

Simulación de los Vórtices de Von Kármán utilizando OpenFOAM y Computación Paralela

López V. Juan / Arroyo L. Juan / González V. Andrés / Rubiano V. Rubén
julopezvi, julopezar, agonzalezva, rrubiano - @unal.edu.co
Universidad Nacional de Colombia - Facultad de Ciencias

Resumen—En este estudio, se utilizó el software OpenFOAM para modelar el comportamiento de un fluido viscoso al pasar por un obstáculo cilíndrico, con el objetivo de estudiar los vórtices de Von Kármán. Para describir las condiciones de frontera, se realizó una discretización del dominio de interés y se generó una malla tridimensional detallada. Además, se implementó una distribución uniforme de velocidades como condición inicial. Para esto se llevó a cabo un análisis del costo computacional comparando la eficiencia del código mediante programación en serie y en paralelo. Los resultados demostraron que la parallelización reduce significativamente el tiempo de cálculo y aprovecha mejor los recursos disponibles del computador.

Palabras Clave—OpenFOAM, Von Karman, Parallelización, Dinámica de fluidos.

I. INTRODUCCIÓN

Este proyecto se centra en la simulación de los vórtices de Von Kármán utilizando herramientas como OpenFOAM y ParaView. El objetivo principal es estudiar el comportamiento de estos vórtices en un fluido, mejorar la eficiencia en el procesamiento de datos necesarios para la simulación, mediante el uso de computación paralela, y evaluar el rendimiento al variar el número de threads (hilos).

II. OBJETIVOS

1. Realizar una investigación y análisis del fenómeno físico de los vórtices de Von Kármán.
 - a) Identificar las condiciones físicas y parámetros relevantes para la formación de los vórtices.
2. Desarrollar una simulación del fenómeno físico utilizando OpenFOAM y Paraview.
 - a) Documentar las características y comportamiento del fenómeno.

3. Evaluar el rendimiento de la simulación cuando se ejecuta en un entorno de computación paralela.
 - a) Configurar un entorno de computación paralela utilizando herramientas como MPI.
 - b) Documentar los resultados obtenidos al realizar la parallelización de la simulación.

III. MARCO TEÓRICO

III-A. OpenFOAM

OpenFOAM es un software de código abierto utilizado para la simulación de dinámica de fluidos computacional (CFD). Se ha convertido en una herramienta estándar en la comunidad de CFD debido a su flexibilidad y capacidad para resolver una gran variedad de problemas.

Características principales:

- **Versatilidad:** OpenFOAM puede simular flujos de fluidos tanto compresibles como incompresibles, flujos turbulentos y laminares, flujos multifásicos y reacciones químicas.
- **Arquitectura de código abierto:** Esto permite a los usuarios modificar y personalizar el código según sus necesidades específicas, promoviendo la innovación y la colaboración en la comunidad científica y de ingeniería.
- **Biblioteca de solvers:** Incluye una amplia gama de solvers predefinidos para diferentes aplicaciones, como el flujo aerodinámico, la combustión, la transferencia de calor, entre otros.
- **Integración de herramientas:** OpenFOAM puede integrarse con otras herra-

mientas de análisis y post-procesamiento, como ParaView para la visualización de resultados. [1]

III-B. ParaView

ParaView es una aplicación de visualización de datos de código abierto que permite a los usuarios analizar y visualizar grandes conjuntos de datos.

Características principales:

- **Visualización de datos a gran escala:** ParaView es capaz de manejar grandes volúmenes de datos, lo que lo hace perfecto para la visualización de resultados de simulaciones CFD.
- **Interfaz de usuario:** Ofrece una interfaz gráfica intuitiva que permite a los usuarios crear visualizaciones complejas mediante la combinación de diferentes tipos de gráficos, cortes, y representaciones volumétricas.
- **Capacidades de scripting:** ParaView soporta el scripting en Python, lo que permite a los usuarios automatizar tareas repetitivas y crear visualizaciones personalizadas.
- **Compatibilidad:** Es compatible con numerosos formatos de archivo y puede integrarse fácilmente con otros software de simulación y análisis. [2]

III-C. Comportamiento de un fluido real sobre un cilindro

Cuando el flujo pasa cerca de la superficie de un obstáculo, experimenta un “frenado” debido a la fricción viscosa. La capa de fluido que experimenta este fenómeno se conoce como capa límite.

