



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

SIMULACIÓN MEDIANTE DINÁMICA DE FLUIDOS COMPUTACIONAL
(CFD) EN EL TANQUE FLOCULADOR DE LA PTAR PILOTO DE
UDLAPARK

AUTOR

Diego Felipe Monroy Román

AÑO

2020



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

SIMULACIÓN MEDIANTE DINÁMICA DE FLUIDOS COMPUTACIONAL (CFD) EN
EL TANQUE FLOCULADOR DE LA PTAR PILOTO DE UDLAPARK

Trabajo de Titulación presentado en conformidad con los requisitos establecidos
para optar por el título de Ingeniero Ambiental en Prevención y Remediación.

Profesor guía

M.Sc. Santiago Daniel Piedra Burgos

Autor

Diego Felipe Monroy Román

Año

2020

DECLARACIÓN DEL PROFESOR GUÍA

"Declaro haber dirigido el trabajo, Simulación mediante dinámica de fluidos computacional (CFD) en el tanque floculador de la pta piloto de udlapark, a través de reuniones periódicas con el estudiante Diego Felipe Monroy Román, en el semestre 2020-1, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación".



Santiago Daniel Piedra Burgos

Master of Science in Hydro Science and Engineering

C.I. 1715384150

DECLARACIÓN DEL PROFESOR CORRECTOR

"Declaro haber revisado este trabajo, Simulación mediante dinámica de fluidos computacional (CFD) en el tanque floculador de la pta piloto de udlapark, de Diego Felipe Monroy Román, en el semestre 2020-1, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación".

A handwritten signature in blue ink, appearing to read "Daniel Hernán Hidalgo Villalba", is written over several horizontal lines. The signature is stylized and somewhat obscured by the lines.

Daniel Hernán Hidalgo Villalba

Magister en Ciencias Especialización Ingeniería Ambiental

C.I. 1801914449

DECLARACIÓN DE AUTORIA DEL ESTUDIANTE

"Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes."



Diego Felipe Monroy Román

C.I. 1716429947

RESUMEN

El tratamiento de aguas residuales debería ser aplicado en todo proceso que involucre alguna afectación o cambio en las características del agua. El proceso físico químico es uno de los más importantes debido a que son aquellos que mayor remoción de sólidos tiene. Hoy en día estudiar la dinámica de fluidos computacionales permite simular virtualmente la hidrodinámica de dichos procesos y entender el comportamiento de los fluidos dentro de los tanques.

En este estudio, se realiza una simulación del comportamiento de un tanque floculador de la PTAR piloto de UDLAPARK, mediante el solver pimpleFoam en el software OpenFoam, mediante un modelo de turbulencia de tipo RANS, que muestra el comportamiento y tiempo del agua dentro del tanque.

Desde el software Salome se creó la geometría utilizada para el proyecto, un tanque floculador (1.16x0.625x0.7) m con dos diferentes cámaras, las cuales se están divididas por una lámina de acrílico. El tanque tiene una velocidad de entrada y de salida del agua de 0.33m/s, por un tubo de 2 pulgadas.

En la última etapa, se visualizaron los resultados obtenidos en el programa ParaView, en el cual, tras de realizar una simulación de 150 segundos con un delta T de 0.001 segundos, se obtuvieron diferentes figuras y gráficas, en donde se observó el comportamiento del agua dentro del tanque y finalmente, se calculó el tiempo de retención hidráulico en el tanque en sus dos dimensiones.

ABSTRACT

The treatment of residual waters should be applied in any process that involves any affectation or change in the characteristics of the water. The physical chemical process is one of the most important because they are those that have the highest solids removal. Nowadays studying the dynamics of computational fluids allows us to virtually simulate the hydrodynamics of these processes and understand the behavior of fluids inside tanks.

In this study, a simulation of the behavior of a flocculator tank of the UDLAPARK pilot WWTP is performed, using the pimpleFoam solver in the OpenFoam software, using a RANS type turbulence model, which shows the behavior and time of the water inside the tank.

The geometry used for the project was created from the Salome software, a flocculator tank (1.16x0.625x0.7) m with two different chambers, which are divided by an acrylic sheet. The tank has a water inlet and outlet speed of 0.33m / s, through a 2-inch tube.

In the last stage, the results obtained in the ParaView program were visualized, in which, after simulating 150 seconds with a delta T of 0.001 seconds, different figures and graphs were obtained, where the behavior of the water was observed inside the tank and finally, the hydraulic retention time in the tank was calculated in its two dimensions.

ÍNDICE

1. INTRODUCCIÓN	1
1.1. Antecedentes.....	1
1.2. Objetivos.....	3
1.2.1. Objetivo general.....	3
1.2.2. Objetivos específicos	3
1.3. Alcance.....	3
1.4. Justificación	4
2. MARCO TEÓRICO.....	5
2.1. Aguas Residuales	5
2.2. Clasificación de Aguas Residuales	5
2.3. Tratamiento de Aguas Residuales	6
2.4. Floculación	7
2.5 Computational Fluid Dynamics	8
2.6 Salome	13
2.7. Propeller	13
2.8. OpenFOAM.....	14
2.8.1 Carpetas	15
2.8.2 Comandos estándar	16
2.8.3 Directorios necesarios	17
2.8.4 Algoritmo pimpleFoam.....	20

2.8.5	Turbulencia RANS.....	21
2.9.	ParaView	22
3.	METODOLOGÍA.....	23
3.1.	Preprocesamiento	23
3.1.1.	Geometría del sistema.....	23
3.1.2.	Condiciones iniciales del sistema	29
3.1.3.	Carpetas	29
3.2.	Procesamiento.....	31
3.3.	Postprocesamiento	33
4.	RESULTADOS Y ANÁLISIS	34
4.1.	Estabilización de Residuales	34
4.2.	ParaView	37
4.3.	Análisis de velocidades.....	39
4.4.	Tiempo de Retención Hidráulico	42
5.	CONCLUSIONES Y RECOMENDACIONES	45
5.1	Conclusiones	45
5.2	Recomendaciones	46
	REFERENCIAS.....	48

1. INTRODUCCIÓN

1.1. Antecedentes

El agua es uno de los fluidos más importantes para la existencia de la Tierra y para la vida de todos los organismos que habitan en ella. Como es de conocimiento el agua es un recurso natural renovable, pero de gran agotamiento y es por ello, que es imprescindible darle una utilización adecuada, ya que el 70% del planeta está cubierto por agua. Sin embargo, de este porcentaje solo un 3.5% es agua dulce disponible para el consumo humano, del cual el 70% está en glaciares y casquetes polares, un 30% en aguas subterráneas y pozos, y menos del 1% está disponible en cuencas hidrográficas y cuerpos hídricos (AQUAE, 2018).

A partir del siglo XX, el tratamiento que recibe el agua se ha ido sistematizando con el paso del tiempo a través de nuevas tecnologías de saneamiento que posibilitan reutilizar el agua (López Vásquez, Buitrón Méndez, García, & Cervantes Carrillo, 2017). Actualmente, la contaminación que se produce diariamente los cuerpos hídricos por el ser humano ha sido la causante de millones de enfermedades a nivel mundial, entre las más comunes, enfermedades gastrointestinales que son causadas por agua contaminada con diferentes tipos de bacterias, virus o parásitos, como por ejemplo bacterias (principalmente *Escherichia coli*, *Salmonella* y *Shigella*), parásitos (*Giardia lamblia* y amibas), y virus (Rotavirus y virus Norwalk), los cuales producen un desorden en el sistema gastrointestinal (Aguilera & Castro-escarpulli, 2011).

Otro factor es la presencia potencial de contaminantes químicos, orgánicos e inorgánicos y metales pesados, procedentes de fuentes industriales, agrícolas y de la escorrentía urbana, los organofosforados y los carbamatos que, al tomar contacto con el ser humano, probablemente, son la causa de enfermedades crónicas (cáncer, daños en órganos internos), alteraciones en el sistema nervioso e inmunológico, lesiones al genoma humano y producto de la reproducción (Unemi, 2016).

A nivel mundial el 90% de los contaminantes llegan al océano, debido a diferentes actividades antrópicas, especialmente en actividades industriales, y que los mismos arrojan estos contaminantes directamente al río, y que por la cadena de arrastre desde los ríos pequeños contaminados por las diferentes industrias que no manejan las medidas o normas adecuadas para evitar la contaminación del agua, estos contaminantes llegan hasta el océano (López Vásquez, Buitrón Méndez, García, & Cervantes Carrillo, 2017).

Así mismo, cabe mencionar que la cantidad de aguas residuales que se generan cada día, depende de cada ser humano y del proceso industrial para el cual se ocupe el agua (UNESCO, 2015). Existe una normativa ambiental, la cual regula la concentración máxima permitida de descarga a diferentes cuerpos hídricos que una industria de cualquier tipo puede realizar (Lopez, 2001).

Para evitar el daño a la salud de los seres humanos y evitar la contaminación del ecosistema, se ha propuesto el uso de herramientas en calidad de agua que consisten en analizar los contenidos, físico químico y biológicos. (Córdova, Gaete, & Aránguiz, 2009) El ser humano necesita consumir de un mínimo de tres litros diarios de agua potable y un total de treinta litros de agua potable diarios, para las actividades antropogénicas, entre el lavado de ropa, el cuidado personal, y actividades domésticas como cocinar, regar el jardín, entre otras.

El Ecuador presenta la gran ventaja de disponer agua dulce 22 500 m³/hab/año, lo cual es superior a 1000 m³/hab/año considerados según datos de la Organización Mundial de la Salud (OMS) (Unemi, 2016). Bajo este escenario la Constitución de la República de Ecuador 2008 y el Plan Toda una Vida 2017-2021, garantizan el uso y aprovechamiento para los seres vivos (salud y vida), así como el cuidado de este recurso, por lo cual es importante considerar el buen uso y manejo del recurso.

Por otra parte, es importante considerar que, debido al crecimiento poblacional, la industrialización y el alto consumo del ser humano han generado modificaciones en las características de los recursos hídricos, causando un incremento en los niveles

de contaminación provocando que el agua no sea apta para el uso y el consumo del ser humano. En este sentido, los procesos de tratamiento del agua son cada vez más complejos y diversos, por lo cual, dichos procesos permiten liberar el agua de microorganismos patógenos, sustancias tóxicas o nocivas para la salud, y de esa manera cumplir con las normas bacteriológicas y fisicoquímicas establecidas por la Constitución (Cristina & Camacho, 2011).

El tratamiento del agua abarca una evaluación continua de su calidad, la cual contiene las siguientes fases: fuente, tratamiento, almacenamiento y distribución a la población; tomando en cuenta que el tratamiento del agua residuales se debe gestionar adecuadamente para su posible reutilización o reciclaje (Cristina & Camacho, 2011).

1.2. Objetivos

1.2.1. Objetivo general

- (i.) Simular el funcionamiento hidráulico del floculador de la planta de tratamiento de aguas residuales piloto de UDLAPARK.

1.2.2. Objetivos específicos

- (i.) Establecer las condiciones de borde del floculador de la PTAR piloto de UDLAPARK
- (ii.) Estimar el tiempo de residencia real del floculador de la PTAR piloto de UDLAPARK

1.3. Alcance

El estudio tiene por objeto realizar un modelo computacional del comportamiento del agua en el floculador de la PTAR piloto de UDLAPARK, el cual se dividirá en 3 fases: preprocesamiento, procesamiento, post procesamiento. Primero se entenderá cuáles son las condiciones iniciales del sistema donde se realizará la simulación mediante diferentes softwares. Dentro de las condiciones iniciales de borde se determinarán diferentes parámetros como la velocidad y presión a tomar

en cuenta previos a la simulación en el software. Consecuentemente, se realizará la simulación CFD en el floculador, en la cual se entenderá como es el comportamiento dinámico del agua dentro de la unidad. Como última fase, se interpretarán los resultados obtenidos para cumplir con los objetivos y estimar el tiempo de retención hidráulico.

