6,18E-03	6,08E-02	4,65E-01	-2,59E-01	3,68E-03	3,12E-02
Ec. (164) en esquinas	4,92E-03	-8,57E-04	5,51E-03	5,90E-02	2,54E-01	-4,55E-01	3,64E-03	2,91E-02

Tabla 6. Comparación de las simulaciones realizadas variando las condiciones de contorno en las esquinas del obstáculo,  $F \cap G$  y  $G \cap H$ , para la función de corriente y para la vorticidad. Los parámetros de las simulaciones son  $L^* = 10$ ,  $H^* = 4$ , con un obstáculo de tamaño  $L_x^* = 1$ ,  $L_y^* = 1/2$ , un paso de red  $h^* = 1/16$ , una tolerancia  $TOL = 10^{-5}$  y un número de Reynolds  $Re = 16$ .

Si se analizan los datos de la Tabla 6 se observa que las desviaciones son del orden de  $10^{-3}$ , por lo que los resultados obtenidos con las tres simulaciones pueden considerarse muy parecidos. Sin embargo, para la vorticidad las diferencias puntuales si son relevantes, y en el vértice posterior del obstáculo las simulaciones difieren en prácticamente media unidad de vorticidad respecto a la simulación de referencia, como se ve en *maxDif* para la simulación con las ecuaciones (160) y (168) en las esquinas y, para *minDif* para la simulación con la ecuación (164) en las esquinas. Esto se debe a que la vorticidad es máxima en las esquinas, siendo la solución mezcla una solución intermedia entre las otras dos.

### 6.6. Importancia de las condiciones de partida para el método iterativo

En el apartado 5 se definió un algoritmo secuencial para calcular la función de corriente y la vorticidad. En el primer paso de dicho algoritmo se hace mención a dos posibles maneras de establecer las condiciones de partida, utilizando bien la ecuación (177), bien la ecuación (178). En este apartado se caracteriza el efecto que tiene sobre la convergencia de las simulaciones la elección de distinta condiciones de partida. Para ello, se van a realizar simulaciones para distintos números de Reynolds con distintas condiciones de partida o arranque.

En la Tabla 7 se recogen los resultados obtenidos para una malla de tamaño  $L^* = 10$ ,  $H^* = 4$ , con un obstáculo de tamaño  $L_x^* = 1$ ,  $L_y^* = 1/2$ , un paso de red  $h^* = 1/16$  y una tolerancia  $TOL = 10^{-4}$ .

De los resultados de la Tabla 7 se puede inferir que las condiciones de arranque son importantes para garantizar la convergencia del método, principalmente para simulaciones en las que el número de Reynolds es alto. Además, como se ha explicado en el apartado 5, la utilización de unas condiciones de arranque cercanas al resultado esperado acelera la convergencia, reduciendo el número de iteraciones necesarias para alcanzar una solución numérica que garantice la tolerancia buscada.

$Re$	Condiciones de partida	$w_{max}$	Nº de iteraciones
0,1	Flujo libre	0,9	2719
1	Flujo libre	0,9	2409
	$Re' = 0,1$	0,9	795
2	Flujo libre	0,9	2099
	$Re' = 0,1$	0,9	857
	$Re' = 1$	0,9	360
5	Flujo libre	0,9	1797
	$Re' = 0,1$	0,9	1200
	$Re' = 1$	0,9	854
	$Re' = 2$	0,9	787
8	Flujo libre	0,9	1700
	$Re' = 0,1$	0,9	1259
	$Re' = 1$	0,9	964
	$Re' = 2$	0,9	932
	$Re' = 5$	0,9	528
16	Flujo libre	0,9	1596
	$Re' = 0,1$	0,9	1355
	$Re' = 1$	0,9	1108
	$Re' = 2$	0,9	1107
	$Re' = 5$	0,9	828
	$Re' = 8$	0,9	608

$Re$	Condiciones de partida	$w_{max}$	Nº de iteraciones
24	Flujo libre	0,9	1553
	$Re' = 0,1$	0,9	1394
	$Re' = 1$	0,9	1168
	$Re' = 2$	0,9	1178
	$Re' = 5$	0,9	943
	$Re' = 8$	0,9	749
	$Re' = 16$	0,9	408
32	Flujo libre	0,6	1665
	$Re' = 0,1$	0,9	1411
	$Re' = 1$	0,9	1198
	$Re' = 2$	0,9	1213
	$Re' = 5$	0,9	1002
	$Re' = 8$	0,9	821
	$Re' = 16$	0,9	512
40	Flujo libre	0,4	1801
	$Re' = 24$	0,9	235
64	Flujo libre	0,2	2043
	$Re' = 40$	0,9	235
88	Flujo libre	0,1	2318
	$Re' = 64$	0,1	303

Tabla 7. Número de iteraciones necesarias para alcanzar la convergencia en distintas simulaciones, con una tolerancia de  $10^{-4}$ , para distintas condiciones de partida. El campo  $w_{max}$  representa el valor máximo que puede tomar el parámetro de relajación para que la simulación converja. Para las simulaciones se ha utilizado una malla de  $10 \times 4$ , con un obstáculo de tamaño  $L_x^* = 1$ ,  $L_y^* = 1/2$ , con un paso de red  $h^* = 1/16$ .

### 6.7. Comparativa del método implementado con el método de Lattice Boltzmann

En el apartado 6.1 se hizo mención de los resultados obtenidos por Breuer et al. [9], Perumal et al. [10] y Yoon et al. [11] mediante métodos de resolución más avanzados y se compararon con los obtenidos en el presente trabajo. Sin embargo, las conclusiones que se pueden sacar de dicha comparación son limitadas, ya que solamente se pueden comparar las gráficas de líneas de corriente y contornos de vorticidad para distintos números de Reynolds y el valor la longitud de recirculación. Por ello, en el presente apartado se va a resolver el problema objeto de este trabajo mediante el método de Lattice Boltzmann[7][13].

El código implementado para desarrollar el método de Lattice Boltzmann se incluye en el anexo como apartado 9.2 y ha sido adaptado al problema de estudio a partir de un desarrollo de la empresa *Flowkit Ltd* [14]. Este método funciona adecuadamente para número de Reynolds superiores a uno, por lo que la comparación se limitará al caso de  $Re = 16$ .

Los contornos de vorticidad obtenidos por este método, representados en la Figura 28, son muy parecidos a los obtenidos por nuestro método (Figura 15), si bien los contornos se extienden algo menos corriente abajo del obstáculo. El campo de velocidad obtenido por Lattice Boltzmann, representado en la Figura 29, tiene un notable parecido con el calculado por nuestro método (Figura 16), encontrándose diferencias únicamente lejos del obstáculo. Esta semejanza da mucha confianza en el método y en los resultados.

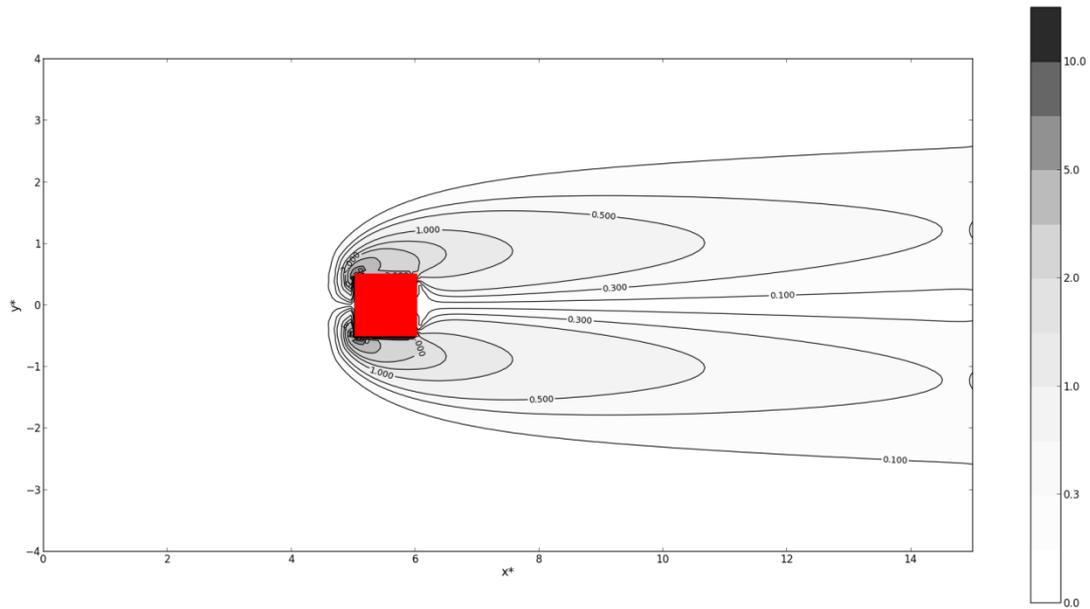


Figura 28. Líneas de contorno del módulo de la vorticidad para  $Re = 16$ , obtenido mediante el método Lattice Boltzmann, en un instante en el que se ha alcanzado el régimen estacionario. Los parámetros de la simulación son  $L^* = 15$ ,  $H^* = 4$ , con un obstáculo de tamaño  $L_x^* = 1$ ,  $L_y^* = 1/2$ , y un paso de red  $h^* = 1/32$ .

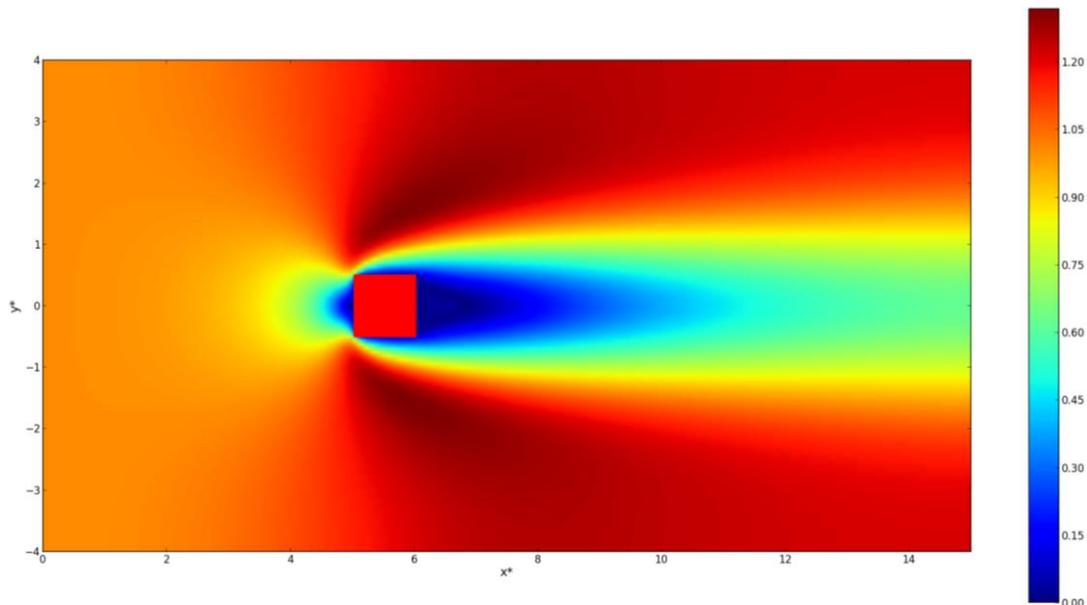


Figura 29. Módulo del campo de velocidad para  $Re = 16$ , obtenido mediante el método Lattice Boltzmann, en un instante en el que se ha alcanzado el régimen estacionario. Los parámetros de la simulación son  $L^* = 15$ ,  $H^* = 4$ , con un obstáculo de tamaño  $L_x^* = 1$ ,  $L_y^* = 1/2$ , y un paso de red  $h^* = 1/32$ .

## 6.8. Número de Reynolds crítico

Según la Tabla 7 el mayor número de Reynolds para el que se puede resolver el problema es  $Re_{max} = 88$ . Para obtenerlo se ha necesitado emplear un parámetro de relajación  $w = 0,1$ , ya que para cualquier otro parámetro el método de resolución no converge.

Este número de Reynolds,  $Re_{max}$ , puede considerarse como un límite superior del denominado número de Reynolds crítico,  $Re_{crit}$ , a partir del cual no se puede alcanzar un régimen estacionario. Este resultado excede el medido experimentalmente por Sohankar et al. [15] ( $Re_{crit} = 47 \pm 2$ ), y al obtenido por el método de Lattice Boltzmann introducido en el apartado 6.7,  $Re_{crit} = 51$ . Por tanto, puede considerarse que los resultados obtenidos para  $Re > Re_{crit}$  son correctos numéricamente hablando, pero no físicamente, ya que en la formulación del problema se está forzando al flujo a permanecer en un régimen estacionario.

Según Layton [12], a partir de este número de Reynolds crítico, los vórtices situados tras las esquinas posteriores del obstáculo se hacen demasiado largos y se rompen, difundiéndose corriente abajo alternativamente. Se genera así un movimiento no estacionario, que lógicamente el código de resolución creado para el problema, debido a su formulación, no es capaz de resolver.