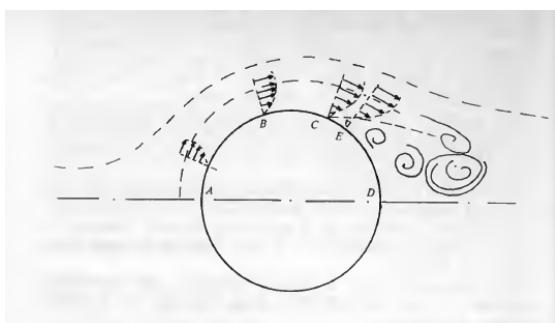


Figura 1: Capa límite en un cilindro

Cuando la velocidad de la capa límite es muy alta, está pasa de ser un flujo laminar a ser turbulento. Como se puede observar en la

imagen anterior, la capa límite se separa del obstáculo cuando está tiene una velocidad muy alta generando zonas de torbellinos detrás del mismo.

En un flujo real, cuando se presentan dos fuerzas sobre el cilindro, se genera resistencia de forma y resistencia de superficie, las cuales juntas componen la fuerza de arrastre F . La importancia relativa de estas resistencias depende del número de Reynolds del flujo. Lejos de los obstáculos, la estela y los torbellinos pueden disiparse, permitiendo que el flujo vuelva a comportarse como un flujo ideal. Es importante resaltar que número de Raynolds define el comportamiento del fluido.

$$Re = \frac{UL}{\mu} \quad (1)$$

Fórmula 1. Número de Reynolds

En el caso de un cilindro circular se puede entonces observar las siguientes situaciones:

- Cuando $r < 0,5$, se tiene un movimiento muy lento (escurrimiento). En este caso, las fuerzas de inercia son despreciables, la viscosidad actúa en todo el fluido, no hay estela aguas abajo del cuerpo y la resistencia es proporcional a la velocidad de acercamiento U , con dos tercios debidos a la resistencia de superficie.
- Cuando $0,5 < Re < 200$, los efectos de la viscosidad se concentran cerca de la pared, formándose una capa límite laminar que permanece pegada a la pared, con una separación muy atrás y una estela estrecha. La resistencia de forma, debido a la distribución asimétrica de las presiones, y es ahí donde cobra importancia.
- Cuando $200 < Re < 2 \times 10^5$, la estela se agranda debido a que la capa límite se separa cada vez más agua arriba del cuerpo, formando vórtices alternantes de Von Karman. La resistencia de forma representa alrededor del 75 % de la fuerza de arrastre total. El valor mínimo de C_D (0.95) se alcanza para Re alrededor de 2000, y después sube ligeramente debido a una mayor turbulencia en la estela. Para valores de $Re > 2000$, el punto de separación sigue avanzando más agua arriba del obstáculo, produciendo una mayor asimetría en el bulbo de presiones, lo cual puede llegar a causar el 95 % del arrastre total. [3]

Aunque el fenómeno de un fluido pasando alrededor de un obstáculo se ha estudiado desde hace mucho tiempo, no fue hasta 1912 cuando el investigador húngaro Teodoro Von Karman estableció las bases para su análisis. Von Karman explicó que este fenómeno ocurre debido a la formación de vórtices al pasar un fluido por un obstáculo, los cuales se mueven en un patrón ordenado, similar a las huellas dejadas al caminar, de ahí el nombre “calle de vórtices de Karman”.

Este fenómeno es común en la mecánica de fluidos, especialmente cuando los números de Reynolds (que miden la relación entre la inercia y la viscosidad de un fluido) son mayores a 34 y hasta alrededor de 10 millones. Algunos puntos importantes son:

- **Vibraciones inducidas por el flujo:** Cuando la frecuencia de generación de vórtices se acerca a la frecuencia natural de un objeto, se pueden generar vibraciones laterales significativas. Esto sucede, por ejemplo, cuando el viento fluye alrededor de estructuras como cables de puentes colgantes.
- **Aplicaciones naturales:** Este fenómeno se observa en la naturaleza, como en los vientos alrededor de picos montañosos, lo que ayuda a predecir ciertos comportamientos meteorológicos.
- **Medidores de flujo:** Desde la década de 1970, se desarrollaron medidores de flujo basados en este fenómeno, que aún son objeto de estudio.