1.4. Justificación

Existen tres métodos diferentes para analizar los fluidos, los cuales son mediante la experimentación, simulación y el enfoque teórico. Los tres campos de estudios nos permiten visualizar y examinar los fluidos de manera diferente, ya que cada uno tiene una metodología que difiere de la otra (Lidia, 2004).

La experimentación utilizar una gran cantidad de recursos, gran disponibilidad de tiempo y resultados inciertos. A diferencia de la simulación, abarca bajos costos, reducción de recursos, se puede realizar las repeticiones necesarias, resultados concretos y más seguridad para quienes realizan el experimento. Por otra parte, el enfoque teórico genera ciertas aristas, ya que no permite visualizar el panorama real del problema, por lo cual, se dificulta encontrar una solución. Es por esto que realizar una simulación es mucho mas eficiente que realizar una experimentación (Lidia, 2004).

En el presente trabajo, el desarrollo fue considerado desde la dinámica de fluidos computacional (CFD) en el tanque floculador, el cual se define como una herramienta de simulación que una máquina realiza mediante matemática aplicada para modelar situaciones de diferentes flujos de fluidos para el diseño óptimo de procesos industriales, predicción de calor, masa y transferencia de impulso (Xia & Sun, 2002).

Se puede definir al CFD como un mecanismo que permite la reducción en la cantidad de recursos a ser utilizados, de manera que reduce el impacto al medio ambiente. Asimismo, a través de la simulación del CFD, se obtienen resultados más reales y concretos, que al realizar una experimentación física. Por otra parte, la

simulación permite visualizar y determinar el correcto funcionamiento de la distribución de agua dentro del tanque, lo cual, evita que se generen puntos críticos o zonas muertas dentro del tanque (Xia & Sun, 2002).

2. MARCO TEÓRICO

2.1. Aguas Residuales

El agua residual es el agua que, por uso antrópico, presentan un peligro y debe ser tratada previa a su descarga a cualquier sistema hídrico, ya que contiene grandes cantidades de sustancias y/o microorganismos (Lopez, 2001).

2.2. Clasificación de Aguas Residuales

Estas aguas residuales tienen cuatro principales orígenes, los cuales se detallan a continuación:

- (i.) Aguas residuales negras o domésticas: Son aguas que tienen procedencia urbana, estas suelen contener materia orgánica, microorganismos y tensoactivos como jabones, detergentes y grasas en pequeñas cantidades, procedentes de heces y orina, aseo personal y limpieza de los hogares.
- (ii.) Aguas blancas: Si bien es cierto que este tipo de agua de procedencias atmosférica como agua de lluvia, nieve o hielo no es considerada residual, sin embargo, en cierto momento de su ciclo se llega a contaminar por diferentes actividades antrópicas. Son aguas a las cuales se les puede dar un tratamiento diferenciado a los otros tipos de aguas residuales que tienen contaminantes más pesados.
- (iii.) Aguas residuales industriales: Procedentes de cualquier empresa o establecimiento industrial, que tenga como fin un producto, normalmente contienen altas cargas de grasas, detergentes y diferentes productos y subproductos de origen químico, mineral o vegetal.

- (iv.) Aguas residuales agrícolas: Se denomina agrícolas a toda agua de uso rural, ya que en su mayoría son para fines agrícolas, y estas suelen tener grandes cantidades de químicos y productos vegetales, como biocidas o pesticidas (Lopez, 2001).

2.3. Tratamiento de Aguas Residuales

Cabe mencionar la importancia de darle a un agua residual el correcto tratamiento previo a su descarga a cualquier cuerpo hídrico receptor, debido a la gran cantidad de sustancias tóxicas, no tóxicas y microorganismos que se encuentran en el agua, que puede ser vehículo o causa de contaminación no solo a recursos naturales sino también para los seres humanos.

La calidad del agua es el principal elemento afectado debido a diferentes actividades antrópicas, ya que la hace impropia y peligrosa para el consumo humano u otras actividades recreativas. Los principales problemas que presentan las aguas residuales son algunos como, por ejemplo: malos olores, toxicidad, adición de compuestos orgánicos o inorgánicos y afectaciones a los recursos naturales como eutrofización, contaminación de pozos subterráneos, entre otros.

En la actualidad, existen un sin número de maneras de tratar el agua, existen cuatro etapas para el tratamiento efectivo de un agua residual, las cuales son:

- (i.) Pretratamiento: En este proceso se retiran los sólidos más grandes, mediante rejillas, para evitar que estos sólidos afecten los demás tratamientos (Lozano, 2016).
- (ii.) Primario: El principio básico de este método consiste en eliminar los sólidos suspendidos del agua residual mediante procesos químicos o físicos, con ayuda de la gravedad y adición de químicos que ayuden a remover las impurezas del agua (Lozano, 2016).
- (iii.) Secundario: En este tratamiento se remueven todas las partículas o contaminantes mediante actividad de microorganismos, esto quiere decir que la materia orgánica en suspensión y/o disuelta es asimilada por una

determinada cantidad de microorganismos, los cuales purifican al agua hasta casi un 90% (Lozano, 2016).

- (iv.) Terciario: Esta es una etapa adicional no obligatoria, en la cual su objetivo es de refinamiento, en otras palabras, se la usa para casos donde el agua este con una gran carga de contaminante o posteriormente, su uso sea para consumo humano, actividades recreativas o que la empresa o ciudad que la esté descargando tenga parámetros más altos de calidad de descargas de agua (Lozano, 2016).

2.4. Floculación

Operación unitaria caracterizada por ser un proceso físico que cumple la función de realizar una separación de la materia solida del medio líquido, en donde se presenta una diferencia de densidades y masas entre el agua y los elementos en suspensión, lo que termina complementando el desempeño del reactor secundario para producir un efluente que no presente elevadas concentraciones de sólidos en suspensión (Amy et al., 2017).

La floculación es el proceso por el cual las partículas se adhieren a las superficies de otras partículas, de tal manera que se forman partículas de mayor peso y tamaño para que se sedimenten con mayor facilidad.

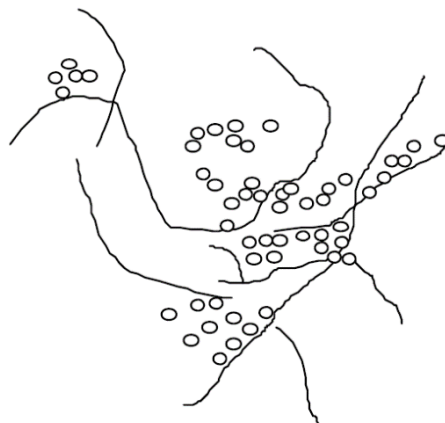


Figura 1. Proceso de floculación. Tomado de (Andía, 2000).

En el proceso de floculación intervienen tres mecanismos de transporte secuencial, los cuales son:

- (i.) Floculación pericinética, es la formación de micro flóculos que sirven como núcleos de floculación.
- (ii.) Floculación ortocinética, su objetivo es la coalición entre dichas partículas con el movimiento del agua impulsado por un movimiento hidráulico o mecánico.
- (iii.) Sedimentación diferencial, es el proceso en el cual los flóculos de menos tamaño se adhieren a los de mayor tamaño.

En otras palabras, una vez colocado el coagulador en el volumen de agua, el mismo desestabiliza las partículas que se encuentran en el agua, de manera que el procedimiento de floculación pericinética, en la cual las partículas más pequeñas empiezan a colisionar entre sí, hasta alcanzar el tamaño ideal. Cuando se alcanza el tamaño mínimo de un micrómetro, empieza la floculación ortocinética, la cual tiene mayor efectividad en temperaturas que rodean los cero grados centígrados (Andía, 2000).

Asimismo, existen parámetros por los cuales la floculación se lleva a cabo de manera adecuada, para esto se debe asegurar que:

- (i.) La gradiente de velocidad impulsada por los propellers sea la adecuada.
- (ii.) Tiempo de retención hidráulica en el tanque.
- (iii.) Densidad y tamaño de los flóculos que se formen, lo cual tiene relación directa con la cantidad de coagulante que se añadió previamente (Andía, 2000).

2.5 Computational Fluid Dynamics

Pertenece a una ramificación asociada a la mecánica de fluidos en donde se aplican análisis numéricos y algoritmos con el fin de analizar el comportamiento hidráulico. Siendo las computadoras la herramienta principal que lleva a cabo los cálculos necesarios para simular la interacción entre líquidos y gases con relación a

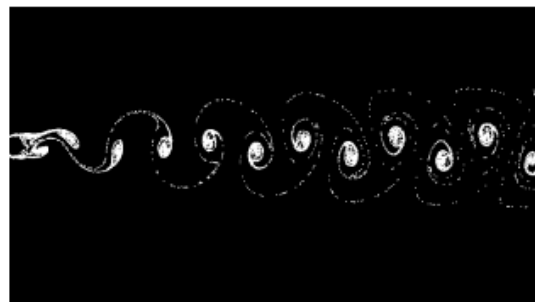
superficies definidas por condiciones de borde de. El CFD trabaja con tres ecuaciones generales de donde parte la resolución del comportamiento de los fluidos y hacen referencia a la ecuación de continuidad, de la cantidad de movimiento y de la conservación de la energía (C. J. Greenshields, 2019).

El procedimiento realiza una discretización de una zona o región en un espacio determinado, creando un mallado computacional de la geometría con la que se requiere trabajar, de tal manera que se generan ciertos volúmenes de control y se dispone a resolver cada volumen de control por medio de las ecuaciones referentes a la conservación que se discretizan de manera iterativa hasta obtener residuales cada vez más pequeños y estables (Sadrehaghighi, 2019).

El origen del CFD trasciende a la Segunda Guerra Mundial cuando los científicos del laboratorio Nacional Los Alamos desarrollaban la bomba atómica, así como también herramientas numéricas que describen el flujo violento que crearía la bomba. El método desarrollado por J. Von. Neumann fue el precursor del avance de la programación electrónica en computadoras (van Leer & Powell, 2010).



real experiment



CFD simulation

Figura 2. Explicación CFD frente a una simulación real. Tomado de (Date, 2005).

El CFD crea una realidad virtual donde los usuarios pueden representar trayectorias de fluidos. En la realidad, este tipo de procesos serian costosos, difíciles, peligrosos o hasta imposibles (van Leer & Powell, 2010).

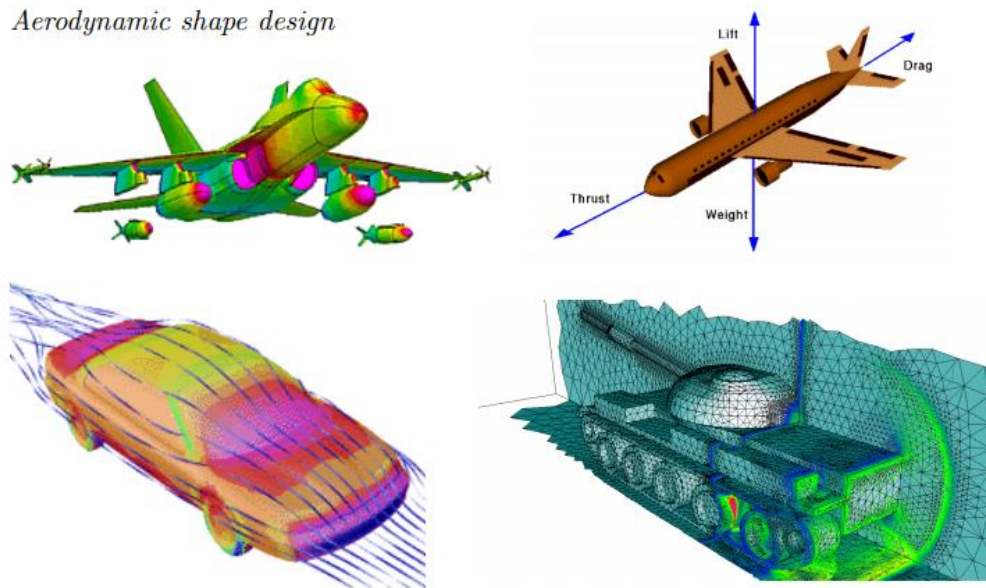


Figura 3. Ejemplos de aplicación de CFD. Tomado de (Date, 2005)

Como se muestra en la Figura 3 los CFD permiten una simulación virtual idéntica a un modelo real y especialmente para entender el completo funcionamiento de una aeronave como se indica en el ejemplo.