Para ilustrar este flujo no estacionario, que el código numérico implementado no puede resolver, volvemos a utilizar el método de Lattice Boltzmann introducido en el apartado 6.7, representando los distintos campos en instantes de tiempo distintos.

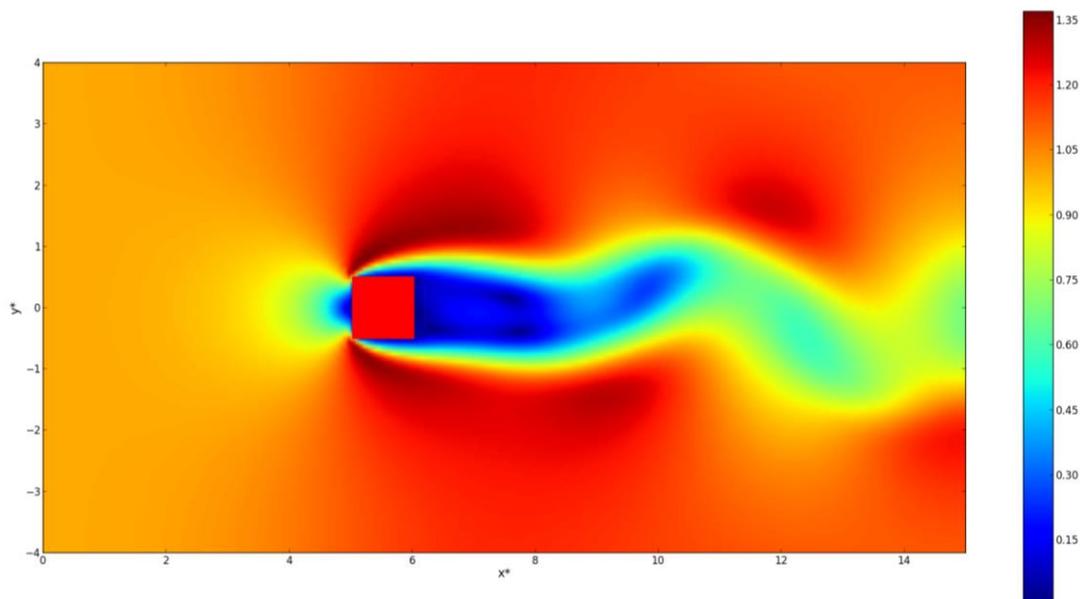


Figura 30. Módulo del campo de velocidad para  $Re = 64$ , obtenido mediante el método Lattice Boltzmann, en la iteración temporal  $t = 100000$ . Los parámetros de la simulación son  $L^* = 15$ ,  $H^* = 4$ , con un obstáculo de tamaño  $L_x^* = 1$ ,  $L_y^* = 1/2$ , y un paso de red  $h^* = 1/32$ .

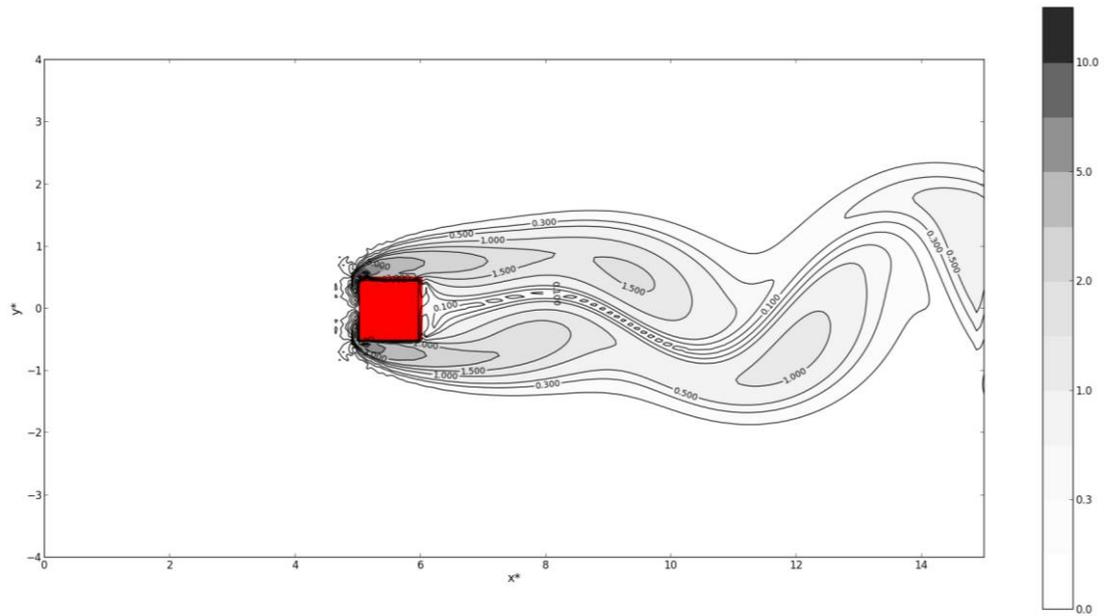


Figura 31. Contornos del campo de vorticidad para  $Re = 64$ , obtenido mediante el método Lattice Boltzmann, en la iteración temporal  $t = 100000$ . Los parámetros de la simulación son  $L^* = 15$ ,  $H^* = 4$ , con un obstáculo de tamaño  $L_x^* = 1$ ,  $L_y^* = 1/2$ , y un paso de red  $h^* = 1/32$ .

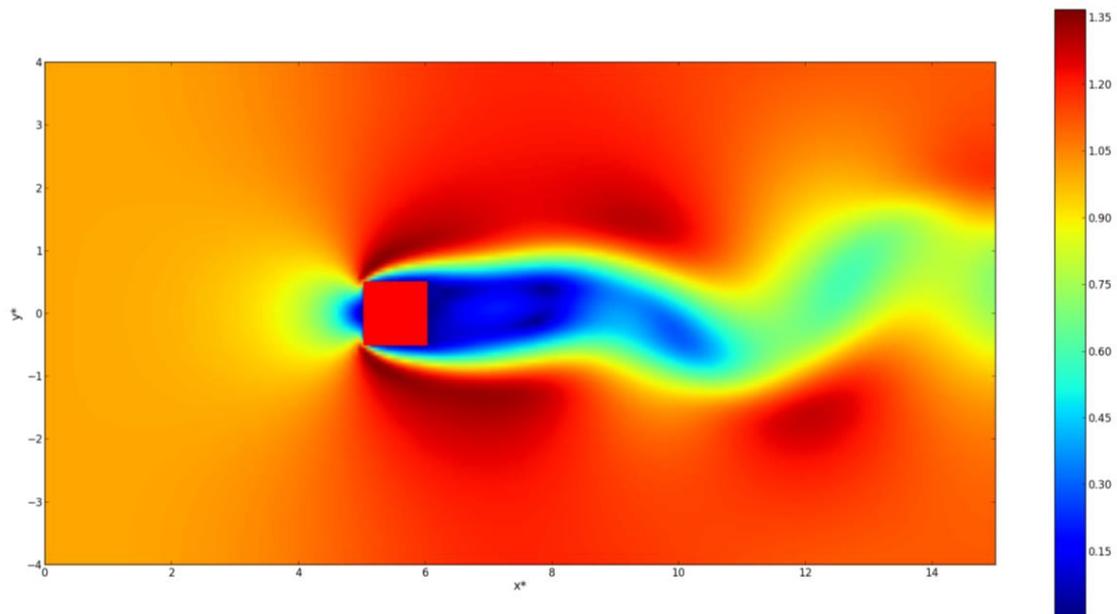


Figura 32. Módulo del campo de velocidad para  $Re = 64$ , obtenido mediante el método Lattice Boltzmann, en la iteración temporal  $t = 120000$ . Los parámetros de la simulación son  $L^* = 15$ ,  $H^* = 4$ , con un obstáculo de tamaño  $L_x^* = 1$ ,  $L_y^* = 1/2$ , y un paso de red  $h^* = 1/32$ .

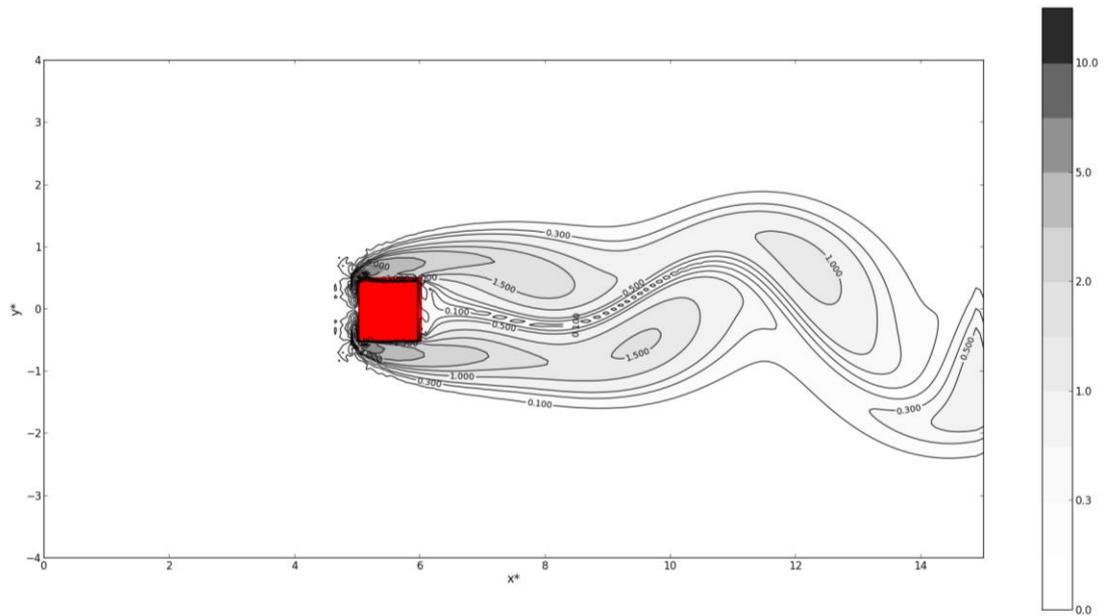


Figura 33. Contornos del campo de vorticidad para  $Re = 64$ , obtenido mediante el método Lattice Boltzmann, en la iteración temporal  $t = 120000$ . Los parámetros de la simulación son  $L^* = 15$ ,  $H^* = 4$ , con un obstáculo de tamaño  $L_x^* = 1$ ,  $L_y^* = 1/2$ , y un paso de red  $h^* = 1/32$ .

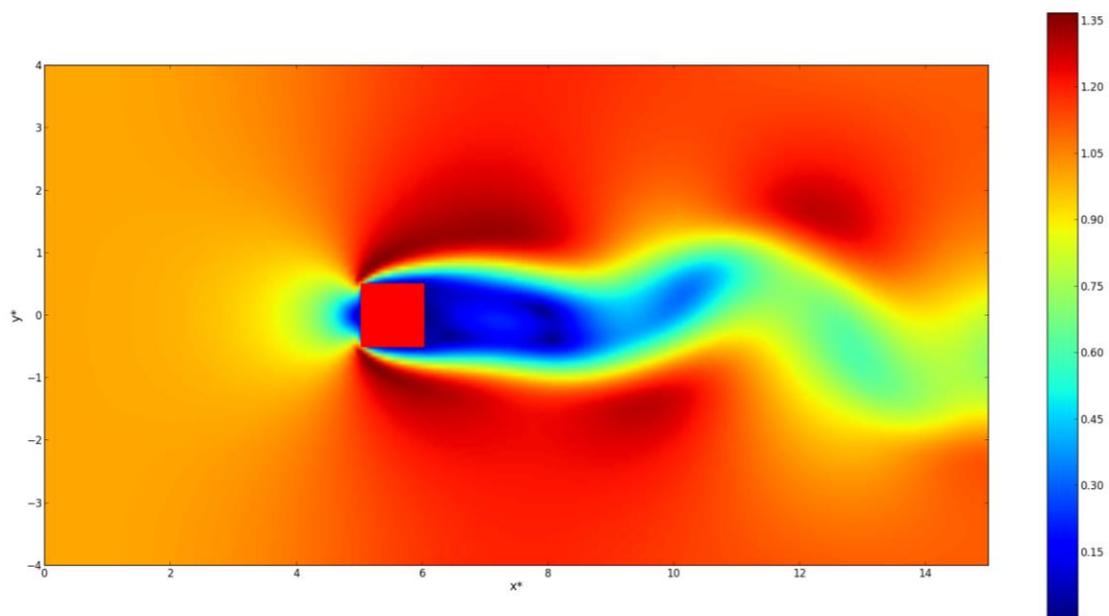


Figura 34. Módulo del campo de velocidad para  $Re=64$ , obtenido mediante el método Lattice Boltzmann, en la iteración temporal  $t=140000$ . Los parámetros de la simulación son  $L^* = 15$ ,  $H^* = 4$ , con un obstáculo de tamaño  $L_x^* = 1$ ,  $L_y^* = 1/2$ , y un paso de red  $h^* = 1/32$ .