III-D. Número adimensional de Strouhal

Durante las décadas de 1950 y 1960, las investigaciones se centraron en la relación entre la generación de vórtices de Karman y las oscilaciones inducidas por el flujo. Se encontró que el número adimensional de Strouhal es crucial para modelar este fenómeno, y se determina mediante la siguiente ecuación: [4]

$$S = \frac{F_v h}{V} \quad (2)$$

Fórmula 2. número adimensional de Strouhal.
Donde:

- F_v : Frecuencia de generación de vórtices.
- h : Longitud característica de la geometría del obstáculo.
- V : Velocidad media del flujo.

III-E. Otros Conceptos Matemáticos Básicos

Para entender cómo se modelan los problemas de dinámica de vorticidad en mecánica de fluidos, a continuación se describirán de manera rápida y sencilla las ecuaciones fundamentales del fenómeno.

Primero, recordemos que la ecuación de Navier-Stokes en dos dimensiones es:

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} \right) = -\nabla p + \mu \nabla^2 \mathbf{v} + \mathbf{f} \quad (3)$$

Fórmula 3. Ecuación de Navier-Stokes en dos dimensiones

Tomando el rotacional de la ecuación 2, y considerando que las fuerzas de volumen pueden ser expresadas como el gradiente de un campo escalar conservativo, se obtiene la siguiente ecuación para flujos bidimensionales: [4]

$$\frac{\partial \omega}{\partial t} + (\mathbf{v} \cdot \nabla) \omega = \frac{\nabla^2 \omega}{Re} \quad (4)$$

Fórmula 4.

Donde:

- \mathbf{v} : es el campo de velocidad del fluido.
- ω : es el rotacional del campo de velocidad.
- Re : es el número de Reynolds.

III-F. Métodos usados en OPENFOAM para la solución de las ecuaciones diferenciales:

Método Euler: El método de Euler se usa para resolver ecuaciones diferenciales ordinarias (EDO) de la forma:

$$y' = f(x, y)$$

donde y' es la derivada de y con respecto a x , y $f(x, y)$ es una función dada.

El objetivo es aproximar la solución de la EDO en el intervalo $[x_0, x_n]$ con una condición inicial $y(x_0) = y_0$.

■ **método de Gauss-Seidel:** El método de Gauss-Seidel es una técnica iterativa utilizada para resolver sistemas de ecuaciones lineales de la forma $A\mathbf{x} = \mathbf{b}$, donde A es una matriz de coeficientes, \mathbf{x} es el vector de incógnitas y \mathbf{b} es el vector de términos independientes. El método de Gauss-Seidel es una mejora del método de Jacobi y suele converger más rápidamente bajo ciertas condiciones. También es importante destacar que este fue el método escogido para la solución.

■ **GAMG (Geometric-Algebraic MultiGrid)**

GAMG (Geometric-Algebraic MultiGrid) es un método de resolución de ecuaciones utilizado principalmente para resolver grandes sistemas de ecuaciones lineales dispersos que aparecen en problemas de dinámica de fluidos computacional (CFD) y otros problemas de física computacional. Combina técnicas geométricas y algebraicas de multigrid para mejorar la eficiencia y la velocidad de convergencia al resolver estos sistemas.

Características Principales de GAMG

- **Multigrid Geométrico y Algebraico:**

- **Multigrid Geométrico:** Utiliza información geométrica del dominio del problema para construir jerarquías de mallas de diferentes resoluciones.
- **Multigrid Algebraico:** No depende de la geometría del problema y construye jerarquías de mallas basándose únicamente en la estructura de la matriz del sistema de ecuaciones.
- **Eficiencia Computacional:** GAMG es especialmente eficiente para resolver sistemas de ecuaciones muy grandes, donde los métodos directos serían computacionalmente prohibitivos.