Las ecuaciones utilizadas para este método son discretizadas en una grilla computacional, lo que conlleva a diferencias finitas, ya sea de volumen o diferentes elementos finitos. Normalmente, este tipo de modelados se utiliza para sistemas continuos, sin embargo, hay ocasiones que no existen colisiones dentro del espacio y escala de tiempo (van Leer & Powell, 2010).

Los principios del CFD se obtienen a partir de aproximaciones numéricas de ecuaciones diferenciales parciales, los cuales están sujetos a diferentes análisis numéricos (van Leer & Powell, 2010). Estos principios son:

- (i.) Discretización.
- (ii.) Consistencia.

- (iii.) Estabilidad.
- (iv.) Convergencia.
- (v.) Monotonicidad.
- (vi.) Conservación.
- (vii.) Irreversibilidad.

Las ecuaciones de Navier Stokes son las gobernantes de los CFD, las cuales definen distintos parámetros como presión, densidad, temperatura y velocidad de un fluido. La ecuación para la continuidad de un flujo incompresible es:

Ecuación 1

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0$$

De las ecuaciones de Navier Stokes se definen 3 diferentes ecuaciones para los diferentes vectores (x,y,z), las cuales son las siguientes:

Para x:

Ecuación 2

$$\rho * \left(\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} \right) = - \frac{\partial P}{\partial x} + \rho g_x + \mu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right)$$

Para y:

Ecuación 3

$$\rho * \left(\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} \right) = - \frac{\partial P}{\partial y} + \rho g_y + \mu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right)$$

Para z:

Ecuación 4

$$\rho * \left(\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} \right) = - \frac{\partial P}{\partial z} + \rho g_z + \mu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right)$$

(Vazquez, 2004).

En donde:

u = Velocidad

u = Velocidad en x

v = Velocidad en y

w = Velocidad en z

t = Tiempo

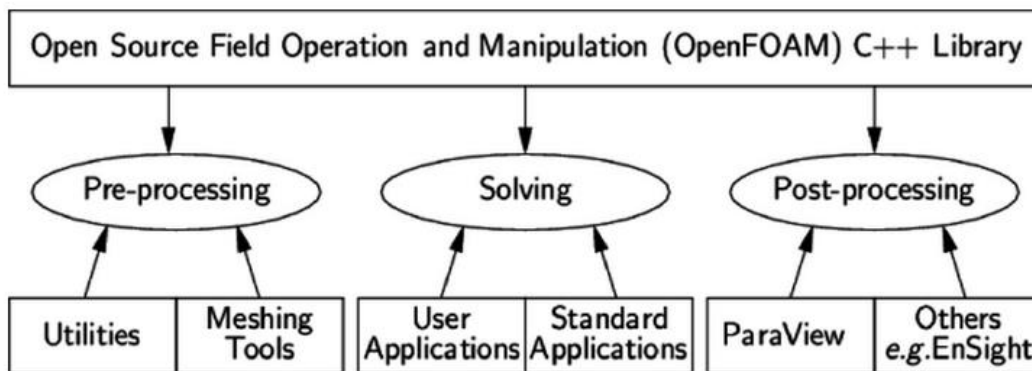


Figura 4. Configuración OpenFOAM. Tomado de (C. Greenshields, 2014)

Los algoritmos numéricos que utilizan diferentes softwares enfocados en los CFD, sirven para simular una geometría específica en función de diferentes solvers. Todos los solvers están determinados por tres elementos principales, los cuales son:

- (i.) Preprocesamiento: Esta es la primera etapa del análisis, en la cual se define la geometría del estudio y a su vez se crea la malla computacional. Dicha malla puede ser estructurada o no estructurada. Para la estructurada la discretización espacial se emplean elementos en forma de hexaedro, al contrario, la no estructurada permite una discretización de volúmenes con elementos en diferentes formas, según su adaptabilidad y forma de la geometría (Alejandro & Bolaños, 2015).

Esta fase culmina al momento de establecer las características del fluido, asignar las condiciones de borde de la simulación y determinar el tipo de análisis que puede ser estacionario o transitorio (Alejandro & Bolaños, 2015).

Para fenómenos estacionarios existen dos tipos de modelos: el de Reynolds-Averaged Navier- Stokes equations (RANS) y el Large Eddy Simulation (LES). En el presente estudio se utilizó las ecuaciones determinadas por RAS, debido a la turbulencia presentada en el sistema

- (ii.) Procesamiento: Mediante un software las ecuaciones diferenciales se transforman en sistemas de ecuaciones lineales, las cuales son resueltas con representaciones explícitas e implícitas. El software determinará según cada caso los métodos numéricos apropiados para la simulación, el cual se describe a continuación (Alejandro & Bolaños, 2015).
- (iii.) Postprocesamiento: En esta etapa final del modelamiento se analizarán los resultados mediante gráficos, diagramas y animaciones que permitirán la visualización de la dinámica y comportamiento del fluido en el tanque (Alejandro & Bolaños, 2015).

2.6 Salome

Es una herramienta de código abierto, cuyo enfoque se basa en el preprocesamiento de un CFD para realizar la simulación. Entre sus principales ventajas, permite un diseño flexible para realizar distintas figuras geométricas (Holzinger, 2020).

2.7. Propeller

Un propeller es un perfil giratorio que está formado por al menos dos bordes afilados unidos a un punto central que está montado en el cigüeñal del motor. La capacidad de la hélice es el cambio sobre la energía del motor a un empuje valioso. Los bordes de corte de la hélice tienen un borde principal, detrás borde, punta, caña, frente y espalda. Las hélices cambian sobre el movimiento giratorio de motores de cilindro,

turbopropulsores o motores eléctricos para dar la potencia propulsora (Dekanski, 1993).

Un propeller tiene dos superficies hidrodinámicas, las cuales son conocidas como la cara de succión (cara trasera) y la cara de presión (cara delantera). Para el diseño y construcción de cada una de las aspas del propeller se deben tener en cuenta una serie de parámetros que determinaran la efectividad de estos propellers independientemente de cuál sea su uso. A continuación, se detallarán los parámetros para el correcto diseño de un propeller.

Primero, se debe determinar la geometría del propeller, dependiendo de cual vaya a ser el uso del propeller, se establecen las dimensiones y forma de cada una de las aspas, ya que en estos casos es importante el lanzamiento axial y circunferencial de cada una de las hélices.

El tipo de hélice a su vez también es fundamental, ya que existen dos tipos de hélices. Hélice de paso fijo (FPP) o hélice de palas orientables (CPP). Para la mayoría de los proyectos se utilizan estas últimas debido a la compleja mecánica que presentan y sus diferentes usos. Además, que la CPP permite funcionar a una velocidad constante y el costo de estas es más económico (Dekanski, 1993).

Como tercer punto el tamaño de la hélice es otro factor significativo es la determinación del diámetro de la hélice, en la cual se debe tomar en cuenta la potencia entregada, la velocidad de la hélice y para cada uno de estos parámetros los procedimientos de cálculo (Dekanski, 1993).

Finalmente, la selección del número de hélices es muy variable, dependiendo del uso que se le vaya a dar varía entre dos y siete hélices. Este número depende de la necesidad de evitar frecuencias de resonancia perjudiciales para la estructura, así como las frecuencias vibratorias, y según menor sea el número de hélices, el rendimiento y diámetro óptimo del propeller aumenta (Figura, n.d.)

2.8. OpenFOAM

El software OpenFoam (Open Source Fields Operation and Manipulation) que se utiliza principalmente para crear ejecutables conocidos como aplicaciones. Las aplicaciones derivan en dos categorías: solvers que son diseñados para resolver un problema mecánico en específico mientras que los utilities están diseñados para generar acciones que involucran una manipulación de la información (Cardiff, 2017).

Este tipo de programa brinda una gran cantidad de aplicaciones, lo cual permite modificar conforme a las necesidades de cada uno de los usuarios. Por lo tanto, cada una de estas aplicaciones vienen incluidas carpetas de arranque, las cuales son indispensables para que la simulación empiece, sin embargo, en cada uno de los casos se pueden extraer carpetas de una aplicación y llevarlas a otra dependiendo de cada una de las necesidades y complejidades de la simulación a realizarse (Cardiff, 2017).

2.8.1 Carpetas

Una de las ventajas del OpenFoam es que guarda la información en diferentes directorios, lo cual permite mantener un orden y en caso de que exista un error se lo pueda encontrar con mayor facilidad. Existen tres tipos de subdirectorios (0, constant, system) (Cardiff, 2017).

0: Contiene las condiciones iniciales de borde de todas las cantidades variables.

Constant: Este subdirectorio contiene las propiedades físicas del fluido y los diferentes parámetros que lo definen. Además, dentro de este subdirectorio existe una carpeta llamada polymesh, la cual contiene la figura geométrica realizada por Salome (Cardiff, 2017).

System: Finalmente en este subdirectorio se describen todos los parámetros de la simulación, como por ejemplo el tiempo que durará la simulación, los esquemas de discretización y los algoritmos de control del solver (Cardiff, 2017).

Dentro de cada uno de estos subdirectorios, existen diferentes archivos, los cuales contendrán información única para que la simulación se desarrolle a la perfección.

A continuación, se irán listando todos los archivos con la descripción y la funcionalidad de cada uno de ellos (Cardiff, 2017).

2.8.2 Comandos estándar

Dentro del software OpenFoam, existen diferentes comandos, los cuales permiten el desarrollo de la simulación, los utilizados para el presente proyecto se describen a continuación:

- (i.) Touch: Este comando permite crear un archivo compatible con ParaView al comienzo para generar la información necesaria al final de la simulación.
- (ii.) ideasUnvToFoam: Es un convertidor que se utiliza para transformar de un formato UNV a formato de OpenFOAM.
- (iii.) surfaceFeatureExtract: Esta herramienta es utilizada para extraer las características desde un archivo. STL. Este comando es controlado por un archivo, el cual permite la conversión de los datos para obtener objetos.
- (iv.) blockMesh: Mediante este comando se genera el mallado que se va a utilizar durante la simulación.
- (v.) snappy Hexmesh: Después de generar el mallado, esta utilidad permite generar una superficie de mejor mallado encima de otra que es generalmente aerodinámica externa.
- (vi.) decomposePar: Posibilita correr en simultaneo con todos los procesadores que cuente la máquina en la que se realice la simulación.
- (vii.) setFields: Define regiones dentro de la geometría ya establecida y asignar valores de campo, en este caso realiza la función de que tan lleno se encuentra el tanque.
- (viii.) interFoam: Solver mediante el cual se resuelven las ecuaciones planteadas y se logra visualizar los resultados en ParaView (Greenshields, 2018).

2.8.3 Directorios necesarios

(i.) Carpeta 0.