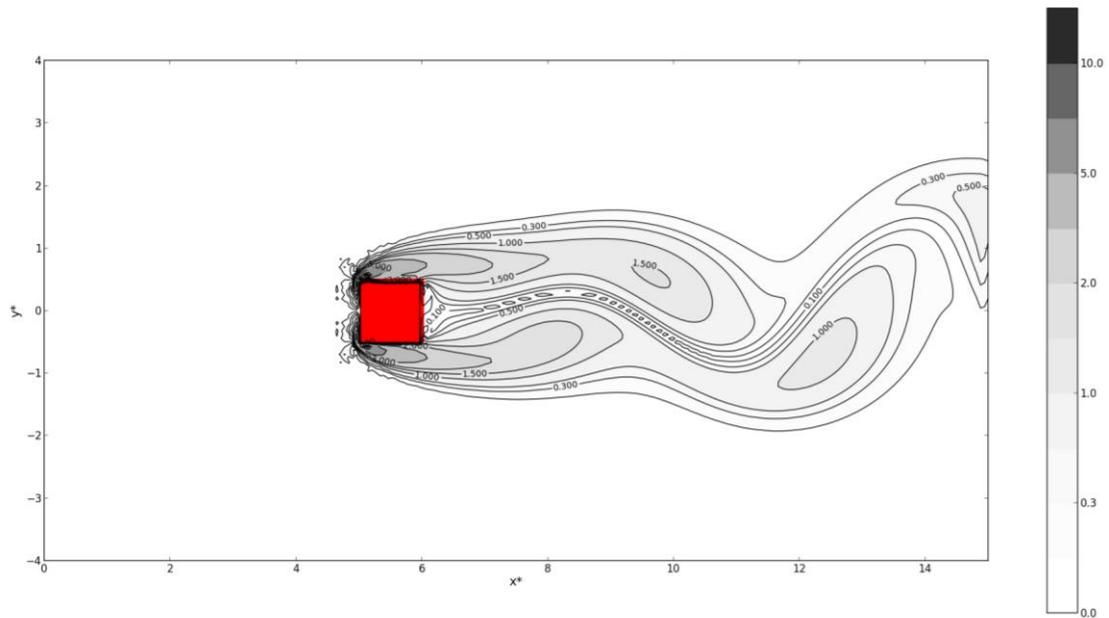


Figura 35. Contornos del campo de vorticidad para  $Re = 64$ , obtenido mediante el método Lattice Boltzmann, en la iteración temporal  $t = 140000$ . Los parámetros de la simulación son  $L^* = 15$ ,  $H^* = 4$ , con un obstáculo de tamaño  $L_x^* = 1$ ,  $L_y^* = 1/2$ , y un paso de red  $h^* = 1/32$ .

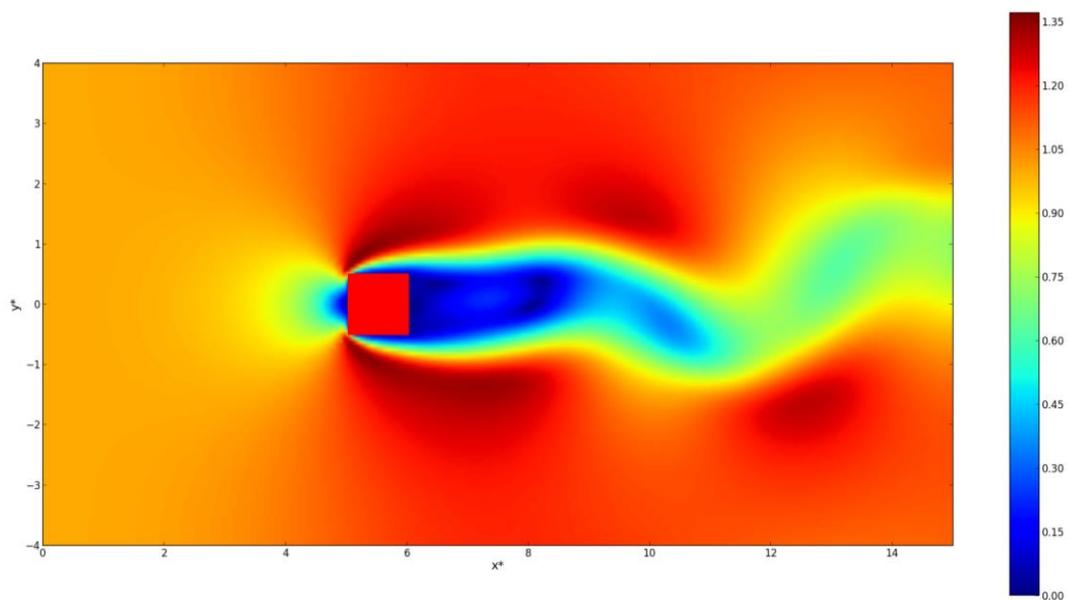


Figura 36. Módulo del campo de velocidad para  $Re = 64$ , obtenido mediante el método Lattice Boltzmann, en la iteración temporal  $t = 160000$ . Los parámetros de la simulación son  $L^* = 15$ ,  $H^* = 4$ , con un obstáculo de tamaño  $L_x^* = 1$ ,  $L_y^* = 1/2$ , y un paso de red  $h^* = 1/32$ .

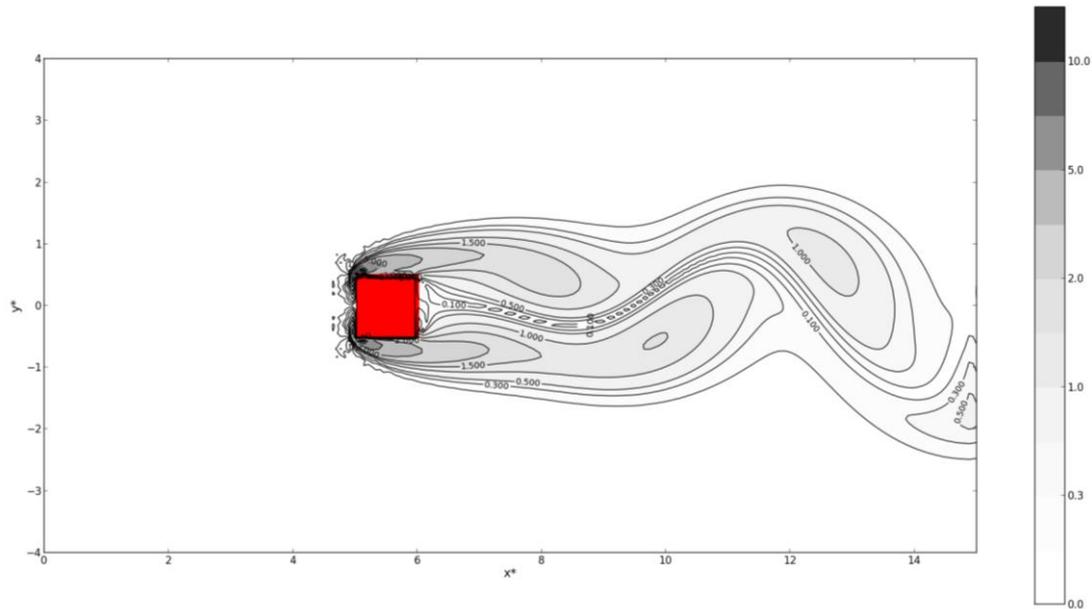


Figura 37. Contornos del campo de vorticidad para  $Re = 64$ , obtenido mediante el método Lattice Boltzmann, en la iteración temporal  $t = 160000$ . Los parámetros de la simulación son  $L^* = 15$ ,  $H^* = 4$ , con un obstáculo de tamaño  $L_x^* = 1$ ,  $L_y^* = 1/2$ , y un paso de red  $h^* = 1/32$ .

En las distintas figuras, para  $Re = 64$ , se puede apreciar cómo los vórtices se difunden corriente abajo rompiendo la simetría del problema y formando la denominada “calle de vórtices de Von Karman”. [16]

### 6.9. Efecto de un segundo obstáculo en las simulaciones

Con el fin de ilustrar las posibilidades que presenta el programa realizado, se ha realizado una adaptación del mismo que incluye un segundo obstáculo, y se han analizado dos casos en los que se ha variado la distancia entre los obstáculos, para dos números de Reynolds,  $Re = 1$  y  $Re = 16$ .

- Obstáculos situados a una distancia  $L_x^*$ .

Se puede observar en la Figura 38 para la función de corriente y en la **¡Error! No se encuentra el origen de la referencia.** para la vorticidad que el comportamiento conjunto de ambos obstáculos, para un número de Reynolds ( $Re = 1$ ) bajo, es muy parecido al que produciría un obstáculo que ocupase el espacio entre la posición inicial del primero y la final del segundo.

Para un número de Reynolds más alto ( $Re = 16$ ) la situación es más interesante, ya que como se aprecia en la Figura 38 se produce una circulación cerrada entre ambos obstáculos. Además, en la parte posterior del segundo obstáculo no se aprecia la formación de un vórtice, como ocurría para un solo obstáculo (Figura 14). Esto se debe a que el flujo que incide sobre la parte anterior del segundo obstáculo prácticamente no se ve alterado, como se aprecia en la **¡Error! No se encuentra el origen de la referencia.** observando los contornos de vorticidad en los vértices de este segundo obstáculo.

- Obstáculos situados a una distancia  $3L_x^*$ .

En este caso, como se puede observar en la Figura 40 para la función de corriente y en la Figura 41 para los contornos de vorticidad, para un número de Reynolds ( $Re = 1$ ) bajo, el comportamiento del flujo cerca del segundo obstáculo no se ve prácticamente influenciado por la existencia del primer obstáculo. Esto se debe a que el primer obstáculo recibe un flujo que perturba y, cuando este flujo está ya recuperándose, se encuentra con el segundo obstáculo que vuelve a perturbarlo.

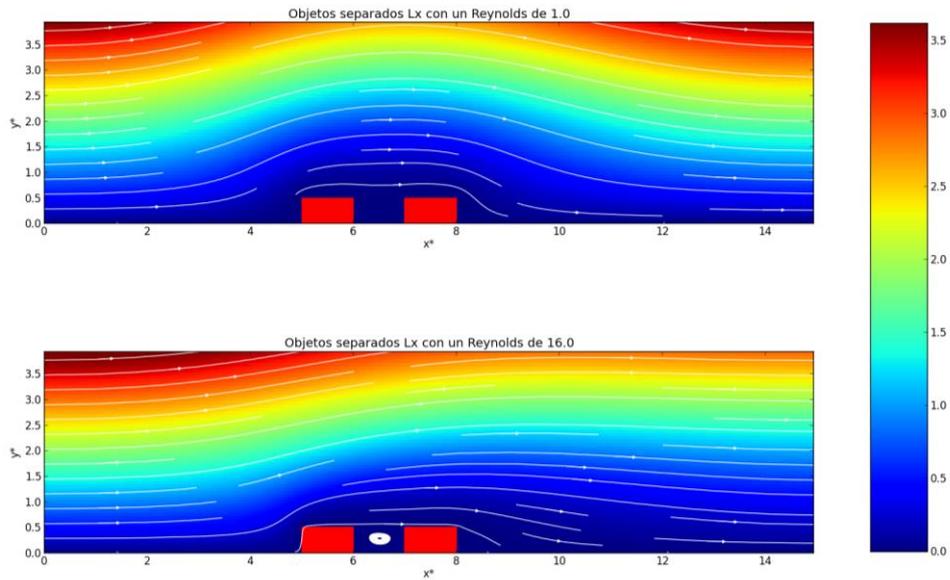


Figura 38. Función de corriente y líneas de corriente (en blanco) para dos obstáculos separados una distancia  $d^* = 1$ . Los parámetros de la simulación son  $L^* = 15$ ,  $H^* = 4$ , con un obstáculo de tamaño  $L_x^* = 1$ ,  $L_y^* = 1/2$ , un paso de red  $h^* = 1/16$  y una tolerancia  $TOL = 10^{-5}$ . En la gráfica superior el número de Reynolds es 1.0 y en la inferior es 16.0.