IV. METODOLOGÍA

IV-A. Preprocesamiento

IV-A1. Discretización del espacio: Para discretizar el espacio se configuro una malla en OpenFOAM compuesta por tres regiones del dominio del problema. Una región antes del obstáculo, donde la malla es uniforme y formada por cuadros. Otra región formada por el obstáculo, donde se genera una discretización en triángulos. finalmente, una región después de que el fluido pasa el obstáculo igualmente uniforme como la primera región. Además, se configuro también en OpenFOAM los demás parámetros físicos para que se pudiera realizar la simulación. Es importante destacar que la estructura de la malla fue creada tomando como referencia el vídeo de la científica Asmaa Hadane.[5]

IV-A2. Parámetros físicos del sistema: Estos parámetros encuentran en las carpetas de constant y 0

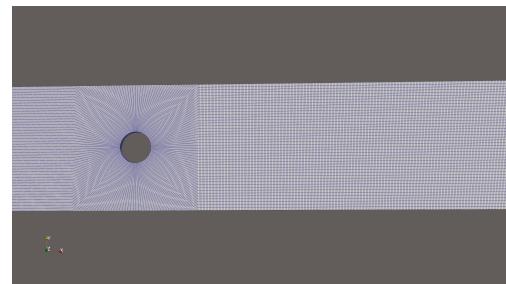


Figura 2: Estructura de la malla

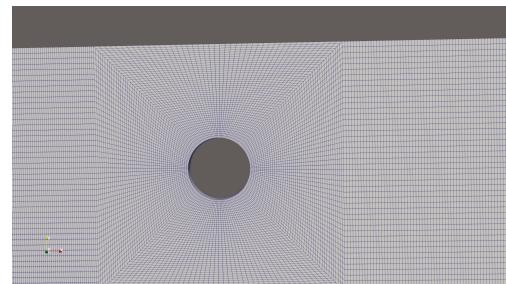


Figura 3: Malla de la simulación (foto ampliada)

Parámetros	Valor
Diámetro Cilindro	0.05 m
Fluido	Agua
Viscosidad	$1 \times 10^{-6} \text{ m}^2/\text{s}$
Número De Re	400
Velocidad entrada	0.008 m/s

Cuadro I: Parámetros físicos que describen el fluido

IV-A3. Configuración de la simulación con el script runPre.Sh: Para ejecutar todas las configuraciones necesarias para desarrollar la simulación se creo el siguiente script de Bash:

```

1 #!/bin/bash
2
3 source /usr/lib/openfoam/openfoam2312/etc/bashrc
4
5 nproc=$1
6
7 cd parallel_simulation
8
9 blockMesh &> /dev/null
10
11 sed "s/nOS/$nproc/"
12     system/decomposeParDict-draft >
13     system/decomposeParDict
14
15 decomposePar &> /dev/null
16
17 sed "s/nOS/$nproc/" machines-draft > machines

```

Listing 1: Script runPre.sh

Este script automatiza varias tareas necesarias para preparar la simulación. Primero, carga el entorno de OpenFOAM con el comando `source /usr/lib/openfoam/open-`

foam2312/etc/bashrc. Luego, navega a la carpeta de parallel_simulation donde se encuentran los archivos de la simulación.

A continuación, ejecuta blockMesh, una utilidad de OpenFOAM que genera la malla basada en el archivo blockMeshDict. Para adaptar la simulación al número de procesadores disponibles, el script usa el comando sed para modificar el archivo decomposeParDict-draft y crear decomposeParDict, que especifica cómo descomponer la malla. Luego, decomposePar se ejecuta para dividir la malla en varias partes para su ejecución en paralelo. Finalmente, el script ajusta el archivo machines para especificar los nodos en los que se ejecutará la simulación paralela.