Dentro de este directorio existen dos archivos principales, en los cuales se realizaron cambios. Para cada configuración se deben configurar los tipos de entrada según dependa cada caso los existentes son: fixedValue, zeroGradient, InletOutlet o pressureInletOutletVelocity (Greenshields, 2014).

Tabla 1.

Estructura archivo 'alphaWater'

Campo de borde	Inlet	type	fixedValue
		Value	0.5
	Outlet	type	zeroGradient
		value	0
	Atmosphere	type	InletOutlet
		inletValue	0
	Walls	type	zeroGradient

En la tabla 1 se indica la configuración realizada en el archivo alpha water, en el cual se estableció un valor de 0.5 en la entrada debido al tubo por el cual ingresa el agua.

Tabla 2.

Estructura archivo 'U'

Campo de borde	Inlet	type	fixedValue
		inletValue	uniform (0.33 0 0)
		Value	\$internalField
	Outlet	type	inletOutlet
		value	uniform (0.33 0 0)

Atmosphere	type	InletOutlet
	inletValue	0
Walls	type	noSlip

En el archivo se utilizó en este caso la configuración para velocidad como se muestra en la tabla 2 en el vértice 'x' tiene una velocidad de 0.33 m/s de entrada y de salida.

Dentro de este directorio existen otros archivos como k, nut, omega, p_rgh a los cuales no se realizaron modificaciones y permanecieron con los valores predeterminados.

(ii.) Carpeta constant

En este directorio el principal archivo que se tomó en consideración es boundary, ya que en este archivo se define la geometría, y en caso de no poseer la configuración correcta el software no permitirá que la simulación continúe.

Tabla 3.

Estructura archivo 'Boundary'

Campo de borde	Inlet	type	wall
	Outlet	type	patch
	Atmosphere	type	patch
	Walls	type	patch

Para la tabla 3 se deben especificar los distintos tipos de parches que tendrá la geometría. Los tipos de parches existentes son los siguientes: patch, wall y empty y cada uno se lo coloca dependiendo de las condiciones del sistema.

(iii.) Carpeta system

En este directorio es en el cual se realizó la mayor cantidad de modificaciones. Como primer archivo blockMeshDict, en el cual se establecieron vértices de un cubo, dentro del cual se encuentre la geometría realizada en Salome.

Tabla 4.

Archivo controlDict

Adaptado de (C. Greenshields, 2018)

application		Define el solver mediante el cual se va a realizar la simulación.
startFrom	startTime	Tiempo desde que se indique en el archivo.
stopAt	endTime	Tiempo hasta el que se corra la simulación.
deltaT		Escritura cada cuanto tiempo se indique.
adjustTimeStep	yes/no	Permite que el software determine el intervalo de escritura delta T.
MaxCo		Define el número de Courant, dependiendo de la velocidad y número de celdas.

En la tabla 4 se indican los parámetros necesarios y más importantes dentro del archivo controlDict, en donde se estableció un tiempo de simulación de 150 segundos con un delta T de 0.001, por lo que la simulación tardó alrededor de dos días en finalizar.

Existen otros archivos como el `setFieldsDict`, en el cual se estableció el nivel de agua al que llegaría dentro del tanque. Para el proyecto el valor fue: (0 0 0) (1.16 0.562 0.685) m. El último archivo fue el `snappyHexMeshDict`, en el cual se definieron el número de celdas que se crearían al momento de generar el mallado con este comando. Finalmente, el archivo `decomposeParDict`, permite señalar la cantidad de procesadores que se utilizar para la simulación en paralelo junto con cual de los ejes necesita mayor cantidad de procesadores. Otros archivos como `surfaceFeatureExtract`, `meshQuality` no se realizaron modificaciones significativas.

Finalmente, en `fvSchemes` y `fvSolutions` se realizaron cambios que permitieron obtener los residuales, los cuales se generan a partir de ejecutar el comando `interFoam`.

2.8.4 Algoritmo `pimpleFoam`

Dentro del software de OpenFOAM, no existe un solver genérico para la aplicación de todos los casos, al contrario, dependiendo de cuál sea la necesidad del usuario el software brinda una variedad de solvers en diferentes categorías para fluidos incompresibles, transferencia de energía, combustión, entre otros.

Actualmente, existen más de cien tipos de algoritmos y solvers, por lo cual es sustancial tener claro cuál será el solver para cada caso, y dada la ocasión que no exista alguno, buscar el que mejor se adapte a las necesidades de cada usuario.

El algoritmo utilizado para el proyecto se llama `pimpleFoam`, el cual es un solver dentro del algoritmo PIMPLE utilizado para fluidos incompresibles, es una combinación entre dos algoritmos, SIMPLE y PISO, aplica a cualquier tipo de turbulencia, como ya se mencionó anteriormente en este caso de tipo RANS.

Existen una serie de ecuaciones gobernantes que determinan el funcionamiento de la simulación y en general se presenta la siguiente ecuación de continuidad y momentum (Greenshields, 2014).

Ecuación de continuidad:

Ecuación 5

$$\nabla \cdot \mathbf{u} = 0$$

Asumiendo que la presión no cambia.

Ecuación de momentum:

Ecuación 6

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla(\mathbf{u}\mathbf{u}) + \nabla \cdot (\text{dev}(-\nu^{eff} (\nabla \mathbf{u} + (\nabla \mathbf{u})^T))) = -\nabla p + \mathbf{Q}$$

Donde:

u= Velocidad

t= Tiempo

p= Presión

Q= Calor

2.8.5 Turbulencia RANS

La mayoría de las simulaciones están basadas en las ecuaciones de Reynolds-Averaged Navier-Stokes, en el cual se desconocen las correlaciones de turbulencia y se expresan mediante una determinada cantidad con información empírica. Con el paso de los años, estas ecuaciones se han ido haciendo más complejas debido a los diferentes problemas que se han presentado para las diferentes simulaciones mediante estas ecuaciones RANS.

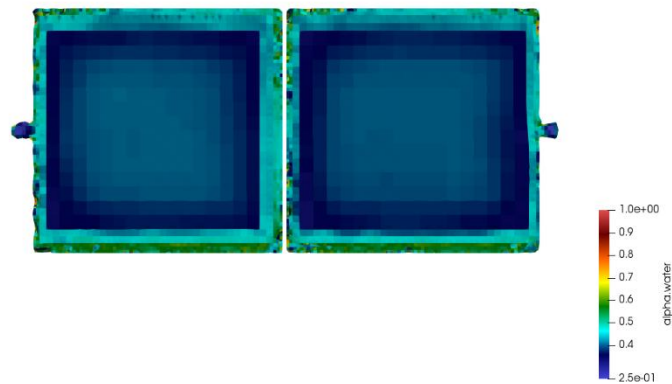


Figura 5. Distribución tipo RANS

Normalmente en los modelos de flujo turbulento, los coeficientes suelen ser constantes, debido a una serie de estudios que permitieron definir a estos valores como constantes, ya que se infirió que su variación era marginal frente a los diferentes casos de turbulencia muestreados (Rahman et al., 2016).

En este caso existe un archivo llamado `turbulenceProperties`, el cual deberá contener la información necesaria en cuanto a la simulación por desarrollarse, en este archivo se deberá especificar todos los parámetros necesarios.

Dentro de este modelo existen dos tipos para flujos incompresibles o compresibles, dependiendo de cuál sea el caso, OpenFoam tiene ya definido dentro de su infinidad de tutoriales que tipo de fluido es, posteriormente, permite hacerlo en flujo laminar o en caso de ser turbulento de tipo RANS o LES (Greenshields, 2014)

2.9. ParaView

Es una aplicación de código abierto que permite la visualización de conjuntos de datos en dos o tres dimensiones. El tamaño de los conjuntos de datos que el software puede manejar es muy amplio dependiendo de la geometría y de la máquina en la cual se emplee ParaView, que va desde una máquina simple con un solo procesador hasta supermáquinas que trabajen en paralelo. El software permite recopilar una gran cantidad de datos y luego mostrar sus resultados.

Las principales ventajas del software son:

- (i.) Aplicación de visualización de código abierto.
- (ii.) Soporte para modelos computacionales distribuida que permite procesas grandes conjuntos de datos.
- (iii.) Licencia BSD flexible de tres clausulas.

Dentro de este software se deben realizar algunas modificaciones para poder visualizar los resultados esperados. Para el trabajo se utilizaron las siguientes herramientas:

- (i.) Contour: permite el uso de datos vectoriales o escalares.
- (ii.) Cutting: Extrae una región para su mejor visualización en el eje que se requiera.
- (iii.) Glyph: Permite realizar una figura en la cual se indican los vectores de las velocidades dentro del tanque.
- (iv.) Stream Tracer: Crea líneas para la visualizar el campo vectorial, en la cual nos indica su magnitud y dirección de flujo.

3. METODOLOGÍA

El presente trabajo está dividido en tres etapas.

- (i.) Preprocesamiento
- (ii.) Procesamiento
- (iii.) Post procesamiento

3.1. Preprocesamiento

Inicialmente, se definieron las diferentes variables para realizar la simulación, es decir se realizó la búsqueda de la información necesaria para alimentar el sistema, de tal manera que se definieron los siguientes parámetros.

3.1.1. Geometría del sistema

Se diseñó el tanque floculador de la planta piloto de UDLAPARK, la cual tiene los planos en AutoCAD. De los cuales se obtendrán las dimensiones para

posteriormente dibujar en el software Salome 9.4, en el cual se realizó un mallado del sistema hidráulico que posteriormente indicó como es el trayecto del flujo dentro del tanque.

Las dimensiones utilizadas para cada uno de los sólidos fueron las siguientes:

Tabla 5.

Dimensiones del tanque floculador

ID	Cara	Dimensiones			
		Altura	Diametro	Ancho	Largo
(i.)	Inlet	-	5.08 cm	-	5cm
(ii.)	Outlet	-	5.08 cm	-	5cm
(iii.)	Walls	70 cm	-	-	116cm
(iv.)	Atmosphere	-	-	56.2 cm	116 cm
(v.)	Propeller	5 cm	30cm	5cm	15cm
(vi.)	AMI	-	30cm	-	-

A continuación, se muestran las figuras obtenidos para cada una de las caras de la geometría realizada:

(i.) Inlet:



Figura 6. Entrada del tanque floculador

(ii.) Outlet

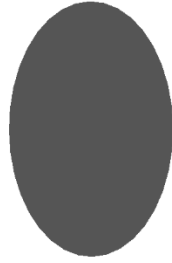


Figura 7. Salida del tanque floculador

(iii.) Walls

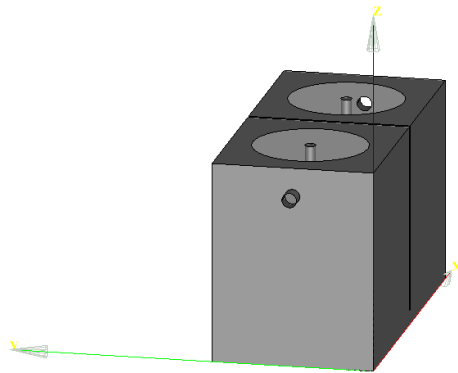


Figura 8. Paredes del tanque floculador

(iv.) Atmosphere

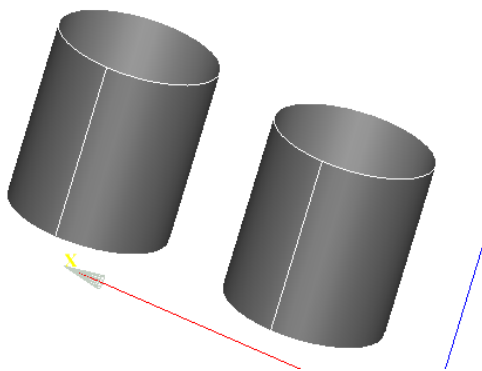


Figura 9. Atmósfera del tanque floculador

(v.) Propeller



Figura 10. Propellers del tanque floculador

(vi.) AMI

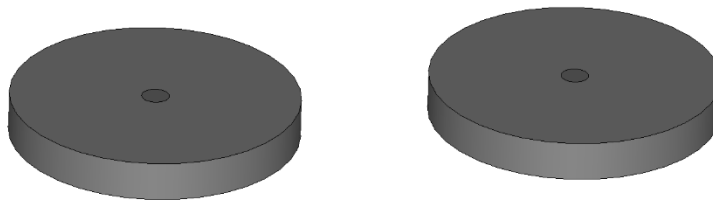


Figura 11. Sección de rotación del tanque floculador

La geometría final tiene el siguiente aspecto:

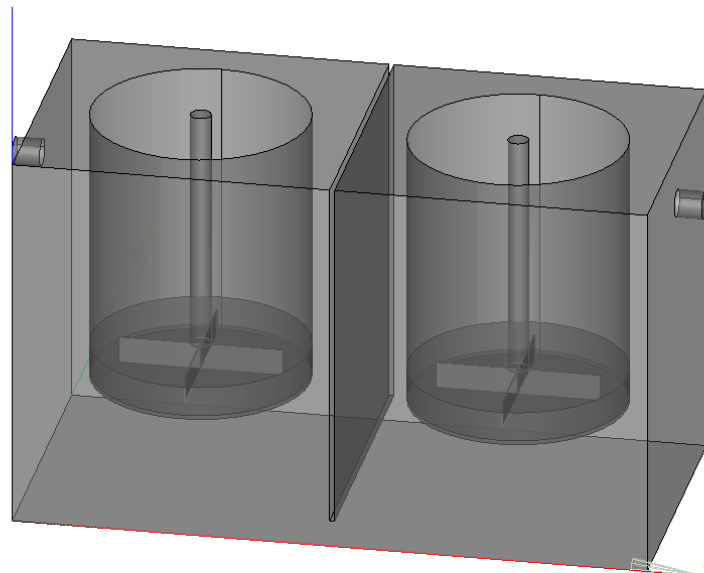


Figura 12. Geometría vista lateral

En la Figura 12 se observa la geometría utilizada, la cual muestra las dos cámaras de floculación con las siguientes dimensiones (1.16mx0.562mx0.7m)

En este caso se realizó la simulación a partir de un caso de propeller que es un moving mesh como se indica en la Figura 13 los propellers giran generar un choque entre las partículas y permitir que se aglomeren y ganen peso molecular.

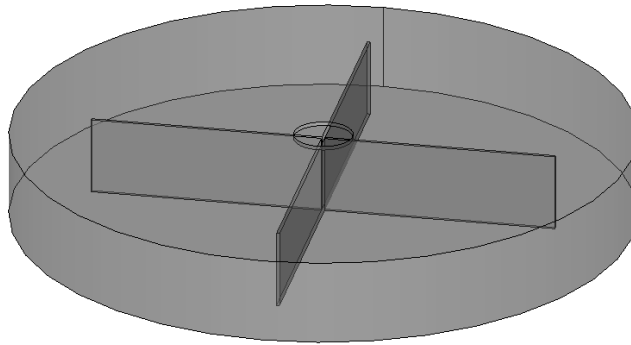


Figura 13. Propeller

Para realizar la geometría se estableció un mallado mediante el cual, se realizó la simulación del funcionamiento del tanque floculador. La malla que se genere definirá el porcentaje de efectividad del tanque en comparación al funcionamiento del tanque en la vida real, por lo que es muy importante definir bien el tamaño del mallado (Moreno, 2013).

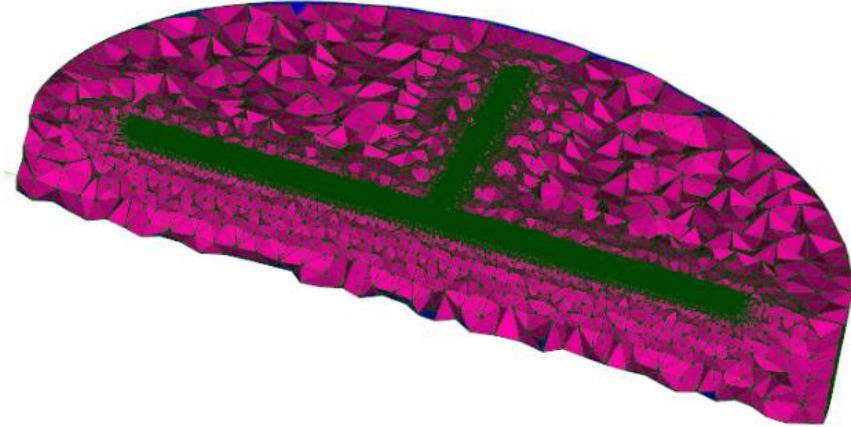


Figura 14. Mallado con corte en el eje y

Se muestra en la figura 14 el mallado realizado con una de las hélices del propeller en la mitad, ya que esta hecho un corte transversal para una mejor visualización. En esta etapa se debió configurar la calidad del mallado estableciendo diferentes capas a las hélices para un mejor procesamiento de la geometría.

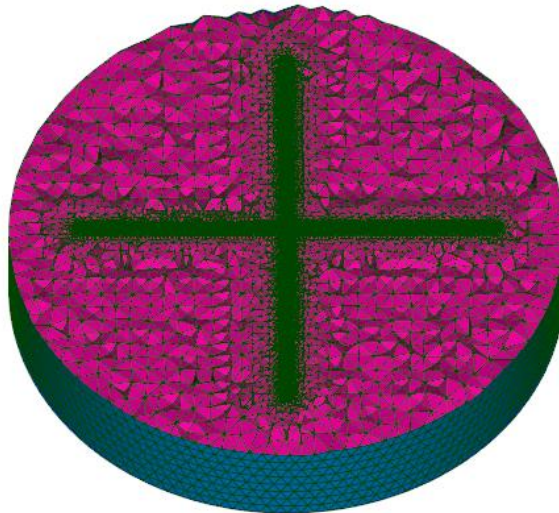


Figura 15. Capas del mallado en el propeller

Una vez realizado el mesh de la geometría, se exportaron los grupos y el mesh en formatos que OpenFoam y ParaView puedan leer (.unv y .stl). Con estos archivos

exportados hacia la carpeta de simulación se procedió a generar las condiciones del sistema para empezar la simulación.

3.1.2. **Condiciones iniciales del sistema**

Para esta simulación se realizó una mezcla entre dos diferentes tipos de mesh, el “snappy hexmesh” y el “moving” mesh. El “snappy hexmesh” permite generar mallas tridimensionales, las cuales contienen hexaedros automáticamente a partir de diferentes figuras geométricas.

Esta malla se aproxima a la superficie mediante el refinamiento iterativo de una malla inicial y la transformación de la malla hexagonal, es decir, esta malla se va adaptando a la capa inicial versus las iteraciones de las nuevas formas que va formando con el mesh (Greenshields, 2018).

El “moving” mesh requiere de un mapeo a partir de un dominio de espacios de forma irregular, mediante la conexión de puntos que permitan una simulación de los mismos en movimiento, en otras palabras, se generó una simulación con un objeto en movimiento, en este caso los propellers (Tang, 2005).

En estas condiciones iniciales se deben establecer los diferentes parámetros para que la simulación inicie. Para esto se deben realizar cambios en las distintas carpetas que se mencionan anteriormente. En el capítulo 3.1.3 se describen los cambios mencionados.

3.1.3. **Carpetas**

Una vez terminada la elaboración de la geometría y de exportarse los archivos necesarios para que OpenFoam los identifique, en este caso el archivo .unv y los archivos .stl de los grupos creados en Salome, se procedió a cambiar todos los directorios necesarios para arrancar con el solver. A continuación, se describen todos los directorios y subdirectorios en los que se realizaron los cambios necesarios.

Primeramente, dentro de la carpeta 0, existen dos archivos, los cuales hacen referencia a la velocidad y a la presión del sistema, en donde se establecieron las velocidades de entrada al tanque floculador; el caudal utilizado fue de 40 litros por segundo igual al de salida, ya que se trata sistema continuo, finalmente se estableció una gradiente de presión cero para las paredes. En este caso se debió transformar de caudal a velocidad lineal mediante la ecuación de caudal:

Ecuación 7

$$Q = A * V$$

Obteniéndose:

Ecuación 8

$$V = \frac{Q}{A}$$

Después de realizar los cálculos, la velocidad obtenida para el sistema fue de 0.33 metros por segundo en la entrada y salida del tanque, mientras que para las paredes se estableció que en las mismas no exista deslizamiento.

Como segundo directorio se tiene la carpeta constant, en la cual se deben establecer diferentes parámetros para la eficacia de la simulación. En el subdirectorio de polymesh se encuentra el archivo más importante, 'boundary', en el cual se establecieron las características de cada uno de los grupos de entrada, salida, paredes y atmósfera.

Otro de los archivos significativos es el DynamicMeshDict, el cual permite establecer los solver para que se pueda visualizar el caso de propeller en movimiento.

Finalmente, se corrigieron los archivos del directorio system, todos los archivos tienen Dict, los cuales definen las condiciones de la simulación. Dentro de los principales está el blockMesh, que define la geometría o la caja externa de la cual se deberá tomar en cuenta. El archivo control, es aquel que, definirá los tiempos de

simulación. CreatePatch, el cual permitirá generar los patches para el caso especial propeller y se deben proporcionar los parches para las dos diferentes figuras.

En este caso se debe generar un mallado especial de tipo de snappy Hexmesh, se incluyeron dos archivos más, los cuales definieron la cantidad de celdas y el surfaceFeatureExtract, que permitirá generar los emesh en tres dimensiones.

3.2. Procesamiento

Posteriormente, una vez terminada la geometría, se empleó el software OpenFoam 19.10, el cual permitió desarrollar la simulación gráfica, en la cual se establecieron los conceptos, para permitir que, mediante este software se desarrolle la simulación en códigos y consecuentemente la obtención de los resultados gráficos. Para esto se debió editar una numerosa cantidad de datos, los cuales permitieron realizar la simulación de forma adecuada, ya que, en caso de que algún valor no esté correcto, el software no permite que se continúe con la simulación.

Una vez con todos los directorios listos, en el software OpenFoam, se desarrollaron los siguientes pasos para llegar a la etapa final.

Primero, mediante un comando 'Touch' se generó un archivo de lectura para el software ParaView, en el cual se visualizaron las imágenes de resultados. Segundo, se ejecutó el comando ideasUnvToFoam que permitió al software leer el archivo importado desde Salome. Después, se inició la extracción de los archivos eMesh mediante 'surfaceFeature Extract' de los archivos .stl exportados previamente. Para generar un mallado dentro del cual se ejecutó el snappy Hexmesh se debió correr el comando blockMesh.

Una vez realizado estos comandos se procedió a generar el mesh mediante los siguientes comandos.

- (i.) `mpiexec -n 6 snappy Hexmesh-parallel`
- (ii.) `setFields`
- (iii.) `mpiexec -n 6 interFoam-parallel`

Para este proyecto el algoritmo utilizado fue pimpleFoam, el cual se basa en una serie de ecuaciones lineales y de diferentes requerimientos para que el software procese y solucione el proyecto. Los principales requerimientos necesarios para iniciar el algoritmo son la presión cinética y la velocidad. Las opciones lineales de el solver para resolver la matriz $Ax=b$ se resuelve de las siguientes formas

Ecuación 9

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

$$\mathbf{A}^{-1} = \frac{1}{\text{diag}(\mathbf{A})}$$

(Greenshields, 2014).