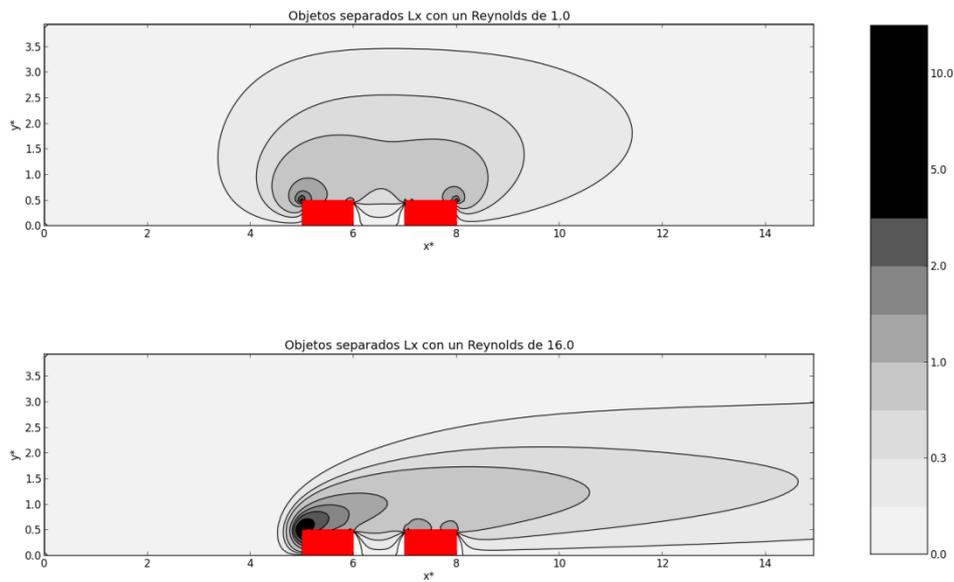


Figura 39. Contornos de vorticidad para dos obstáculos separados una distancia  $d = 1$ . Los parámetros de la simulación son  $L^* = 15$ ,  $H^* = 4$ , con un obstáculo de tamaño  $L_x^* = 1$ ,  $L_y^* = 1/2$ , un paso de red  $h^* = 1/16$  y una tolerancia  $TOL = 10^{-5}$ . En la gráfica superior el número de Reynolds es 1.0 y en la inferior es 16.0.

Para un número de Reynolds más alto ( $Re = 16$ ), se produce un vórtice tras el primer obstáculo, pero que prácticamente alcanza el segundo obstáculo. Además, al igual que en el caso de los obstáculos situados a una distancia  $d = 1$ , en la parte posterior del segundo obstáculo la formación de un vórtice no es tan clara como en el caso de existir un solo obstáculo (Figura 14).

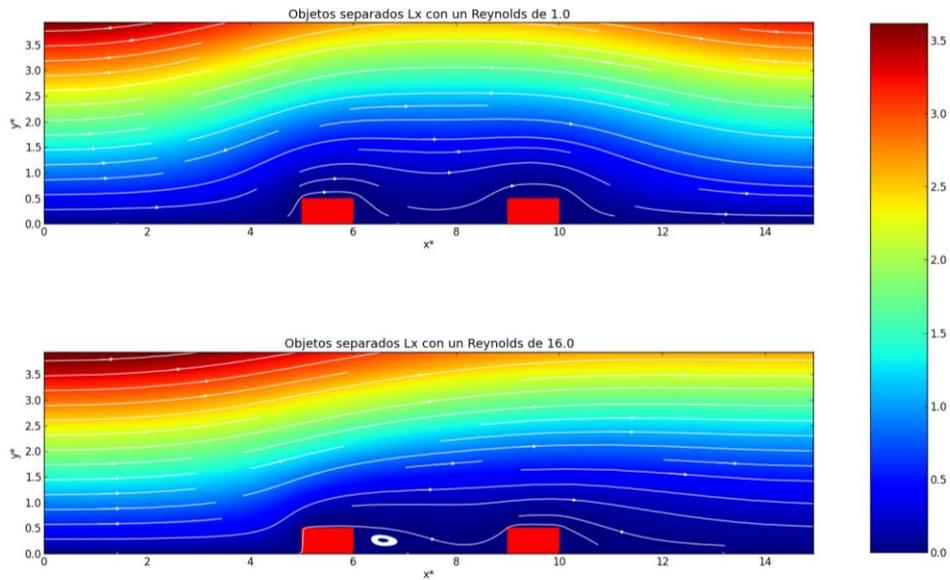


Figura 40. Función de corriente y líneas de corriente (en blanco) para dos obstáculos separados una distancia  $d^* = 3$ . Los parámetros de la simulación son  $L^* = 15$ ,  $H^* = 4$ , con un obstáculo de tamaño  $L_x^* = 1$ ,  $L_y^* = 1/2$ , un paso de red  $h^* = 1/16$  y una tolerancia  $TOL = 10^{-5}$ . En la gráfica superior el número de Reynolds es 1.0 y en la inferior es 16.0.

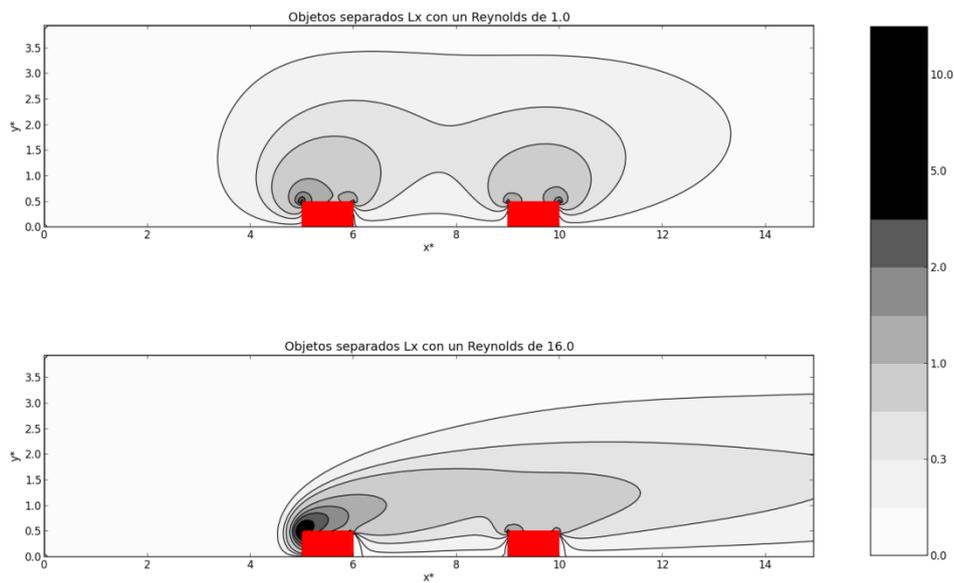


Figura 41. Contornos de vorticidad para dos obstáculos separados una distancia  $d = 3$ . Los parámetros de la simulación son  $L^* = 15$ ,  $H^* = 4$ , con un obstáculo de tamaño  $L_x^* = 1$ ,  $L_y^* = 1/2$ , un paso de red  $h^* = 1/16$  y una tolerancia  $TOL = 10^{-5}$ . En la gráfica superior el número de Reynolds es 1.0 y en la inferior es 16.0.

## 7. Conclusiones

El objetivo del trabajo era resolver numéricamente el comportamiento de un flujo incompresible y viscoso alrededor de un obstáculo ortoédrico para distintos números de Reynolds.

La utilización de la representación de la vorticidad-función de corriente, tal como se describe en el apartado 3.1, parece ventajosa respecto a la habitual representación en componentes de la velocidad, ya que permite desacoplar el cálculo de la presión del resto de las variables. Como contrapartida, la formulación de las condiciones de contorno es algo más complicada.

En el apartado 4.2 se escogió discretizar las ecuaciones del problema mediante el método de diferencias finitas, ya que este método tiene como puntos fuertes la facilidad con la que se pueden interpretar físicamente los resultados obtenidos y la simplicidad a la hora de elegir los puntos donde se van a calcular los campos. La principal desventaja de este método es que mantiene la misma separación entre todos los puntos de la red, sin distinguir las zonas que pueden ser más relevantes del resto. Esto dificulta el estudio del comportamiento del flujo cerca del obstáculo, ya que para obtener una mayor resolución cerca del obstáculo el tiempo de computación se hace muy grande, tal como se aprecia en la Figura 21. Por ello, podría ser interesante utilizar el método de volúmenes finitos, tal como plantean Breuer et al. [9], lo que permitiría un ajuste fino de la malla en las regiones de interés cercanas al obstáculo.

En cuanto a los resultados del problema, se reprodujo adecuadamente el comportamiento, muy cercano al de un flujo de Stokes, de un flujo estacionario con  $Re < 1$ . Además, para  $Re = 16$ , los resultados estaban de acuerdo con los estudios de Breuer et al. [9] y Perumal et al. [10], reproduciendo correctamente no sólo la aparición de una región de recirculación, si no su longitud.

Del análisis de los parámetros de la simulación se puede concluir que una malla de largo  $L^* = 10$  y alto  $H^* = 3$  es suficiente para reproducir unos resultados adecuados, si se utiliza la mejor resolución posible, de acuerdo con el tiempo de computación disponible.

Por último, es interesante hacer referencia al método Lattice-Boltzmann. En este método, utilizado en el apartado 6.8 para explicar el comportamiento del fluido cuando  $Re > Re_c$ , permite obtener una descripción de la física a nivel macroscópico a partir de la física a nivel microscópico [7], lo que lo hace muy interesante como futura línea de estudio del problema analizado.

En conclusión, el presente trabajo demuestra la capacidad del método de diferencias finitas utilizado para resolver el flujo alrededor de un obstáculo para  $Re < Re_{crit}$ , valor a partir del cual los resultados obtenidos dejan de tener un significado físico.

## 8. Bibliografía

- [1] – Dollet, B., Durth, M. & Graner, F. 2006, "Flow of foam past an elliptical obstacle", *Physical review. E, Statistical, nonlinear, and soft matter physics*, vol. 73, no. 6 Pt 1, pp. 061404.
- [2] – "Discussions of bridge aerodynamics", 1997, *Journal of Wind Engineering & Industrial Aerodynamics*, vol. 67, pp. 962-965.
- [3] – Lacaze, L., Paci, A., Cid, E., Cazin, S., Eiff, O., Esler, J.G. & Johnson, E.R. 2013, "Wave patterns generated by an axisymmetric obstacle in a two-layer flow", *Experiments in Fluids*, vol. 54, no. 12, pp. 1-10.
- [4] – Lauga, E., Brenner, M.P. & Stone, H.A. 2005, "Microfluidics: The no-slip boundary condition".
- [5] – Bustamante, M. D., *Mixed Eulerian-Lagrangian approaches for the simulation of fundamental turbulence problems*. Meeting "EULERIAN-LAGRANGIAN", WPI, Vienna, May 2012. [http://www.wpi.ac.at/themedata/euler\\_lagrangian\\_2012/Bustamante\\_discussion\\_1\\_plus\\_Euler\\_Lagrange\\_models.pdf](http://www.wpi.ac.at/themedata/euler_lagrangian_2012/Bustamante_discussion_1_plus_Euler_Lagrange_models.pdf)
- [6] – Burden, R.L. & Faires, J.D. 2001, *Numerical analysis*, Brooks/Cole, Pacific Grove (California).
- [7] – He, X. & Luo, L. 1997, "Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation", *Physical Review E*, vol. 56, no. 6, pp. 6811-6817.
- [8] – Kirby, B.J. (2010). [\*Micro- and Nanoscale Fluid Mechanics: Transport in Microfluidic Devices\*](#). Cambridge University Press
- [9] – Breuer, M., Bernsdorf, J., Zeiser, T. & Durst, F. 2000, "Accurate computations of the laminar flow past a square cylinder based on two different methods: lattice-Boltzmann and finite-volume", *International Journal of Heat and Fluid Flow*, vol. 21, no. 2, pp. 186-196.
- [10] – Perumal, D.A., Kumar, G.V.S. & Dass, A.K. 2012, "Numerical Simulation of Viscous Flow over a Square Cylinder Using Lattice Boltzmann Method", *ISRN Mathematical Physics*, vol. 2012, pp. 1-16.
- [11] – Dong-Hyeog Yoon, Kyung-Soo Yang, Choon-Bum Choi. 2010, "Flow past a square cylinder with an angle of incidence", *Physics of Fluids*, vol. 22, no. 4, pp. 043603-043603-12
- [12] – Layton, W.J. 2008, *Introduction to the numerical analysis of incompressible viscous flows*, Society for Industrial and Applied Mathematics, Philadelphia.
- [13] – Uribe Laverde, M.A. & Arévalo Díaz, L.A. 2009, "Simulación del Comportamiento de un Fluido Bordeado un Obstáculo Cilíndrico por el Método Lattice Boltzmann", *Revista de la Sociedad Colombiana de Física*, vol. 41, no. 1, pp. 229-232.
- [14] – <http://www.flowkit.com/>

[15] – Sohankar, A., Norberg, C. & Davidson, L. 1998, "Low-Reynolds-number flow around a square cylinder at incidence: study of blockage, onset of vortex shedding and outlet boundary condition", *International Journal for Numerical Methods in Fluids*, vol. 26, no. 1, pp. 39-56.

[16] – Theodore von Kármán, *Aerodynamics*. McGraw-Hill (1963).