IV-B. Solución

IV-B1. Ejecución de los datos necesarios para la simulación con el script runAll.sh: El script 'runAll.sh' automatiza la ejecución de simulaciones en paralelo. Primero, carga las configuraciones necesarias de OpenFOAM y cambia al directorio de simulación. Luego, extrae el número de procesadores a partir del archivo 'machines'. A continuación, ejecuta la simulación paralela con 'mpirun' y 'pimpleFoam', midiendo el tiempo real de ejecución y almacenándolo en el archivo de tiempo, el cual se mueve al directorio de escalado para análisis adicionales. Finalmente, reintegra la malla participada en una sola malla utilizando 'reconstructPar', completando así el proceso de simulación. Además registra en diferentes carpetas los datos obtenidos para visualizar la simulación en cada step de tiempo configurado previamente el archivo 'controlDict'.

```

1 #!/bin/bash
2
3 source /usr/lib/openfoam/openfoam2312/etc/bashrc
4
5 cd parallel_simulation
6
7 # Obtener el valor de nOS de la primera linea
    # del archivo machines
8
9 nOS=$(awk 'NR==1 {split($2, a, "="); print
    a[2]}' machines)
10 time_file="time_${nOS}.log"
11
12 echo -n "$nOS " > $time_file
13 # Ejecutar el comando mpirun con el valor de nOS
14 { time mpirun --hostfile machines -np "$nOS"
    pimpleFoam -parallel > log.FOAMout 2>
    log.FOAMerror ; } 2>&1 | grep real | awk
    '{print $2}' | awk -Fm '{print $1*60 + $2}'
    >> $time_file
15
16 mv $time_file ..../scaling/$time_file
17
18 # Reintegrar la malla
19 reconstructPar &> /dev/null

```

Listing 2: Script runAll.sh

IV-C. Postprocesamiento

IV-C1. Visualización de la simulación en ParaView: En el proceso de simulación con OpenFOAM, se generan múltiples directorios numerados que corresponden a los pasos de tiempo especificados en el archivo controlDict. Cada uno de estos directorios contiene los resultados de la simulación para un instante específico en el tiempo, lo que permite un análisis detallado de la evolución temporal de la simulación.

En cada directorio incluye archivos de datos que describen el estado del flujo en ese instante particular, como las condiciones de presión, velocidad y otros parámetros relevantes. Lo que permite que posteriormente se analice como la simulación cambia con el tiempo.

Una vez que se han generado los directorios de tiempo y se han completado las simulaciones, el archivo case.foam se puede utilizar para visualizar los resultados en ParaView. Este archivo case.foam actúa como un enlace entre los datos de simulación y el software de visualización, permitiendo abrir y visualizar los resultados de manera interactiva en ParaView.

IV-C2. Uso del script de Bash runClean.sh para liberar memoria: Este script encarga de limpiar los archivos generados durante las simulaciones de OpenFOAM. A continuación se adjunta en el informe:

```

1 #!/bin/bash
2
3 source /usr/lib/openfoam/openfoam2312/etc/bashrc
4
5 cd parallel_simulation
6
7 # Leer los valores de tmin, tmax, deltaT y
    # writeInterval desde el archivo controlDict
    # de OpenFOAM utilizando awk
8 startTime=$(awk 'NR==21 {print $2}'
    system/controlDict | tr -d ';')
9 endTime=$(awk 'NR==25 {print $2}'
    system/controlDict | tr -d ';')
10 deltaT=$(awk 'NR==27 {print $2}'
    system/controlDict | tr -d ';')
11 writeInterval=$(awk 'NR==31 {print $2}'
    system/controlDict | tr -d ';')
12
13 # Calcular tstep como el producto de deltaT y
    # writeInterval
14 stepTime=$((deltaT * writeInterval))
15
16 # Eliminar los directorios numerados segun los
    # valores leidos
17
18 rm -fr $(seq $stepTime $stepTime $endTime)
19 rm -fr constant/polyMesh
20 rm -fr processor*
21 rm -f system/decomposeParDict

```

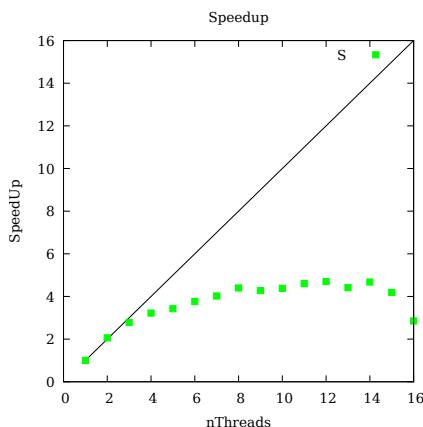
```
22 rm -f log.FOAMout log.FOAMerror
23 rm -f machines
```