En este caso el método de solución utilizado es el de Gauss-Seidel, el cual se indica en el directorio system en el archivo fvSolution con una tolerancia baja de 0.02.

Debido al detalle mediante el cual se desarrolló la geometría del sistema, es muy importante realizar la simulación en paralelo, que tiene por objetivo realizar una descomposición del dominio físico, y en este caso dividir la geometría realizada en seis partes debido a que la máquina utilizada posee seis procesadores. Para lograr liberar los diferentes procesadores se ejecutó el comando de mpiexec, el cual permite activar la simulación en paralelo y para el proyecto el comando utilizado fue:

Mpiexec -n 6 interFoam-parallel.

Una vez escrito este comando en la herramienta, OpenFOAM realiza la simulación con el número de procesadores especificados (seis) y en el sentido que se le indique (parallel).

De igual manera una vez finalizada la simulación en paralelo se ejecutó un comando para reconstruir la información generada en los seis diferentes procesadores para

su posterior visualización, el cual su función es unir los directorios creados en un solo dominio.

3.3. **Postprocesamiento**

En la última etapa, una vez que todos los parámetros hayan sido fijados y el software haya terminado de realizar la simulación, se visualizó la simulación desde un software complementario en este caso ParaView 5.6, en el cual se evidenció la interfaz gráfica de la simulación que se realizó previamente. Después de obtenido el resultado, se establecieron diferentes procesos para obtener la simulación gráfica del fluido y su comportamiento dentro del tanque floculador. Finalmente, con todos los datos calculados dentro del proceso se determinó el tiempo medio de residencia del agua en el tanque.

Dentro del ParaView se establecieron los diferentes parámetros como por ejemplo dentro de Contour se estableció un valor de 0.5 en el rango de agua/ aire, se modificaron los valores de rangos de velocidades y las escalas para las grillas mediante las cuales se estimó el tiempo de retención hidráulico.

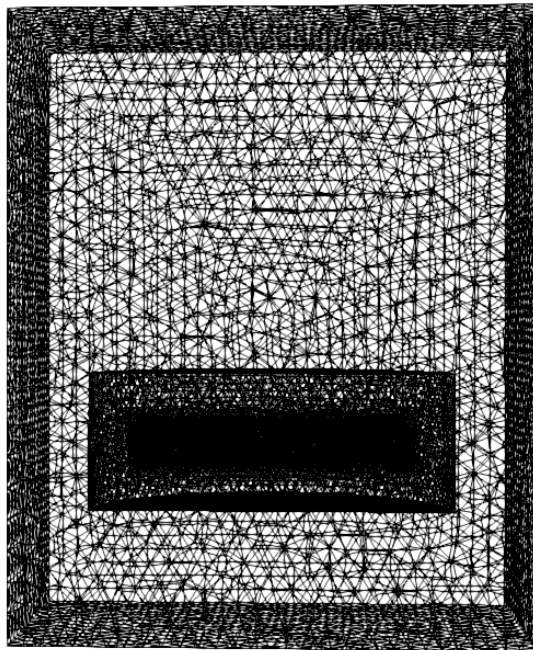


Figura 16. Visualización del mallado en ParaVlew

Se muestra la vista en corte del tanque floculador con uno de sus propellers, después de haber realizado el mallado y ejecutado el comando Clip.

Dentro de este capítulo, se definieron los residuales que se obtuvieron durante el procesamiento mediante graficas que se muestran en el siguiente capítulo, los cuales se obtuvieron mediante un nuevo software llamado GnuPlot 5.2. Finalmente, se aplicó una serie de fórmulas para diferentes tramos del tanque floculador y obtener el tiempo de retención hidráulico del tanque.

4. RESULTADOS Y ANÁLISIS

Dentro de este capítulo se evaluaron y analizaron los resultados, mediante diferentes gráficas y figuras en distintos softwares, además se estimó el tiempo de retención hidráulico del agua en el tanque floculador.

4.1. Estabilización de Residuales

En esta etapa mediante un comando utilizado en OpenFoam, se obtuvieron los valores residuales de estabilización en los diferentes vectores de velocidad: U_x , U_y y U_z . A continuación, se muestran las gráficas con los resultados obtenidos y su respectivo análisis.

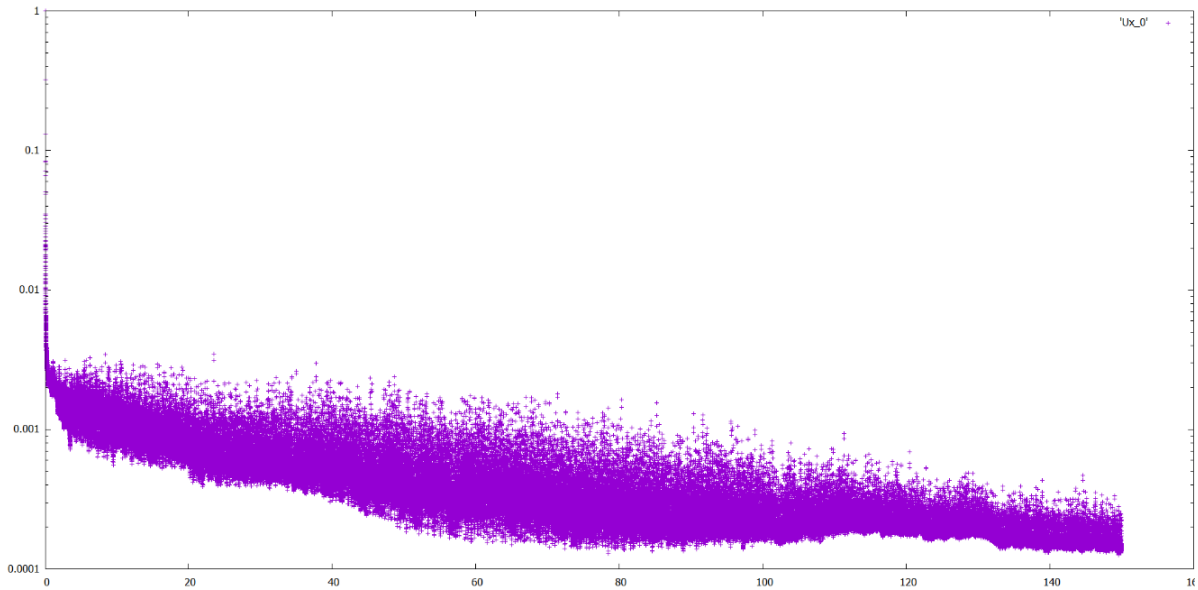


Figura 17. Velocidades en el eje x

Como se observa en la Figura 17 el tiempo de estabilización llega cerca a los 110 segundos y posterior a esto se hace casi una línea de tendencia recta, lo que corresponde a la estabilización de los residuales.

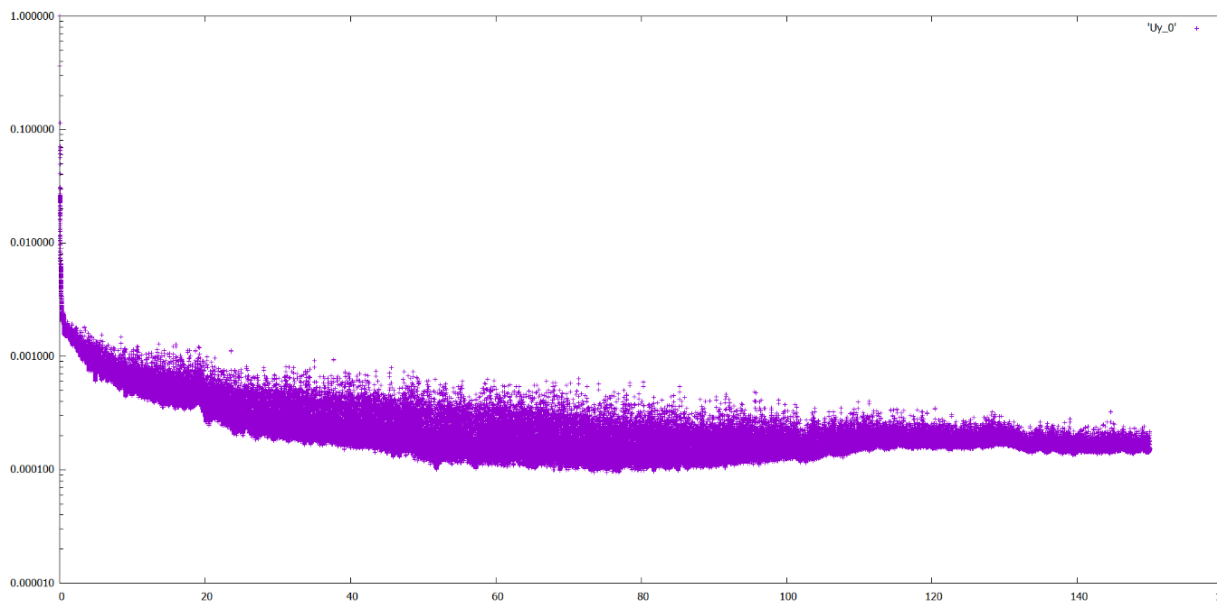


Figura 18. Velocidades en el eje y.

De igual manera en la Figura 18 se indican los valores de estabilización de los residuales alrededor de los 110 segundos, en general los valores en los tres vectores deberían ser muy parecidos como se indican en las gráficas respectivas.

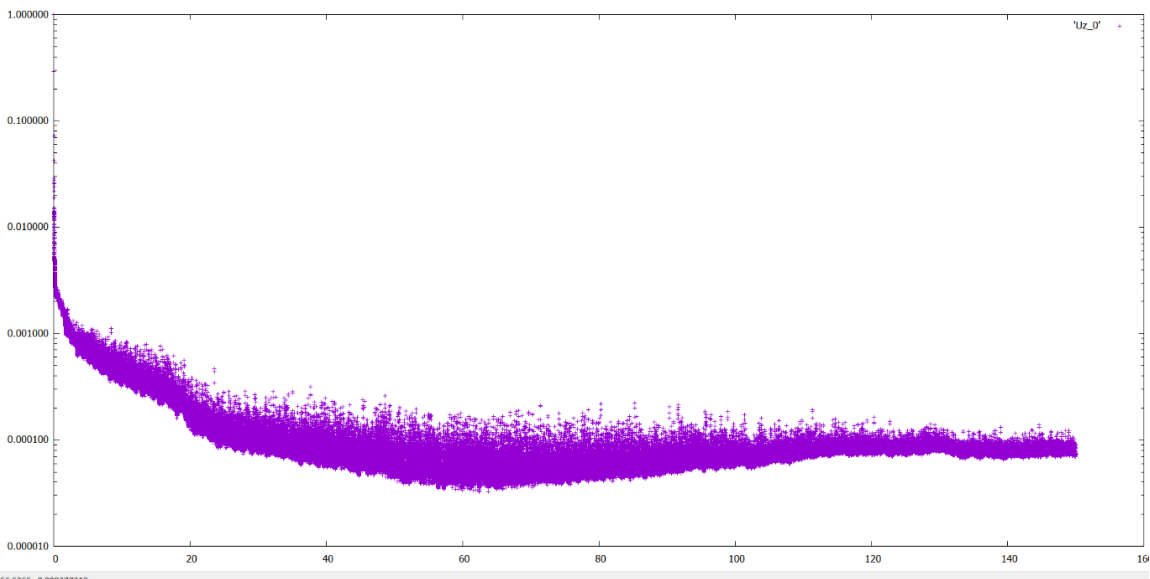


Figura 19. Velocidades en el eje z.

En este caso es un poco más notorio la estabilización de la curva a los 110 segundos respecto a los graficados en el eje 'x', por lo que este es el tiempo que se ingresó

en el software ParaView para proceder con el cálculo de los TRH, y el análisis de las velocidades.