## 9. Anexo

### 9.1. Código general del programa

```
'''
CVentana
En el archivo CVentana se define La interfaz del programa. Sera el archivo a ejecutar
para correr el programa

@author: Jaime
'''

import Tkinter as tk
import os
import CSimulacion
import numpy as np
import CPintar

DIR_TRABAJO = '\Users\Jaime\workspace\Trabajo2' #Para trabajar en Windows
# DIR_TRABAJO = '/home/jreyv/Desktop/Trabajo2' #Para trabajar en Linux

class CAplicacion(tk.Frame):
    '''
    Clase base a partir de La cual se genera La ventana con La que podremos realizar
    Las simulaciones y dibujar Los resultados. Hereda de La clase Frame de Tkinter
    '''

    def __init__(self, master):
        '''
        El Constructor de La clase
        '''
        tk.Frame.__init__(self, master)
        self.pack()
        self.crearWidgets()
        self.representarLin = False
        self.representarVor = False
        self.representarPres = False
        self.representarVel = False
        self.representarTodo = False

    def crearWidgets(self):
        '''
        El metodo en el que creamos el contenido de La ventana
        '''

        self.etiquetaMalla = tk.Label(text="Tamano de La malla: ",
fg="blue").place(x=10, y=10)
        self.tiquetaObjeto = tk.Label(text="Tamano del objeto: ",
fg="blue").place(x=10, y=40)
        self.etiquetaParametros = tk.Label(text="Parametros de La simulacion:",
fg="blue").place(x=10, y=70)
        self.etiquetaNx = tk.Label(text="Nx: ", fg="red").place(x=180, y=10)
        self.etiquetaNy = tk.Label(text="Ny: ", fg="red").place(x=250, y=10)

        self.entradaNx = tk.StringVar()
        self.entradaNx.set(160)
        self.entradaNy = tk.StringVar(self)
        self.entradaNy.set(64)
        self.txtetiquetaNx = tk.Entry(textvariable=self.entradaNx,
width=5).place(x=210, y=10)
        self.txtetiquetaNy = tk.Entry(textvariable=self.entradaNy,
width=5).place(x=280, y=10)
```

```

self.etiquetaLx = tk.Label(text="Lx: ", fg="red").place(x=180, y=40)
self.etiquetaLy = tk.Label(text="Ly: ", fg="red").place(x=250, y=40)
self.entradaLx = tk.StringVar()
self.entradaLx.set(16)
self.txtetiquetaLx = tk.Entry(textvariable=self.entradaLx,
width=5).place(x=210, y=40)
self.entradaLy = tk.StringVar()
self.entradaLy.set(8)
self.txtetiquetaLy = tk.Entry(textvariable=self.entradaLy,
width=5).place(x=280, y=40)

self.etiquetaW = tk.Label(text="w: ", fg="red").place(x=180, y=70)
self.etiquetaRe = tk.Label(text="Re: ", fg="red").place(x=250, y=70)
self.etiquetaTol = tk.Label(text="TOL:", fg="red").place(x=250, y=100)
self.etiquetaH = tk.Label(text="h:", fg="red").place(x=180, y=100)
self.etiquetaL = tk.Label(text="L:", fg="red").place(x=110, y=100)
self.entradaW = tk.StringVar()
self.entradaW.set(0.9)
self.txtetiquetaW = tk.Entry(textvariable=self.entradaW,
width=5).place(x=210, y=70)
self.entradaRe = tk.StringVar()
self.entradaRe.set(16.0)
self.txtetiquetaRe = tk.Entry(textvariable=self.entradaRe,
width=5).place(x=280, y=70)
self.entradaTol = tk.StringVar()
self.entradaTol.set(1e-4)
self.txtetiquetaTol = tk.Entry(textvariable=self.entradaTol,
width=5).place(x=280, y=100)
self.entradaH = tk.StringVar()
self.entradaH.set(1.0)
self.txtetiquetaH = tk.Entry(textvariable=self.entradaH,
width=5).place(x=210, y=100)

# Botones
self.botonCorrer = tk.Button(text="Correr", command=self.correr,
font=("14")).place(x=20, y=140)
self.botonReinicio = tk.Button(text="Valores por defecto",
command=self.restablecer, font=("14")).place(x=180, y=140)
self.botonDibujar = tk.Button(text="Dibujar", command=self.dibujar,
font=("14")).place(x=265, y=240)
self.botonCorrerArchivo = tk.Button(text="Correr ci", command=self.correrCi,
font=("14")).place(x=90, y=140)
self.casillaMascara = tk.IntVar()
self.casillaLin = tk.IntVar()
self.casillaVor = tk.IntVar()
self.casillaPres = tk.IntVar()
self.casillaVel = tk.IntVar()
self.casillaTodo = tk.IntVar()

self.ElegirPintarLin = tk.Checkbutton(text="Funcion de corriente",
variable=self.casillaLin,command=self.elegirPintarLin).place(x=265,y=310)
self.ElegirPintarVor = tk.Checkbutton(text="Vorticidad",
variable=self.casillaVor,command=self.elegirPintarVor).place(x=265,y=330)
self.ElegirPintarPres = tk.Checkbutton(text="Presion",
variable=self.casillaPres,command=self.elegirPintarPres).place(x=265,y=350)
self.ElegirPintarVel = tk.Checkbutton(text="VeLocidad",
variable=self.casillaVel,command=self.elegirPintarVel).place(x=265,y=370)
self.ElegirPintarTodo = tk.Checkbutton(text="Todo",
variable=self.casillaTodo,command=self.elegirPintarTodo).place(x=265,y=390)

# ListBox
ficheros = os.listdir(DIR_TRABAJO)
ficherosDat = []
for fichero in ficheros:
    if fichero.endswith('.dat'):

```

```

        ficherosDat.append(fichero)

self.etiquetaArchivos = tk.Label(text="Archivos ya creados").place(x=10,
y=200)
self.listaArchivos = tk.Listbox(width=41, selectmode=tk.SINGLE)
for ficheroDat in ficherosDat:
    self.listaArchivos.insert(tk.END, ficheroDat)
# listaArchivos.bind("<<ListboxSelect>>", elegir)
self.listaArchivos.place(x=10, y=230)

def elegirPintarLin(self):
    self.representarLin = not self.representarLin

def elegirPintarVor(self):
    self.representarVor = not self.representarVor

def elegirPintarPres(self):
    self.representarPres = not self.representarPres

def elegirPintarVel(self):
    self.representarVel = not self.representarVel

def elegirPintarTodo(self):
    self.representarTodo = not self.representarTodo

def restablecer(self):
    self.entradaNx.set(240)
    self.entradaNy.set(60)
    self.entradaLx.set(16)
    self.entradaLy.set(8)
    self.entradaW.set(1.0)
    self.entradaRe.set(1.0)
    self.entradaTol.set(1e-3)
    self.entradah.set(1.0)
    #self.entradal.set(16)
    self.etiquetaSimular = tk.Label(text="Valores por defecto restablecidos",
fg="blue", background="yellow").place(x=10, y=180)
    return

def leerArchivo(self, nombreArchivo):
    npzfile = np.load(nombreArchivo)
    lin = npzfile['arr_0']
    vor = npzfile['arr_1']
    mask = npzfile['arr_2']
    mask2 = npzfile['arr_3']
    return lin, vor, mask, mask2

def dibujar(self):
    nombreArchivo = self.listaArchivos.get(self.listaArchivos.curselection())
    dibujo = CPintar.CPintar(nombreArchivo)
    dibujo.leerArchivo(nombreArchivo)
    if self.representarTodo == True:
        camposPintar = [True, True, True, True]
    else:
        camposPintar = [self.representarLin, self.representarVor,
self.representarPres, self.representarVel]
    if any(camposPintar):
        dibujo.plot(camposPintar)
    else:
        self.etiquetaSimular = tk.Label(text="No ha seleccionado que campos
quiere pintar", fg="blue", background="yellow").place(x=10, y=180)

def getVariables(self):
    Nx = int(self.entradaNx.get())

```

```

Ny = int(self.entradaNy.get())
Lx = int(self.entradaLx.get())
Ly = int(self.entradaLy.get())
w = float(self.entradaW.get())
Re = float(self.entradaRe.get())
h = float(self.entradaH.get())
TOL = float(self.entradaTol.get())

return Nx, Ny, Lx, Ly, w, Re, h, TOL

def correr(self):
    Nx, Ny, Lx, Ly, w, Re, h, TOL = self.getVariables()
    if (Nx > 300 or Nx < 20) or (Ny > 100 and Ny < 20):
        self.etiquetaSimular = tk.Label(text="La malla no es adecuada debido a su
tamano [" + self.entradaNx.get() + "," + self.entradaLy.get() + "]", fg="blue",
background="yellow").place(x=10, y=180)
    else:
        sim = CSimulacion.CSimulacion(Nx, Ny, Lx, Ly, w, Re, h, TOL)
        simulacion=sim.simular()
        if simulacion == "error":
            self.etiquetaSimular = tk.Label(text="El metodo no converge para Re="
+ self.entradaNx.get() + " y w=" + self.entradaNx.get(), fg="blue",
background="yellow").place(x=10, y=180)
        else:
            self.etiquetaSimular = tk.Label(text="Simulacion correctamente
realizada en " + str(simulacion[2]) + " segundos y \n" + str(simulacion[1]) + "
pasos", fg="blue", background="yellow").place(x=10, y=180)
            with open("simulaciones.txt", "a") as myfile:
                logText = "{0}x{1} {2}x{3} h:{4} Re:{5} w:{6} TOL:{7}
iteraciones:{8}
tiempo:{9}\n".format(Nx,Ny,Lx,Ly,h,Re,w,TOL,simulacion[1],simulacion[2])
                myfile.write(logText)
                myfile.close()
            self.listaArchivos.insert(tk.END, simulacion[0])
    return

def correrCi(self):
    Nx, Ny, Lx, Ly, w, Re, h, TOL = self.getVariables()
    if (Nx > 300 or Nx < 20) or (Ny > 100 and Ny < 20):
        self.etiquetaSimular = tk.Label(text="La malla no es adecuada debido a su
tamano [" + self.entradaNx.get() + "," + self.entradaLy.get() + "]", fg="blue",
background="yellow").place(x=10, y=180)
    elif Lx > Nx / 5 or Ly > Ny / 3:
        self.etiquetaSimular = tk.Label(text="El tamano del objeto [" +
self.entradaLx.get() + "," + self.entradaLy.get() + "] no es adecuada su malla [" +
self.entradaNx.get() + "," + self.entradaNy.get() + "]",
fg="blue", background="yellow").place(x=10,
y=180)
    else:
        try:
            nombreArchivo =
self.listaArchivos.get(self.listaArchivos.curselection())
        except:
            self.etiquetaSimular = tk.Label(text="No ha seleccionado un archivo",
fg="blue", background="yellow").place(x=10, y=180)
            return
        print nombreArchivo
        nxArchivo = int(nombreArchivo[:3])
        nyArchivo = int(nombreArchivo[4:7])
        lxArchivo = int(nombreArchivo[8:10])
        lyArchivo = int(nombreArchivo[11:13])
        hArchivo = float(nombreArchivo[21:25])
        if (nxArchivo != Nx or nyArchivo != Ny) or (lxArchivo != Lx or lyArchivo
!= Ly) or (hArchivo != h):
            self.etiquetaSimular = tk.Label(text="Las condiciones iniciales no
son validas para realizar simular", fg="blue", background="yellow").place(x=10,
y=180)

```

```

        return
        lin, vor, _, _ = self.leerArchivo(nombreArchivo)
        sim = CSimulacion.CSimulacion(Nx, Ny, Lx, Ly, w, Re, h, TOL, lin, vor)
        simulacion=sim.simular()
        if sim == "error":
            self.etiquetaSimular = tk.Label(text="El metodo no converge para Re="
+ self.entradaNx.get() + " y w=" + self.entradaNx.get(), fg="blue",
background="yellow").place(x=10, y=180)
        else:
            self.etiquetaSimular = tk.Label(text="Simulacion correctamente
realizada en " + str(simulacion[2]) + " segundos y \n" + str(simulacion[1]) + "
pasos", fg="blue", background="yellow").place(x=10, y=180)
            self.listaArchivos.insert(tk.END, simulacion[0])
        return

def main():
    ventana = tk.Tk()
    ventana.geometry("430x430+100+100")
    ventana.title("Parametros del problema")
    CAplicacion(ventana)
    ventana.mainloop()

if __name__ == "__main__":
    main()