Listing 3: Script runClean.sh

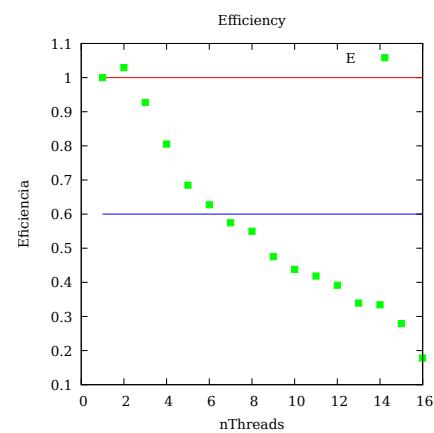
Este script es importante porque asegura que el entorno este listo para futuras simulaciones, cargando las configuraciones necesarias de OpenFOAM y cambia al directorio de simulación. Luego, utiliza awk para leer los valores de startTime, endTime, deltaT y writeInterval del archivo controlDict. Calcula el tiempo de paso (stepTime) como el producto de deltaT y writeInterval. A continuación, elimina los directorios numerados generados durante la simulación, así como otros archivos y directorios específicos utilizados por OpenFOAM, incluyendo constant/polyMesh, processor*, system/decomposeParDict, log.FOAMout, log.FOAMerror y machines.

V. RESULTADOS

V-A. Resultados de la paralelización:

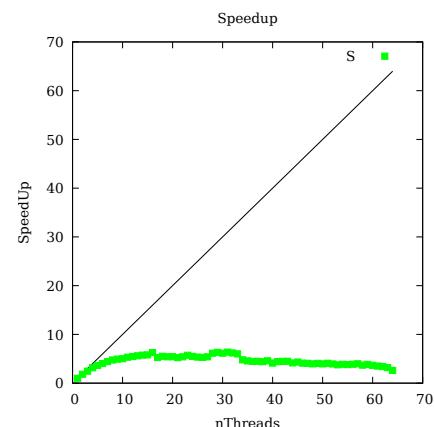
**Figura 4:** Velocidad de ejecución de la simulación para nThreads

Se observa que el aumento en el número de Threads (hilos) resulta en un crecimiento lineal hasta llegar a tres hilos. Posteriormente, el incremento se ralentiza y eventualmente se estanca. Finalmente, cuando se utilizan 16 hilos, la velocidad disminuye.

**Figura 5:** Eficiencia en relación al número de Threads

Se puede observar que la eficiencia es inversamente proporcional al aumento del número de Threads (hilos). Considerando que con 1 hilo se alcanza la eficiencia máxima posible (1), con 16 hilos la eficiencia disminuye a 0.2. Un valor óptimo para garantizar eficiencia y aprovechar el número máximo de hilos es 6, con una eficiencia del 60%.

V-A1. Resultados obtenidos con 64 Threads disponibles para la paralelización: Los resultados obtenidos en el rendimiento y velocidad de la paralelización para la simulación se muestran en las siguientes gráficas.

**Figura 6:** SpeedUp y eficiencia con 64 Threads

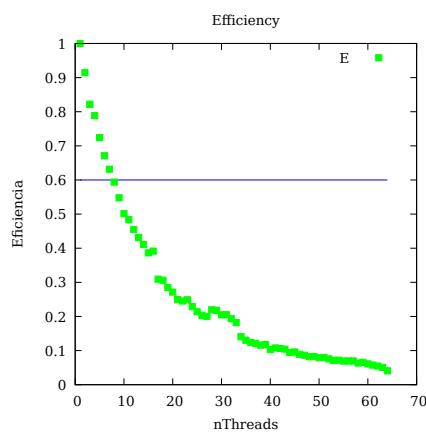


Figura 7: eficiencia de la paralelización hasta 64 Threads

Vemos en la grafica 6 que la velocidad de procesamiento de la simulacion no crece con el numero de Threads, sino que decrece despues de un numero optimo de Threads.Mientras, que en la segunda grafica 7 vemos como es el redimiento hasta 64 Threads, donde una eficiencia aceptable es de minimo 60%, esta aproximadamente en 10 Threads.