4.2. ParaView

Dentro del software se configuraron distintos parámetros para obtener diferentes vistas según lo necesario para sacar los resultados, en esta primera figura se muestra el tanque flocculador con el nivel de agua habitual, es decir casi lleno, mediante el cual se obtuvieron los demás resultados.

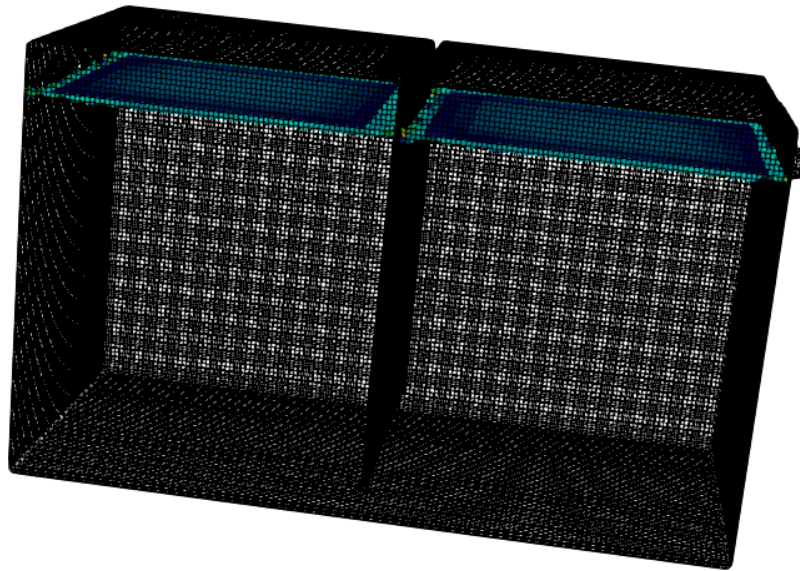


Figura 20. Tanque con el nivel de agua especificado

Mediante el comando clip se realizó el corte en el eje 'x' para poder analizar las velocidades dentro del tanque transcurridos los 110 segundos de estabilización.

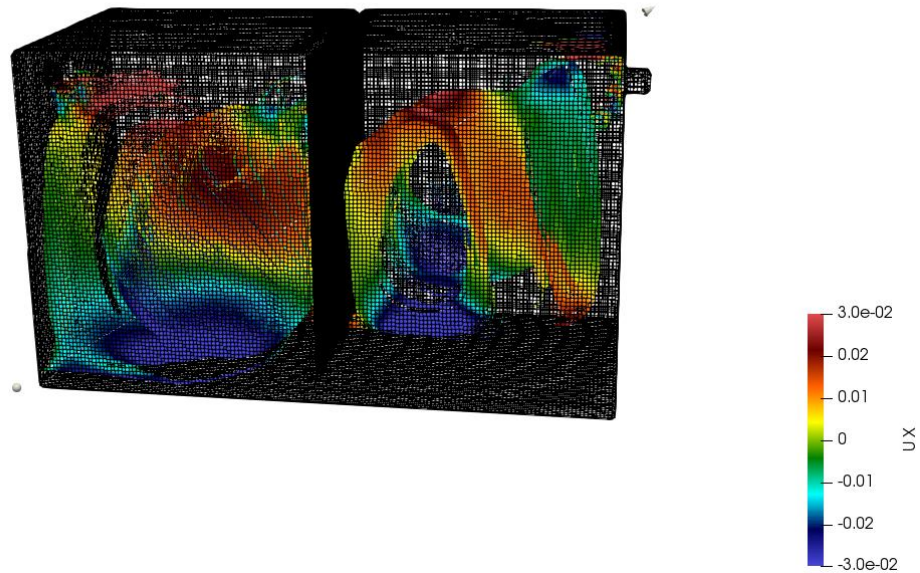


Figura 21. Dirección del campo vectorial en x.

En la Figura 21 se muestran las diferentes velocidades del eje x dentro del sistema, las que se explican en la sección 4.3 pero de manera general el esquema presente en el eje x tiene velocidades que oscilan entre valores de 0.02 y -0.02.

De esta manera se presenta un plano general de los resultados obtenidos en la simulación, luego de los 110 segundos de estabilización de los residuales detallados anteriormente.

Mediante el comando Glyph, se establecieron los vectores de velocidad en el eje x, a fin de ir definiendo dichos vectores, se modificaron los diferentes tamaños de flechas hasta encontrar una vista clara.

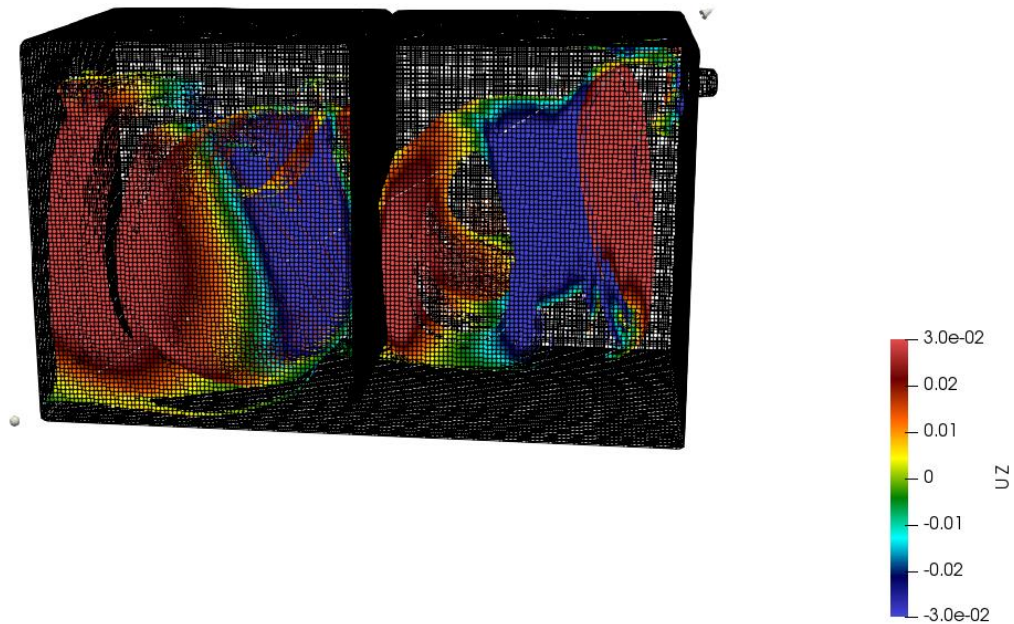


Figura 22. Dirección del campo vectorial en z.

En la Figura 22 se indican los valores de una manera más gráfica que como se comporta el agua dentro del tanque en el eje z. De manera general es el resultado de la simulación tras los 110 segundos de estabilización, como se observa, el comportamiento en los extremos del tanque el agua tiende a subir, mientras que, en el centro el agua baja.

4.3. Análisis de velocidades

En el estudio realizado se estableció que el agua se mueve en dos dimensiones, en el eje x y en el eje z, limitándose el análisis a estos dos casos.

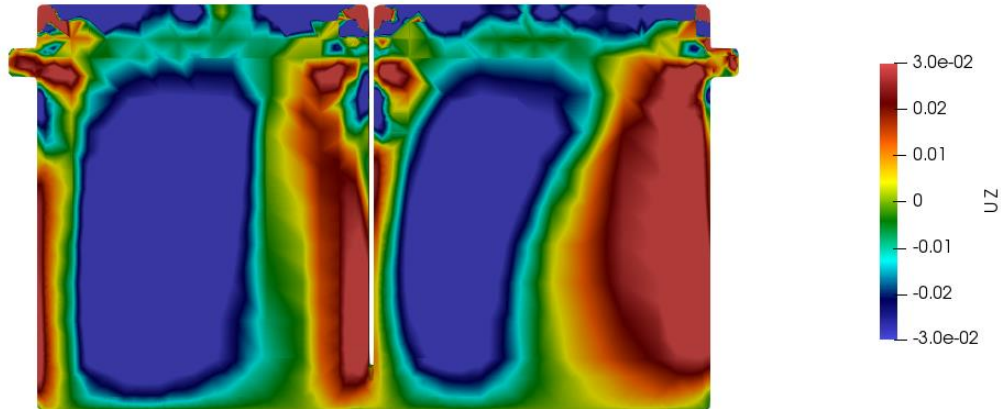


Figura 23. Análisis de velocidades en z.

En la Figura 23 se muestra el comportamiento de las velocidades dentro del tanque floculador. Como se indica en la leyenda, todas las velocidades en el eje positivo son aquellas que su dirección es hacia abajo, mientras que aquellas que están en el eje negativo son velocidades con dirección hacia arriba. Como se puede observar en la figura existen cerca de ocho diferentes tramos o subdivisiones en los que el agua sube o baja para este eje.

La leyenda indica que los tonos rojizos amarillentos son aquellas zonas en las cuales el agua tiende a subir. En ciertas áreas se encuentran tonos de verde, lo cual indica que estas podrían ser zonas 'muertas' o que a su vez el agua llega a estancarse y no tiene movimiento, en la vida real no suele pasar debido al resto de movimiento del tanque ocasiona que el agua se mueva y no quede estancada. Finalmente, las zonas de tonos cianes y azules, son aquellas zonas en las que el

agua tiende a bajar, y esto se debe a que es la zona en la cual el mayor número de partículas se aglomeran.

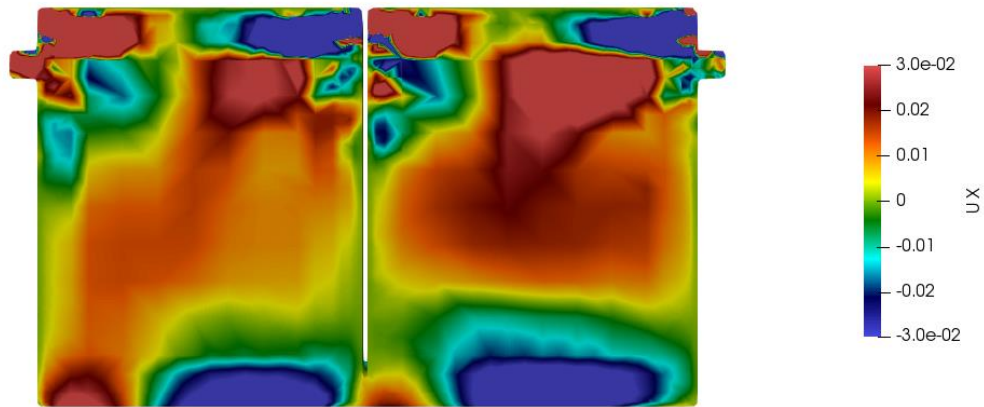


Figura 24. Análisis de velocidades en x.

En la Figura 24 se indica el comportamiento de las velocidades dentro del tanque floculador en el eje x. Las velocidades positivas son aquellas que se desplazan de izquierda a derecha, mientras que las velocidades negativas son aquellas que se desplazan de derecha a izquierda, cabe recalcar que los signos son utilizados para distinguir la dirección del flujo.

Como se indica en la leyenda y en la figura, las zonas rojizas y amarillentas son en la entrada en la parte izquierda del tanque debido a la velocidad en la que entra el agua y, una vez que pasa al segundo tanque, de igual manera el agua va de izquierda a derecha por la entrada de cada una de las cámaras del tanque. De igual manera las zonas con colores verdes, muestra aquellas zonas en las cuales, dentro del eje x, el agua permanece inmóvil, es decir no se mueve hacia ninguna dirección y finalmente, las zonas en las cuales tienen colores azules, o parecidos son aquellas zonas en las cuales el agua se desplaza hacia la izquierda, que como se evidencia en la figura son zonas puntuales y abajo del tanque, esto es porque el flujo de entrada y de salida del tanque en general es en la dirección de izquierda a derecha.