```

```
'''
```

```
CSimulacion
```

```
Modulo principal de simulación. En él se implementan todos los métodos de resolución del problema.
```

```
@author: Jaime
```

```
'''
```

```
import time
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt; from matplotlib import cm
```

```
class CSimulacion(object):
```

```
'''
```

```
Classdocs
```

```
'''
```

```
def __init__(self, nx, ny, lx, ly, w, Re, h, TOL, linCi=None, vorCi=None): #quito
```

```
L
```

```
'''
```

```
Constructor
```

```
'''
```

```
self.nx = nx
```

```
self.ny = ny
```

```
self.lx = lx
```

```
self.ly = ly
```

```
self.L = lx
```

```
self.h = h
```

```
self.hestrella=h/lx
```

```
self.Nx=int(nx/h)
```

```
self.Ny=int(ny/h)
```

```
self.lxh=lx/h
```

```
self.lyh=ly/h
```

```
self.puntosx = np.linspace(0, float(nx)/float(lx), self.Nx)
```

```
self.puntosy = np.linspace(0, float(ny)/float(lx), self.Ny)
```

```
self.w = w
```

```
self.Re = Re
```

```
self.TOL = TOL
```

```
self.lin = linCi
```

```

self.vor = vorCi
self.mascara = []

self.presion = None
self.velocidad = []
self.u = []
self.v = []
self.derivadax = []
self.derivaday = []
self.derivadaxx = []
self.derivadaxy = []
self.derivadaxxy = []

def condicionesIniciales(self):
    """
    Metodo donde se crean La mascara y, en caso de ser necesario, se crean unos
    valores iniciales para La vorticidad y La funcion flujo
    """

    #
    def crearLinVor():
        """
        Metodo, dentro de condicionesIniciales, para La nicializacion o arranque
        de La funcion de corriente y La vorticidad
        """
        self.vor = np.zeros((Nx, Ny))
        lin = np.ones((Nx, Ny))
        for i in range(Nx):
            for j in range(Ny):
                if self.mascara[i, j] != np.nan:
                    lin[i, j] = self.puntosy[j] # Importante para garantizar la
convergenia
        self.lin = lin

    # Inicializacion o arranque de la presion
    def crearPresion():
        self.pres = np.ones((Nx, Ny))

    def crearMascara():
        """
        Metodo, dentro de condicionesIniciales, para La creacion de La mascara
        con un objeto
        Returns:
            mascara (np.darray): La mascara con La que se simula.
        """
        mascara = np.zeros((Nx, Ny))

        for i in range(Nx):
            for j in range(Ny):
                # CONTORNOS LATERALES Y SUPERIOR DE LA MALLA
                if i == 0 and j == (Ny - 1):mascara[i, j] = 1.0 # ESQUINA
SUPERIOR IZQUIERDA A+B
                elif i == (Nx - 1) and j == (Ny - 1):mascara[i, j] = 2.0 #
ESQUINA SUPERIOR DERECHA B+C
                elif j == (Ny - 1) :mascara[i, j] = 3.0 # CONTORNO SUPERIOR B
                elif i == 0 and j == 0: mascara[i, j] = 4.0 # ESQUINA INFERIOR
IZQUIERDA A+D
                elif i == 0: mascara[i, j] = 5.0 # CONTORNO IZQUIERDO A
                elif i == (Nx - 1) and j == 0 :mascara[i, j] = 6.0 # ESQUINA
INFERIOR DERECHA C+E
                elif i == (Nx - 1): mascara[i, j] = 7.0 # CONTORNO DERECHO C
                # OBJETO
                elif i == po and j == 0: mascara[i, j] = 8.0 # ESQUINA INFERIOR
IZQUIERDA OBJETO D+F

```

```

        elif i == po and j == lyh: mascara[i, j] = 9.0 # ESQUINA
SUPERIOR IZQUIERDA OBJETO    F+G
        elif i == po and j < lyh: mascara[i, j] = 10.0 # LATERAL
IZQUIERDO DEL OBJETO    F
        elif i == po + lxh and j == 0: mascara[i, j] = 11.0 # ESQUINA
INFERIOR DERECHA OBJETO    E+H
        elif i == po + lxh and j == lyh: mascara[i, j] = 12.0 # ESQUINA
SUPERIOR DERECHA OBJETO    G+H
        elif i == po + lxh and j < lyh: mascara[i, j] = 13.0 # LATERAL
DERECHO DEL OBJETO    H
        elif j == lyh and (i > po + 1 and i < po + lxh): mascara[i, j] =
14.0 # PARTE SUPERIOR DEL OBJETO    G
        # CONTORNOS INFERIORES
        elif j == 0 and (i > 0 and i < po):mascara[i, j] = 15.0 #
CONTORNO INFERIOR ANTERIOR AL OBJETO    D
        elif j == 0 and (i > po + lxh and i < Nx - 1):mascara[i, j] =
16.0 # CONTORNO INFERIOR POSTERIOR AL OBJETO    E
        # INTERIOR DEL OBJETO
        elif j < lyh and (i > po and i < po + lxh): mascara[i, j] =
np.nan # INTERIOR DEL OBJETO

```

```

self.mascara = mascara

```

```

Nx = self.Nx
Ny = self.Ny
lxh = self.lxh
lyh = self.lyh

```

```

#Posicion del obstaculo. Lo coloco a 1/3 del grid, en la posicion que sea un
entero de la longitud caracteristica del objeto
#de esta manera estara situado a una cierto multiplo de Lx del origen
po = float(self.lxh*int(Nx/self.lxh/3))

```

```

# Clasificacion de los puntos de la malla. Mascara.

```

```

crearMascara()
# Si no he arrancado con unas CI dejo lin y vor a None
if self.lin == None or self.vor == None:
    crearLinVor()
#     if self.pres == None:
crearPresion()

```

```

# METODO DE RELAJACION DE LAS LINEAS DE CORRIENTE

```

```

def relajacionLineas(self, vor, lin):
    """
    METODO DE RELAJACION DE LAS LINEAS DE CORRIENTE
    Args:
        vor: Matriz de vorticidad en k
        lin: Matriz de funcion corriente en k
        w: Parametro de relajacion
    Returns: Matriz de funcion corriente en k+1
    """
    mascara = self.mascara
    w = self.w
    hes=self.hestrella
    hes2=hes*hes
    Nx = self.Nx
    Ny = self.Ny
    prueba = np.zeros((Nx, Ny))
    for i in range(Nx):
        for j in range(Ny):

```

```

#CONTORNOS LATERALES Y SUPERIOR DE LA MALLA
    if mascara[i, j] == 5.0: # CONTORNO IZQUIERDO(F)
        prueba [i,j] = lin[i+1,j]

    elif mascara[i, j] == 3.0: # CONTORNO SUPERIOR(G)
        prueba[i,j] = lin[i,j-1]+ hes

    elif mascara[i, j] == 7.0: # CONTORNO DERECHO(H)
        prueba[i,j] = lin[i-1,j]

#ESQUINAS SUPERIORES DE LA MALLA
    elif mascara[i, j] == 1.0: # ESQUINA SUPERIOR IZQUIERDA
        prueba[i,j] = lin[i+1,j]

    elif mascara[i, j] == 2.0: # ESQUINA SUPERIOR DERECHA
        prueba[i,j] = lin[i-1,j]

#CONTORNO INFERIOR DE LA MALLA Y EL OBJETO
    elif mascara[i, j] in
(4.0,6.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0,16.0): # CONTORNO INFERIOR Y OBJETO
        prueba[i, j] = 0.0

#INTERIOR DE LA MALLA
    elif mascara[i, j] == 0.0: # INTERIOR DE LA MALLA
        prueba[i, j] = (lin[i + 1, j] + lin[i - 1, j] + lin[i, j + 1] +
lin[i, j - 1] - hes2*vor[i, j]) / 4.0

return w * prueba + lin * (1.0 - w)

# METODO DE RELAJACION DE LA VORTICIDAD
def relajacionVorticidad(self, vor, lin):
    """
    METODO DE RELAJACION DE LA VORTICIDAD
    mascara: Mascara de definicion de Los distintos puntos de La malla
    vor: Matriz de vorticidad en k
    lin: Matriz de funcion corriente en k+1
    w: Parametro de relajacion
    Re: Numero de Reynolds de Red
    return: Matriz de vorticidad en k+1
    """
    mascara = self.mascara
    w = self.w
    Re = self.Re
    hes=self.hestrella
    hes2=hes*hes
    Nx = self.Nx
    Ny = self.Ny
    temporal = np.zeros((Nx, Ny))
    Re16 = Re / 16.0
    for i in range(Nx):
        for j in range(Ny):

#CONTORNOS LATERALES Y SUPERIOR DE LA MALLA

            if mascara[i, j] in (1.0,2.0,3.0,5.0): # CONTORNOS SUPERIOR(B) E
IZQUIERDO (A) DE LA MALLA
                temporal[i, j] = 0.0

            elif mascara[i, j] == 7.0: # CONTORNO DERECHO(H)
#
                temporal[i, j] = vor[i-1,j] #condicion de contorno: primera
derivada vorticidad en x = 0
                temporal[i,j] = 2*vor[i-1,j]-vor[i-2,j] #condicion de contorno:
segunda derivada vorticidad en x = 0
#
                temporal[i, j] = (2.0 * vor[i - 1, j] + vor[i, j + 1] + vor[i,
j - 1]) / 4.0

```

```

#CONTORNO INFERIOR DE LA MALLA
elif mascara[i, j] in (4.0,6.0,8.0,11.0,15.0,16.0): # CONTORNO
INFERIOR DE LA MALLA (D) Y (E)
    temporal[i, j] = 0.0

#CONTORNO DEL OBJETO
elif mascara[i, j] == 10.0: # LATERAL IZQUIERDO DEL OBJETO (F)
    temporal[i, j] = 2.0 * lin[i - 1, j]/hes2
elif mascara[i, j] == 13.0: # LATERAL DERECHO DEL OBJETO (H)
    temporal[i, j] = 2.0 * lin[i + 1, j]/hes2
elif mascara[i, j] == 14.0 : # PARTE SUPERIOR DEL OBJETO (G)
    temporal[i, j] = 2.0 * lin[i, j + 1]/hes2

#ESQUINAS DEL OBJETO

elif mascara[i, j] ==9.0: #ESQUINA F-G
#
    temporal[i,j] = 2.0 * lin[i - 1, j]/hes2 #CONDICION DE F
#
    temporal[i, j] = 2.0 * lin[i, j + 1]/hes2 #CONDICION DE G
MIXTA
    temporal[i, j] = (lin[i, j + 1] + lin[i - 1, j])/hes2 #CONDICION

elif mascara[i, j] == 12.0: #ESQUINA H-G
#
    temporal[i, j] = 2.0 * lin[i + 1, j]/hes2 #CONDICION DE H
#
    temporal[i, j] = 2.0 * lin[i, j + 1]/hes2 #CONDICION DE G
MIXTA
    temporal[i, j] = (lin[i, j + 1] + lin[i + 1, j])/hes2 #CONDICION

#INTERIOR DE LA MALLA

elif mascara[i, j] == 0.0: # INTERIOR DE LA MALLA
    temporal[i, j] = (vor[i + 1, j] + vor[i - 1, j] + vor[i, j + 1] +
vor[i, j - 1]) / 4.0
    temporal[i, j] -= Re16 * ((lin[i, j + 1] - lin[i, j - 1]) *
(vor[i + 1, j] - vor[i - 1, j]) - (lin[i + 1, j] - lin[i - 1, j]) * (vor[i, j + 1] -
vor[i, j - 1]))

return w * temporal + vor * (1.0 - w)

def calculoPresionAparte(self,presion):
    """
    METODO DE RELAJACION DE LA PRESION
    mascara: Mascara de definicion de los distintos puntos de la malla
    pres: Matriz de presion en k
    lin: Matriz de lineas de corriente en k+1
    return: Matriz de presion en k+1
    """
    mascara = self.mascara
    w = self.w
    Re = self.Re
    hes=self.hestrella
    hes2=hes*hes
    Nx = self.Nx
    Ny = self.Ny
    vor = self.vor
    lin = self.lin
    Eu = 1.0
    C = 1.0/(Re*Eu)

#
    temporal = np.zeros((Nx, Ny))
    for i in range(Nx):
        for j in range(Ny):