V-B. Vortices de Von Karman:

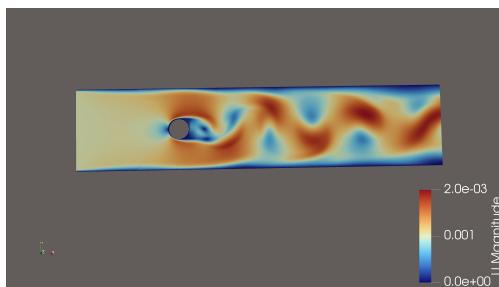


Figura 8: Simulación: Visualización de la velocidad

En la simulación cuando se aprecian regiones de diferentes velocidades, la zonas de baja velocidad (azul) y las regiones de alta velocidad (rojo), por lo tanto, en la regiones azul el flujo se retrocede, generando así regiones de baja velocidad con respecto al flujo.

En la figura 9 se muestra la distribución de presión resultante. Se usa la misma alternancia de colores (rojo y azul) como en la figura 8 lo cual nos indica que detrás del cilindro se presenta la formación y desprendimiento de estos vórtices, con áreas de alta presión y baja presión que se generan debido a la interacción del fluido con el cilindro.

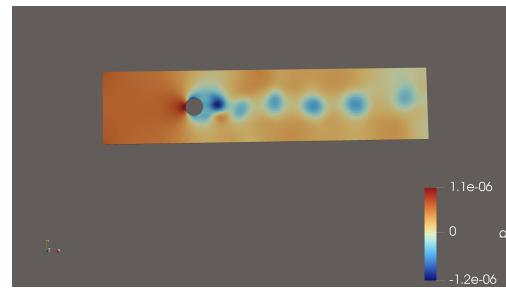


Figura 9: Simulación: Visualización de la presión

VI. ANÁLISIS DE RESULTADOS

1. Como se observa en la figura 4, el escalamiento del problema para tres hilos muestra un crecimiento lineal, pero, después de ese numero de Threads la paralelización deja de tener un crecimiento notable. Las posibles causas son la escalabilidad del problema, compartir recursos físicos del computador como la memoria cache o la contención de recursos donde muchos hilos acceden simultáneamente a los mismos recursos (memoria, discos, etc.), pueden ocurrir conflictos y contenciones, ralentizando el rendimiento. [6]
2. en la figura 5 se observa que a medida que se aumenta el número de Threads la eficiencia para el uso de los recursos disponibles del computador se hace menor, esto de acuerdo a la ley de Amdahl, la fracción de la tarea que debe ser ejecutada secuencialmente limitará la aceleración y, por ende, la eficiencia que se puede obtener a medida que se incrementa el número de procesadores.
3. Como se observa en las figuras 8 y 9, cuando el fluido se encuentra con el cilindro, la velocidad del flujo disminuye y la presión aumenta en la región frontal. Además que a medida que el fluido fluye alrededor del cilindro, se separa en dos flujos en la parte posterior, creando zonas alternadas de alta y baja presión. También se destaca que en las áreas de baja presión, la velocidad del flujo aumenta, mientras que en las áreas de alta presión, la velocidad disminuye. Lo cual permite que este fenómeno genere la formación de los vórtices de Von Kármán.

4. La visualización de los vórtices depende del numero de Raynolds y el coeficiente de viscosidad del fluido, donde se define un rango óptimo de $100 < Re < 2000$. Densidad y Viscosidad: La densidad (ρ) y la viscosidad (μ) del agua son $\rho \approx 998 \text{ kg/m}^3$ y $\mu \approx 0,001 \text{ Pa} \cdot \text{s}$. La limitación esta también determinada por el algoritmo de solución y visualización.
5. Comparando los resultados al paralelizar la simulación con 16 Threads y 64 Threads, vemos que las graficas muestran el mismo comportamiento, por lo tanto, se puede inferir que si se sigue aumentando el número de threads se seguirá con este comportamiento.