4.4. Tiempo de Retención Hidráulico

Para la determinación del TRH se dividieron en tramos o zonas a cada una de las diferentes secciones de colores dentro del tanque. En este caso la movilidad del agua es en dos dimensiones 'z' y 'x'. A continuación, se detalla el procedimiento para la obtención del tiempo en el cual el agua permanece dentro del tanque, adicionalmente se verificó que el tiempo en cada una de las cámaras sea el mismo.

Para la determinación del TRH se utilizó la siguiente ecuación:

Ecuación 10

$$t = \frac{d}{v}$$

Donde:

t= Tiempo

d= Distancia

v= Velocidad

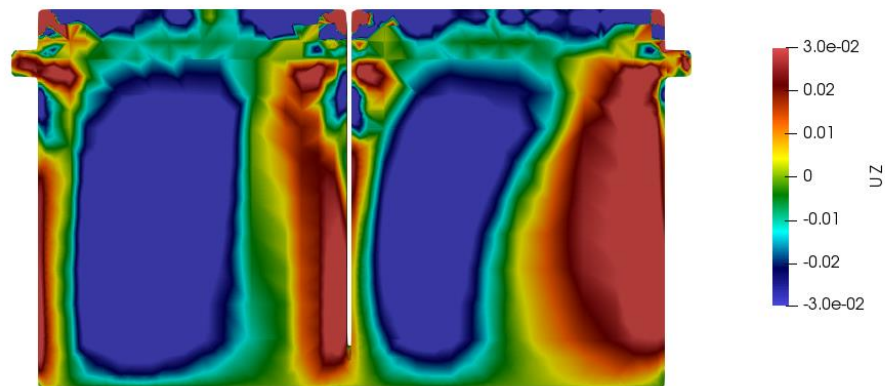


Figura 25. Determinación tramos TRH en z.

En este caso de la Figura 25 el eje tomado en consideración es el 'z', en donde se realizó una división de 7 zonas, mediante las cuáles se calculó el TRH en el sentido de flujo del agua para arriba o para abajo. Como se puede observar en el margen izquierdo se encuentra la altura del tanque, por lo que se utilizaron los siguientes datos para la estimación del tiempo

Tabla 6.

Tramos TRH eje z

Distancia	Velocidad
0.55	0.03
0.55	0.005
0.45	0.03
0.55	0.015
0.55	0.1
0.45	0.03
0.55	0.005

Una vez realizados los cálculos mediante la ecuación 3 se obtuvo que en eje 'z' el agua permanece 310.5 segundos, es decir alrededor de 5.17 minutos, debido a la recirculación del agua dentro del tanque y el movimiento de los propellers.

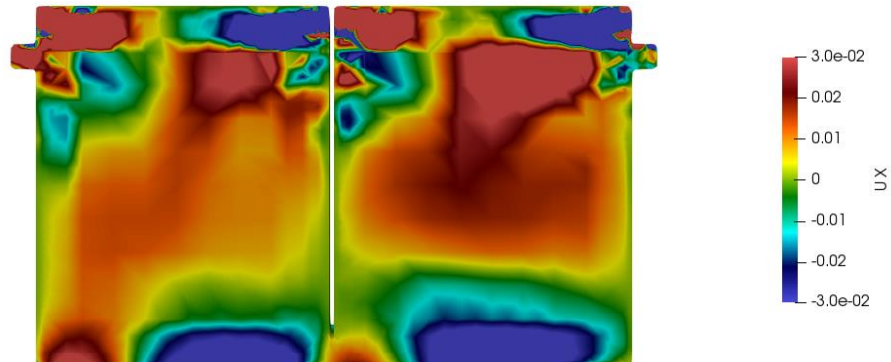


Figura 26. Determinación tramos TRH en x.

En la Figura 26 se muestran las velocidades dentro del tanque en el eje 'x'. A continuación, se muestra la tabla con los valores utilizados para la determinación del tiempo de retención del agua en el tanque.

Tabla 7.

Tramos TRH eje x

Distancia	Velocidad
0.3	0.03
0.15	0.03
0.3	0.013
0.1	0.03
0.35	0.03
0.07	0.03
0.45	0.005
0.1	0.001

En este caso después de realizar los cálculos con la ecuación 3 se obtuvo un valor de 245.41 segundos en el tanque en el eje x, que equivale a 4.1 minutos.

Realizando la suma de los dos tiempos en sus respectivos vectores, se obtuvo un tiempo final de 556 segundos del agua en su paso dentro del tanque, equivalente a 9.26 minutos. Tiempo que no se aleja mucho del tiempo utilizado como parámetro de diseño del prototipo, mismo que corresponde a 6.46 minutos

5. CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

Se realizó el diseño de una geometría de un tanque floculador de la planta de tratamiento de aguas residuales piloto de UDLAPARK, mediante el software Salome 9.4, el cual permitió por medio de una interfaz simple realizar el diseño del tanque en sus tres dimensiones con las condiciones de borde, entrada, salida, paredes, y atmósfera. Los valores utilizados para la velocidad y presión influyeron mucho en la eficiencia de la simulación, así también como los valores determinados para el tamaño y diseño del mallado de la geometría, debido a que, si eran valores muy grandes perdía similitud con la realidad, y al contrario, si era muy pequeño las celdas del mallado, la máquina no hubiese podido completar la simulación en el tiempo establecido.

Las dimensiones del tanque fueron tomadas de un proyecto previo “Diseño de un sistema de coagulación- floculación para el tratamiento de aguas residuales” de Daniela Garcés realizado en el año 2019 y en comparación a la simulación realizada se obtuvo una alta eficiencia dentro del tanque, ya que el mismo presenta muy pocos puntos o zonas muertas, con las dimensiones utilizadas en el prototipo.

El proyecto se desarrolló a través del solver pimpleFoam para este caso se utilizó el modelo de turbulencia RANS, ya que las condiciones del modelo permitieron obtener resultados más exactos en una menor cantidad de tiempo para simular el movimiento del agua dentro del tanque.

Tras los 110 segundos de simulación los residuales entraron en un estado estacionario, donde los residuales se estabilizaron, sin embargo, se tomó el valor

de los 115 segundos para realizar los resultados para asegurarse que el estado estacionario sea estable. El tiempo total de simulación fue de 150 segundos.

Tras realizar la simulación del funcionamiento hidráulico del tanque floculador se obtuvieron las diferentes figuras de la 23 a la 26, en las cuales se demuestra el comportamiento del agua, en las zonas en las cuales, el agua baja o a su vez sube, dependiendo de la velocidad y así se evidencia las características del tanque floculador que permite la aglomeración de las partículas mas pequeñas.

El tiempo de retención hidráulico fue de un total de 556 segundos entre sus dos diferentes dimensiones. Un total de 310.5 segundos en el eje z y un total de 245.5 segundos en el eje x. Además, se verificó que el agua permanezca la misma cantidad de tiempo en cada una de las dos cámaras.

De igual manera, se determinó que cualquier alteración de alguno de los parámetros establecidos inicialmente puede afectar la eficiencia del tanque floculador. Como principal parámetro de simulación, si la velocidad es muy alta o baja el software no permitió que se continúe con la simulación. De igual manera se debió realizar transformaciones de las unidades, debido a que si en algún punto existía un valor con otra unidad el software detenía la simulación. Otro factor muy importante es las características de la máquina en la cual se realiza la simulación, en este caso debido a los 6 procesadores y la memoria RAM, se logró realizar la simulación en 48 horas seguidas.

Se concluye que la simulación fue bien realizada, debido a que en comparación con el prototipo las condiciones de borde fueron bien llevadas y se establecieron de manera idónea, a su vez los parámetros establecidos como el tamaño de la malla, los tiempos de simulación y el diseño de la geometría permitieron visualizar el funcionamiento ideal del tanque.

5.2 Recomendaciones

Se recomienda que, para futuras simulaciones, se tomen en cuenta las características de la máquina, ya que si no posee ciertas características pueden

presentarse diferentes problemas, como un tiempo de simulación mucho mas extenso de lo esperado, daños hacia el disco de la máquina o incluso que la máquina llegue a un estado crítico.

Además, se propone tomar en cuenta las dificultades del diseño y para futuras simulaciones no olvidar que cada uno de los valores establecidos para las carpetas 0, system y constant, deben tener coherencia con los datos del proyecto, entre ellos caudales, áreas, presiones, longitudes, entre otros.

REFERENCIAS

- Alejandro, G., & Bolaños, I. (2015). *Dinámica de fluidos computacional: una herramienta versátil para el desarrollo de la ciencia y la tecnología*. (166).
- Andía, Y. (2000). Tratamiento de agua coagulación y floculación. *Sedapal*, 1–44. Retrieved from http://www.sedapal.com.pe/c/document_library/get_file?uuid=2792d3e3-59b7-4b9e-ae55-56209841d9b8&groupId=10154
- Cardiff, P. (2017). Introduction to Meshing in OpenFOAM. *University College Dublin*, m(May).
- Cristina, N., & Camacho, C. (2011). *Tratamiento de agua para consumo humano*. 153–170.
- Date, A. W. (2005). Introduction to computational fluid dynamics. *Introduction to Computational Fluid Dynamics*, 9780521853, 1–377. <https://doi.org/10.1017/CBO9780511808975>
- Dekanski, C. W. (1993). *Design and Analysis of Propeller Blade Geometry using the PDE Method*. (August).
- Figura, L. (n.d.). *Hélice*. 50–56.
- Greenshields, C. (2014). *The OpenFOAM 7 User Guide*. (February), 212.
- Greenshields, C. (2018). OpenFOAM v6 User Guide: 5.4 Mesh generation with snappyHexMesh.
- Lopez, C. M. (2001). Tratamiento biológico de aguas residuales. In *Ingeniería Química* (Vol. 33).
- López Vásquez, C., Buitron Méndez, G., García, H., & Cervantes Carrillo, F. (2017). *Tratamiento biológico de aguas residuales: Principios, modelación y diseño*. <https://doi.org/10.2166/9781780409146>
- Lozano, W. A. (2016). *Diseño de Plantas de Tratamiento de Aguas Residuales Autor: Ing. WILLIAM ANTONIO LOZANO-RIVAS, MSc, PhD BOGOTÁ D. C., COLOMBIA Octubre de 2012*. (March).
- Moreno, J. (2013). *Desarrollo y Optimización de Un Generador de Mallas Superficiales y/o Volumétricas para Aplicaciones de Simulación Electromagnética*.
- Rahman, M. M., Agarwal, R. K., & Siikonen, T. (2016). RAS one-equation turbulence model with non-singular eddy-viscosity coefficient. *International Journal of Computational Fluid Dynamics*, 30(2), 89–106. <https://doi.org/10.1080/10618562.2016.1164847>
- Tang, T. (2005). *Moving mesh methods for computational fluid dynamics*.

0000(Mmm), 141–173. <https://doi.org/10.1090/conm/383/07162>

Unemi, R. C. (2016). *Calidad del agua destinada al consumo humano en un cantón de Ecuador* *Quality of water intended for human consumption in a canton of Ecuador*. 9, 109–117.

van Leer, B., & Powell, K. G. (2010). Introduction to Computational Fluid Dynamics. *Encyclopedia of Aerospace Engineering*, (December 2010). <https://doi.org/10.1002/9780470686652.eae048>

Xia, B., & Sun, D. W. (2002). Applications of computational fluid dynamics (CFD) in the food industry: A review. *Computers and Electronics in Agriculture*, 34(1–3), 5–24. [https://doi.org/10.1016/S0168-1699\(01\)00177-6](https://doi.org/10.1016/S0168-1699(01)00177-6)