```

```

#CONTORNOS LATERALES Y SUPERIOR DE LA MALLA

    if mascara[i, j] == 5.0: # CONTORNO IZQUIERDO(A)
        temporal[i,j] = presion[i+1,j]+(lin[i+1,j+1]-lin[i,j+1]-
lin[i+1,j]+lin[i,j]))/(hes*Eu)

    elif mascara[i, j] == 3.0: # CONTORNO SUPERIOR(B)
        temporal[i,j] = presion[i,j-1]

    elif mascara[i, j] == 7.0: # CONTORNO DERECHO(C)
        temporal[i,j] = presion[i-1,j]+C*(vor[i,j+1]-vor[i,j])

#ESQUINAS SUPERIORES DE LA MALLA
    elif mascara[i,j] == 1.0: # ESQUINA SUPERIOR IZQUIERDA
        temporal[i,j] = presion[i,j-1]
    elif mascara[i,j] == 2.0: # ESQUINA SUPERIOR DERECHA
#        temporal[i,j] = presion[i,j-1]
        temporal[i,j] = presion[i-1,j]+C*(vor[i,j]-vor[i,j-1])

#CONTORNO INFERIOR DE LA MALLA
    elif mascara[i, j] == 15.0 : # CONTORNO INFERIOR ANTERIOR AL OBJETO
(D)
        temporal[i,j] = presion[i,j+1]
    elif mascara[i, j] == 16.0 : # CONTORNO INFERIOR POSTERIOR AL OBJETO
(E)
        temporal[i,j] = presion[i,j+1]

    elif mascara[i,j] == 8.0: # ESQUINA INFERIOR IZQUIERDA OBJETO
        temporal[i,j] = presion[i-1,j]+C*(vor[i,j+1]-vor[i,j])
    elif mascara[i,j] == 11.0: # ESQUINA INFERIOR DERECHA OBJETO
        temporal[i,j] = presion[i+1,j]-C*(vor[i,j+1]-vor[i,j])

#ESQUINAS INFERIORES DE LA MALLA
    elif mascara[i,j] == 4.0: # ESQUINA INFERIOR IZQUIERDA
        temporal[i,j] = presion[i,j+1]

    elif mascara[i,j] == 6.0: # ESQUINA INFERIOR DERECHA
        temporal[i,j] = presion[i-1,j]+C*(vor[i,j+1]-vor[i,j]) #misma
condicion que en (C)

#CONTORNO DEL OBJETO
    elif mascara[i, j] == 10.0: # LATERAL IZQUIERDO DEL OBJETO (F)
        temporal[i,j] = presion[i-1,j]+C*(vor[i,j+1]-vor[i,j-1])/2.0

    elif mascara[i,j] == 13.0: # LATERAL DERECHO DEL OBJETO (H)
        temporal[i,j] = presion[i+1,j]-C*(vor[i,j+1]-vor[i,j-1])/2.0

    elif mascara[i, j] == 14.0 : # PARTE SUPERIOR DEL OBJETO (G)
        temporal[i,j] = presion[i,j+1]+C*(vor[i+1,j]-vor[i-1,j])/2.0
#ESQUINAS DEL OBJETO
    elif mascara[i, j] == 9.0: # ESQUINA SUPERIOR IZQUIERDA OBJETO
        temporal[i,j] = presion[i-1,j]+C*(vor[i,j+1]-vor[i,j-1])/2.0 #
condicion de F
    elif mascara[i,j] == 12.0: # ESQUINA SUPERIOR DERECHA OBJETO
        temporal[i,j] = presion[i+1,j]-C*(vor[i,j+1]-vor[i,j-1])/2.0 #
condicion de H

#INTERIOR DE LA MALLA

    elif mascara[i,j] == 0.0: #INTERIOR
        temporal[i,j] = (presion[i+1,j] + presion[i-1,j] + presion[i,j+1]
+ presion[i,j-1])/4.0
        temporal[i,j] -= (lin[i+1,j] + lin[i-1,j] - 2 * lin[i,j]) *
(lin[i,j+1] + lin[i,j-1] - 2 * lin[i,j])/(2.0*hes2*Eu)

```

```

        temporal[i,j] += ((lin[i+1,j+1] - lin[i+1,j-1] - lin[i-1,j+1] +
lin[i-1,j-1])*(lin[i+1,j+1] - lin[i+1,j-1] - lin[i-1,j+1] + lin[i-1,j-
1]))/(2.0*hes2*Eu)

```

```
#
```

```
    return w * temporal + presion * (1.0 - w)
```

```
def calculoVelocidad(self):
```

```
    """
```

```
    METODO DE CALCULO DE LA VELOCIDAD EN EL INTERIOR DEL DOMINIO
```

```
    mascara: Mascara de definicion de Los distintos puntos de la malla
```

```
    lin: Matriz de Lineas de corriente en k
```

```
    return: Matriz de velocidad en k
```

```
    """
```

```
    Nx = self.Nx
```

```
    Ny = self.Ny
```

```
    mask = self.mascara
```

```
    lin = self.lin
```

```
    hes = self.hestrella
```

```
    velocidad=np.zeros((Nx,Ny))
```

```
    u=np.zeros((Nx,Ny))
```

```
    v=np.zeros((Nx,Ny))
```

```
    for i in range(Nx):
```

```
        for j in range(Ny):
```

```
            if mask[i,j] == 0.0:
```

```
                v[i,j] = -(lin[i+1,j]-lin[i-1,j])/(2*hes)
```

```
                u[i,j] = (lin[i,j+1]-lin[i,j-1])/(2*hes)
```

```
                velocidad[i,j]=u[i,j]*u[i,j]+v[i,j]*v[i,j]
```

```
                velocidad[i,j]=np.sqrt(velocidad[i,j])
```

```
            elif mask[i,j] in (15.0,16.0):
```

```
                v[i,j] = -(lin[i+1,j]-lin[i-1,j])/(2*hes)
```

```
                u[i,j] = (lin[i,j+1]-lin[i,j])/(hes)
```

```
                velocidad[i,j]=u[i,j]*u[i,j]+v[i,j]*v[i,j]
```

```
                velocidad[i,j]=np.sqrt(velocidad[i,j])
```

```
    self.u = u
```

```
    self.v = v
```

```
    self.velocidad = velocidad
```

```
def calculoPrimerasDerivadas(self):
```

```
    """
```

```
    METODO DE CALCULO DE LA VELOCIDAD EN EL INTERIOR DEL DOMINIO
```

```
    mascara: Mascara de definicion de Los distintos puntos de la malla
```

```
    lin: Matriz de Lineas de corriente en k
```

```
    return: Matriz de velocidad en k
```

```
    """
```

```
    Nx = self.Nx
```

```
    Ny = self.Ny
```

```
    mask = self.mascara
```

```
    lin = self.lin
```

```
    hes = self.hestrella
```

```
    derivadax=np.zeros((Nx,Ny))
```

```
    derivaday=np.zeros((Nx,Ny))
```

```
    for i in range(Nx):
```

```
        for j in range(Ny):
```

```
            if mask[i,j] == 0.0:
```

```
                derivadax[i,j] = (lin[i+1,j]-lin[i-1,j])/(2*hes)
```

```
                derivaday[i,j] = (lin[i,j+1]-lin[i,j-1])/(2*hes)
```

```
            elif mask[i,j] == 1.0:
```

```
                derivadax[i,j] = (lin[i+1,j]-lin[i,j])/(hes)
```

```
                derivaday[i,j] = (lin[i,j]-lin[i,j-1])/(hes)
```

```
            elif mask[i,j] == 2.0:
```

```
                derivadax[i,j] = (lin[i,j]-lin[i-1,j])/(hes)
```

```
                derivaday[i,j] = (lin[i,j]-lin[i,j-1])/(hes)
```

```

elif mask[i,j] == 3.0:
    derivadax[i,j] = (lin[i+1,j]-lin[i-1,j])/(2*hes)
    derivaday[i,j] = (lin[i,j]-lin[i,j-1])/(hes)
elif mask[i,j] == 4.0:
    derivadax[i,j] = (lin[i+1,j]-lin[i,j])/(hes)
    derivaday[i,j] = (lin[i,j+1]-lin[i,j])/(hes)
elif mask[i,j] == 5.0:
    derivadax[i,j] = (lin[i+1,j]-lin[i,j])/(hes)
    derivaday[i,j] = (lin[i,j+1]-lin[i,j-1])/(2*hes)
elif mask[i,j] == 6.0:
    derivadax[i,j] = (lin[i,j]-lin[i-1,j])/(hes)
    derivaday[i,j] = (lin[i,j+1]-lin[i,j])/(hes)
elif mask[i,j] == 7.0:
    derivadax[i,j] = (lin[i,j]-lin[i-1,j])/(hes)
    derivaday[i,j] = (lin[i,j+1]-lin[i,j-1])/(2*hes)
elif mask[i,j] == 8.0:
    derivadax[i,j] = (lin[i,j]-lin[i-1,j])/(hes)
    derivaday[i,j] = (lin[i,j+1]-lin[i,j])/(hes)
elif mask[i,j] == 9.0:
    derivadax[i,j] = (lin[i+1,j]-lin[i-1,j])/(2*hes)
    derivaday[i,j] = (lin[i,j+1]-lin[i,j-1])/(2*hes)
elif mask[i,j] == 10.0:
    derivadax[i,j] = (lin[i,j]-lin[i-1,j])/(hes)
    derivaday[i,j] = (lin[i,j+1]-lin[i,j-1])/(2*hes)
elif mask[i,j] == 11.0:
    derivadax[i,j] = (lin[i+1,j]-lin[i,j])/(hes)
    derivaday[i,j] = (lin[i,j+1]-lin[i,j])/(hes)
elif mask[i,j] == 12.0:
    derivadax[i,j] = (lin[i+1,j]-lin[i-1,j])/(2*hes)
    derivaday[i,j] = (lin[i,j+1]-lin[i,j-1])/(2*hes)
elif mask[i,j] == 13.0:
    derivadax[i,j] = (lin[i+1,j]-lin[i,j])/(hes)
    derivaday[i,j] = (lin[i,j+1]-lin[i,j-1])/(2*hes)
elif mask[i,j] == 14.0:
    derivadax[i,j] = (lin[i+1,j]-lin[i-1,j])/(2*hes)
    derivaday[i,j] = (lin[i,j+1]-lin[i,j])/(hes)
elif mask[i,j] == 15.0:
    derivadax[i,j] = (lin[i+1,j]-lin[i-1,j])/(2*hes)
    derivaday[i,j] = (lin[i,j+1]-lin[i,j])/(hes)
elif mask[i,j] == 16.0:
    derivadax[i,j] = (lin[i+1,j]-lin[i-1,j])/(2*hes)
    derivaday[i,j] = (lin[i,j+1]-lin[i,j])/(hes)

return derivaday,-derivadax

```

```

def simular(self):
    """
    Metodo en el que se realiza la simulacion completa
    Al finalizar la simulacion guardamos los resultados en un archivo en forma de
    np.darray's
    Results:
        Archivo con los siguientes elementos:
        vor: Matriz de vorticidad una vez alcanzada la TOL requerida
        lin: Matriz de funcion corriente una vez alcanzada la TOL requerida
        mascara (np.darray): La mascara con la que se simula.
        mascaraContorno (np.darray): Mascara con la que se puede representar.
    El contorno
        del objeto ha sido puesto a nan, para mejorar la visualizacion de
    los campos
    """
    t1 = time.time()
    self.condicionesIniciales()
    linprev = 0
    vorprev = 1

```

```

    presprev = 1
    i = 0
    lin = self.lin
    vor = self.vor
    pres = self.pres
    while np.max(np.abs(lin - linprev))/np.max(np.abs(lin)) > self.TOL or
np.max(np.abs(vor - vorprev))/np.max(np.abs(vor)) > self.TOL:
        print "|Lin-Linprev|={0}\n|vor-vorprev|={1}".format(np.max(np.abs(lin -
linprev))/np.max(np.abs(lin)), np.max(np.abs(vor - vorprev))/np.max(np.abs(vor)))
        if np.max(np.abs(lin - linprev)) > 1000:
            return "error"
        i += 1
        print "Numero de iteracion", i
        vorprev = vor
        linprev = lin
        lin = self.relajacionLineas(vor, lin)
        vor = self.relajacionVorticidad(vor, lin)

self.lin = lin
self.vor = vor

self.calculoVelocidad()
uprueba,vprueba = self.calculoPrimerasDerivadas()
self.derivaday = uprueba
self.derivadax = -vprueba
#####PARA CALCULAR LA PRESION EN CASO DE INTERESARME
#     j=0
#     while np.max(np.abs(pres - presprev))/np.max(np.abs(pres)) > 1e-05 or
j<10:
#         presprev = pres
#         pres = self.calculoPresionAparte(pres)
#
#         j+=1
#     self.pres = pres
#####
if self.nx < 100: strnx = "0" + str(self.nx)
else: strnx = str(self.nx)
if self.ny < 100: strny = "0" + str(self.ny)
else: strny = str(self.ny)
if self.lx < 10: strlx = "0" + str(self.lx)
else: strlx = str(self.lx)
if self.ly < 10: strly = "0" + str(self.ly)
else: strly = str(self.ly)
if self.L <10: strL = "0" + str(self.L)
else: strL = str(self.L)
if self.h < 10: strh = "0" + str(self.h)
else: strh = str(self.h)
print strh
if len(strh) == 4: strh += "0"
print strh

NOMBREARCHIVO = "{0}x{1}_{2}x{3}_L_{4}_h_{5}_Re_{6}_w_{7}.dat".format(strnx,
strny, strlx, strly, strL, strh, str(self.Re), str(self.w))
f = open(NOMBREARCHIVO,"wb")
print NOMBREARCHIVO

#Para quitar:
self.mascaraContorno = self.mascara

np.savez(f, lin, vor, self.mascara, self.mascaraContorno, self.puntosx
,self.puntosy, self.velocidad, self.u, self.v, self.pres)

f.close()
return NOMBREARCHIVO, i, np.round((time.time() - t1), decimals=2)