VII. DIFICULTADES

- Una de las principales dificultades fue la construcción del entorno de simulación en Docker. La creación de un Dockerfile que incluya todas las dependencias y herramientas necesarias para ejecutar la simulación en OpenFOAM, ya que fue necesario garantizar que todas las bibliotecas y paquetes requeridos por OpenFOAM estuvieran correctamente instalados y configurados dentro del contenedor. Esto incluyó la instalación de herramientas esenciales como Bash, así como la configuración de variables de entorno específicas que OpenFOAM necesita para funcionar correctamente.
- Otra dificultad fue el proceso de exportar los datos de la solución de OpenFOAM desde el contenedor Docker en la sala de cómputo, al computador personal, para su visualización en ParaView. Al principio, no se tenía un claro conocimiento de cómo realizar esta transferencia. Fue necesario investigar y aprender los métodos adecuados para copiar los archivos generados dentro del contenedor Docker y transferirlos a un entorno local donde ParaView pudiera acceder a ellos. Este proceso involucró el uso de comandos de Docker para copiar archivos.
- Durante la paralelización del código, se observó un estancamiento en la velocidad de ejecución. Aunque se consideraron varias posibles causas para este problema,

no se pudo identificar con precisión la razón detrás de la dificultad de escalamiento. Como resultado, ninguna de las posibles causas pudo ser descartada definitivamente.

- Adicionalmente, la creación de la malla fue un desafío considerable. Inicialmente, se planificó usar Gmsh para generar la malla, pero surgieron muchos problemas que complicaron su uso. Estos problemas llevaron a la decisión de optar por una malla en tres regiones. La configuración y el diseño de esta malla alternativa tomaron como referencia el video de la científica Asmaa Hadane [5]

VIII. CONCLUSIONES

Este proyecto logró exitosamente la simulación de los vórtices de Von Karman, en donde OpenFOAM fue crucial para este logro, ya que permitió simular con precisión el comportamiento de los vórtices mediante la correcta definición de la geometría, la malla, y la implementación de los parámetros físicos necesarios para recrear el fenómeno.

Además, se consiguió paralelizar la simulación, lo cual fue fundamental para aumentar la eficiencia en el cálculo de datos para la solución del problema, resultando en un incremento significativo de la velocidad de la simulación.

Por último, se concluyó que ParaView, al permitir visualizar los resultados de manera interactiva, facilita la comprensión precisa y detallada de los patrones de flujo y la dinámica de los vórtices de Von Karman, entendiendo de una forma mucho más completa de cómo surge el fenómeno físico.

REFERENCIAS

- [1] OpenFOAM, *OpenFOAM - The Open Source CFD Toolbox*, <https://www.openfoam.com>, Accessed: 2024-06-21.
- [2] ParaView, *ParaView - Open Source Scientific Visualization*, <https://www.paraview.org>, Accessed: 2024-06-21.
- [3] R. Roca Vila, *Introducción a la mecánica de los fluidos*, 2.^a ed. Mexico DC, Mexico: Editorial LIMUSA, 1993.

- [4] O. D. López, «Modelado computacional de la calle de vértices de Karman por dinámica de vorticidad,» *Mecánica Computacional*, vol. 274, págs. 274-292, 2002.
- [5] A. Hadane. «How to simulate a Flow around Cylinder - Von Karman in #OpenFoam - #ParaView #AsmaaHadane.» Consultado el 20 de junio de 2024. (2020), dirección: <https://www.youtube.com/watch?v=YfK53PKpdgE>.
- [6] J. L. Aguilar Castro y E. Leiss, *Introducción a la Computación Paralela*, 1.^a ed. Mérida, Venezuela: CEMISID, Departamento de Computación, Universidad de Los Andes, 2004, Depósito Legal: LF395920040041698, ISBN: 980-12-0752-3.

APÉNDICE

Visualización de la velocidad: <https://youtu.be/eL0JGFMLmnc>

Visualización de la presión: <https://youtu.be/gdy-RjusXG0>

Sustentación del Proyecto: <https://youtu.be/-qipzMt7ynk>