```

```

'''
CPintar
Archivo básico de representación

@author: Jaime
'''
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.colors import LogNorm

class CPintar(object):
    '''
    Classdocs
    '''

    def __init__(self, nombreArchivo):
        '''
        Constructor
        '''
        self.nombreArchivo=nombreArchivo
        self.Nx = int(nombreArchivo[0:3])
        self.Lx = int(nombreArchivo[8:10])
        self.Re=nombreArchivo[31:34]
        self.lin=[]
        self.vor=[]
        self.pres=[]
        self.mask=[]
        self.mask2=[]
        self.h = float(nombreArchivo[21:25])

    def leerArchivo(self, nombreArchivo):
        npzfile = np.load(nombreArchivo)
        self.lin = npzfile['arr_0']
        self.vor = npzfile['arr_1']
        self.mask = npzfile['arr_2']
        self.mask2 = npzfile['arr_3']
        self.puntosx = npzfile['arr_4']
        self.puntosy = npzfile['arr_5']
        self.velocidad = npzfile['arr_6']
        self.componenteu = npzfile['arr_7']
        self.componentev = npzfile['arr_8']
        self.pres = npzfile['arr_9']

    # REPRESENTACION
    def plot(self, camposAPintar):
        mask=self.mask

        representarLin, representarVor, representarPres, representarVel =
camposAPintar
        numeroCamposPintar = sum(camposAPintar)
        numFilas = 1
        numColumnas = 1
        contador = 1

        if numeroCamposPintar > 1:
            numFilas = 2
        if numeroCamposPintar > 2:
            numColumnas = 2

        xobjeto = int(self.Nx/self.Lx/3)
        xlenobjeto = 1.0
        medioPaso = 1.0/self.Lx/2.0*self.h

```

```

        extentneg = [-medioPaso,max(self.puntosx)-medioPaso,+medioPaso, -
max(self.puntosy)+medioPaso]
        extent = [-medioPaso,max(self.puntosx)-medioPaso, -
medioPaso,max(self.puntosy)-medioPaso]

        X, Y = np.meshgrid(self.puntosy, self.puntosx-medioPaso*3.0/4.0)

        fig = plt.figure()
        fig.set_size_inches(24,6)

        def graficaPuntos(campo ,signo = '+', cmap = None):
            # La paleta de colores por defecto es cmap2=plt.cm.jet
            cmap2 = plt.cm.jet_r
            if signo == '-':
                return ax.imshow(campo.T, cmap = cmap2,
interpolation='none',extent=extentneg,origin = 'Lower')
            else:
                return ax.imshow(campo.T, cmap = cmap,
interpolation='none',extent=extent,origin = 'Lower')

        def graficaLineas(campo1,campo2,signo = '+'):
            if signo == '-':
                return plt.streamplot(self.puntosx, -self.puntosy, campo1.T, -
campo2.T,density=0.5,linewidth=2,arrowsize =2, color='w')
            else:
                return
plt.streamplot(self.puntosx, self.puntosy, campo1.T, campo2.T, density=0.5,linewidth=2,ar
rowsize =2, color='w')

        def graficaContornosRellenos(campo, niveles = None, cmap = None, signo = '+'):
            if signo == '-':
                return plt.contourf(campo.T,extent=extentneg,origin
='Lower',levels=niveles,cmap = cmap,)
            else:
                return plt.contourf(campo.T,extent=extent,origin
='Lower',levels=niveles,cmap = cmap,)

        def graficaContornos(campo, niveles = None, signo = '+'):
            if signo == '-':
                return plt.contour(campo.T,extent=extentneg,origin
='Lower',levels=niveles,linewidth=2,colors = 'k')
            else:
                return plt.contour(campo.T,extent=extent,origin
='Lower',levels=niveles,linewidth=2,colors = 'k')

        def graficaVectores(campo1,campo2,n=1):
            return plt.quiver(Y[:, :n],X[:, :n],campo1[:, :n],campo2[:, :n,
:], angles='xy', scale_units='xy',scale =1.0)

##### FUNCION DE CORRIENTE
#####
        if representarLin:
            lin = self.lin
            u = self.componenteu
            v = self.componentev
            lin[np.isnan(mask)] = np.nan
            u[np.isnan(mask)] = np.nan
            v[np.isnan(mask)] = np.nan

            ax = fig.add_subplot(numFilas,numColumnas,contador)
            ax.set_aspect(1)

```

```

__#         # pintamos el obstaculo
           # add_patch no permite definir un objeto para todas las subplots, por lo
           # que habra que ir definiendolo grafica a grafica
           rect = plt.Rectangle((xobjeto, -0.5), xlenobjeto,1.0,alpha=1.0,color =
'r')
           ax.add_patch(rect)

           # elegimos las graficas a pintar
           CS = graficaLineas(u,v)
           CS = graficaPuntos(lin)
           plt.colorbar(CS)
           plt.axis([-medioPaso,max(self.puntosx)-medioPaso,0,max(self.puntosy)-
medioPaso])

           # actualizamos el contador
           contador += 1

# ##### VORTICIDAD
#####
           if representarVor:
               vor = self.vor
               vor[np.isnan(mask)] = np.nan
               vor[np.isnan(mask)] = np.nan

               print "vor",vor
               ax = fig.add_subplot(numFilas, numColumnas, contador)
               ax.set_aspect(1)

#         # pintamos el obstaculo
'r')         rect = plt.Rectangle((xobjeto, -0.5), xlenobjeto, 1.0, alpha=1.0,color =

           ax.add_patch(rect)
           levels = [0.0, 0.1, 0.3, 0.5, 1.0, 1.5, 2.0, 3.0, 5.0,8.0,10.0,15.0]

           # elegimos las graficas a pintar
           CS = graficaContornos(vor, levels)
           plt.clabel(CS, inline=1, colors='k',fontsize=10)

           CS = graficaContornosRellenos(vor, levels, plt.cm.gist_yarg)
           plt.colorbar(CS)

           plt.axis([-medioPaso, max(self.puntosx) - medioPaso, 0, max(self.puntosy)
- medioPaso])

#         actualizamos el contador
           contador += 1

#
#         ##### PRESION
#####
           if representarPres:
               pres = self.pres
               pres[np.isnan(mask)] = np.nan
               print pres

               ax = fig.add_subplot(numFilas,numColumnas,contador)
               ax.set_aspect(1)

#         # pintamos el obstaculo
           rect = plt.Rectangle((xobjeto, -0.5), xlenobjeto,1.0,alpha=1.0,
facecolor='r')
           ax.add_patch(rect)

_____# elegimos las graficas a pintar
           nivelespres = [-60,-40,-25,-15,-10, -5, -2, -1, 0, 1,1.1,1.2, 1.5, 2,
5,8, 10,12, 15, 25, 40]

```

```

        CS = graficaContornos(pres,nivelespres)

        plt.clabel(CS, inline=2, fontsize=8)
        CS = graficaPuntos(pres)

        plt.colorbar(CS)
        plt.axis([-medioPaso,max(self.puntosx)-medioPaso,0,max(self.puntosy)-
medioPaso])

_____ # actualizamos el contador
        contador += 1

#
# ##### VELOCIDAD
#####
    if representarVel:
        u = self.componenteu
        v = self.componentev
        velocidad = np.sqrt(u**2+v**2)
        u[np.isnan(mask)] = np.nan
        v[np.isnan(mask)] = np.nan
        velocidad[np.isnan(mask)] = np.nan

        ax = fig.add_subplot(numFilas,numColumnas,contador)
        ax.set_aspect(1)

        # # pintamos el obstaculo
        rect = plt.Rectangle((xobjeto, 0), xlenobjeto,0.5,alpha=1.0,color='r')

        ax.add_patch(rect)

        # elegimos las graficas a pintar

        CS = graficaVectores(u,v,2)

_____ # actualizamos el contador
        contador += 1

        plt.xlabel("x*", fontsize=14)
        plt.ylabel("y*", fontsize=14)

#
#####
#####
        nombreSalvado=self.nombreArchivo[:-4]+".png"
        plt.savefig(nombreSalvado, bbox_inches=0)
        plt.show()

```

## 9.2. Código del método de Boltzmann

```

#!/usr/bin/python
# Copyright (C) 2013 FlowKit Ltd, Lausanne, Switzerland
# E-mail contact: contact@flowkit.com
#
# This program is free software: you can redistribute it and/or
# modify it under the terms of the GNU General Public License, either
# version 3 of the License, or (at your option) any later version.
#

```

```

# 2D flow around a cylinder
#

from numpy import *; from numpy.linalg import *
import matplotlib.pyplot as plt; from matplotlib import cm
##### Flow definition #####
maxIter = 21000 # Total number of time iterations.
Re = 8.0 # Reynolds number.
## IMPORTANTE!!! Este numero de Reynolds debe multiplicarse por 2 para que sea el
#mismo que el de nuestro problema.
# nx = 520; ny = 180; ly=ny-1.0; q = 9 # Lattice dimensions and populations.
# nx = 390; ny = 135; ly=ny-1.0; q = 9 # Lattice dimensions and populations.
# cx = nx/4; cy=ny/2; r=ny/9; # Coordinates of the cylinder.
nx = 240; ny = 127; ly=ny-1.0; q = 9 # Lattice dimensions and populations.
cx = nx/3+5; cy=ny/2; r=8; # Coordinates of the cylinder.
uLB = 0.04 # Velocity in lattice units.
nulb = uLB*r/Re; omega = 1.0 / (3.*nulb+0.5); # Relaxation parameter.

##### Lattice Constants #####
c = array([(x,y) for x in [0,-1,1] for y in [0,-1,1]]) # Lattice velocities.
t = 1./36. * ones(q) # Lattice weights.
t[asarray([norm(ci)<1.1 for ci in c])] = 1./9.; t[0] = 4./9.
noslip = [c.tolist().index((-c[i]).tolist()) for i in range(q)]
print "noslip",noslip
i1 = arange(q)[asarray([ci[0]<0 for ci in c])] # Unknown on right wall.
i2 = arange(q)[asarray([ci[0]==0 for ci in c])] # Vertical middle.
i3 = arange(q)[asarray([ci[0]>0 for ci in c])] # Unknown on left wall.

##### Function Definitions #####
sumpop = lambda fin: sum(fin,axis=0) # Helper function for density computation.
def equilibrium(rho,u): # Equilibrium distribution function.
    cu = 3.0 * dot(c,u.transpose(1,0,2))
    usqr = 3./2.*(u[0]**2+u[1]**2)
    feq = zeros((q,nx,ny))
    for i in range(q): feq[i,:,:] = rho*t[i]*(1.+cu[i]+0.5*cu[i]**2-usqr)
    return feq

##### Setup: cylindrical obstacle and velocity inlet with perturbation #####
# obstacle = fromfunction(lambda x,y: (x-cx)**2+(y-cy)**2<r**2, (nx,ny))
obstacle = zeros((nx,ny), dtype=bool)
for i in range(nx):
    for j in range(ny):
        if (i>cx-r and i<cx+r) and (j>cy-r and j<cy+r): obstacle[i,j]=True

vel = fromfunction(lambda d,x,y: (1-d)*uLB*(1.0+1e-4*sin(y/ly*2*pi)),(2,nx,ny))
feq = equilibrium(1.0,vel); fin = feq.copy()

##### Main time loop #####
for time in range(maxIter):
    if (time%10==0): print time
    fin[i1,-1,:] = fin[i1,-2,:] # Right wall: outflow condition.
    rho = sumpop(fin) # Calculate macroscopic density and velocity.
    u = dot(c.transpose(), fin.transpose((1,0,2)))/rho

    u[:,0,:] = vel[:,0,:] # Left wall: compute density from known populations.
    rho[0,:] = 1./(1.-u[0,0,:]) * (sumpop(fin[i2,0,:])+2.*sumpop(fin[i1,0,:]))

    feq = equilibrium(rho,u) # Left wall: Zou/He boundary condition.
    fin[i3,0,:] = fin[i1,0,:] + feq[i3,0,:] - fin[i1,0,:]
    fout = fin - omega * (fin - feq) # Collision step.
    for i in range(q): fout[i,obstacle] = fin[noslip[i],obstacle]
    for i in range(q): # Streaming step.
        fin[i,:,:] = roll(roll(fout[i,:,:],c[i,0],axis=0),c[i,1],axis=1)

    if (time%5000==0) and time>10: # Visualization
        # plt.gca()
        fig = plt.figure()

```

```
fig.set_size_inches(nx/10,ny/10-2)
plt.gca()
# extent = [0,16,-4,4]
CS = plt.imshow(sqrt((u[0]*25.0)**2+(u[1]*25.0)**2).transpose(),cmap=cm.jet)
rect = plt.Rectangle((cx-r,cy-r), 2*r-0.5,2*r-0.5,alpha=1.0, color = 'r')
plt.gca().add_patch(rect)
plt.xlabel("i", fontsize=14)
plt.ylabel("j", fontsize=14)
plt.colorbar(CS)
plt.savefig("vel."+str(time/100).zfill(4)+".png")
```