IMT2116: Taller de matemáticas aplicadas

Profesor: Alejandro Cataldo

Ayudante: Andrés Moraga

Informe Proyecto 6

Empresa: Holon SPA.

Integrantes: Martín Alcántara, Vicente Iligaray, Rocío Ladrón de Guevara, Vicente Olivares y Javiera Sanhueza

23 de diciembre, 2024

Índice

1. Introducción

1.1. La empresa

El Centro de Investigación en Recursos Naturales HOLON SpA. es una consultora que presta asesoría a diversas empresas y entidades gubernamentales, principalmente sobre el manejo sostenible de recursos hidrobiológicos, la evaluación de impacto ambiental sobre el borde costero, y la conservación de ecosistemas.

Dentro de los servicios que ofrecen, se encuentra el estudio del comportamiento de difusión de plumas de descarga. Se entiende que una pluma es una columna de fluido que se mueve a través de otro fluido. De esta manera, se estudia la difusión del fluido descargado por los clientes en un medio marítimo o el cauce de un río.

Su relevancia consiste en estudiar si los sedimentos o contaminantes del efluente logran disolverse según los estándares ambientales permitidos. En este contexto, se refiere a efluente como curso de agua que descarga vertidos utilizados en procesos productivos.

1.2. El problema

Los estudios de descarga de efluentes se realizan mediante simulaciones computacionales dado que medir la pluma físicamente es complejo y costoso (CITA). De este modo, para realizar el modelo se deben obtener los parámetros necesarios para definir los datos de corriente para las cuatro etapas de fase lunar y mareas, escenarios que serán modelados y simulados por separado.

Sin embargo, el equipo de la empresa carece de especialización en modelos matemáticos. Por lo tanto, para crear una modelación numérica de los distintos casos, se utilizan programas computacionales como *Visual Plumes*, *Cormix*, *Visjet* y *OpenFOAM*. Por el estado del arte inicial para realizar simulaciones de dos líquidos incompresibles y miscibles con uno o más difusores, se utiliza *Cormix*. Este programa es relativamente sencillo de utilizar, pero poseía una baja precisión, utilizaba modelos simplificados y extremadamente acotados. De esta forma, la empresa ha migrado al programa de código abierto *OpenFOAM*, dado que les permite analizar la hidrodinámica tridimensional de manera más completa mediante su enfoque en Dinámica de Fluidos Computacional (CFD, por sus siglas en inglés) y resolución de ecuaciones diferenciales parciales de tres dimensiones espaciales y una temporal mediante el método de

volúmenes finitos. Dada la inexperiencia en el modelado computacional, no cuentan con las herramientas necesarias para complejizar el modelo ni pueden garantizar un uso óptimo y adecuado del programa, y optan por heredar los modelos sobre simplificados utilizados en *Cormix*, esto resulta en altos costos computacionales, la falta de certeza de sus resultados y poca adaptabilidad a otros contextos, además de prescindir de parámetros como mayor cantidad de difusores, turbulencia, o un mallado más fino.

Además, dado que se busca visualizar la estabilización del sistema, se debe iterar la simulación la cantidad de veces necesarias para alcanzar la convergencia. Actualmente, HOLON realiza este proceso mediante prueba y error, esto contribuye a aumentar los costos computacionales hasta demorar sus tiempos de cómputo más de una semana, lo cual se debe repetir varias veces para el mismo escenario con condiciones distintas, incluyendo de manera base los cuatro escenarios de marea mencionados anteriormente.

2. Objetivos

Los objetivos de nuestro proyecto son optimizar la implementación computacional de problema y complejizar el modelo utilizado para que pueda adaptarse a situaciones específicas que requieran de mayor precisión. Para alcanzar este objetivo, se planteó inicialmente la implementación criterios de convergencia que permitan detener la simulación, evitando iterar de manera innecesaria. Por otro lado, se planteó la reformulación del modelo a uno más detallado que pueda adaptarse mediante parámetros a los distintos contextos donde se quiera aplicar.

Así, los entregables del proyecto son un código ejecutable de C++ que implemente el criterio de convergencia a la modelación del problema a través de *OpenFOAM*, obteniendo un menor tiempo de cómputo de la resolución de cada caso, junto a un informe técnico que recopile y documente la investigación realizada.

3. Datos

Se asume que los datos recibidos para formar el mallado del modelo se encuentra bien configurado. Esta suposición potencialmente puede ser una fuente de dificultades al solucionar el problema.

Estos datos sobre la corriente consideran los cuatro escenarios de fase lunar sicigia (luna llena y luna nueva) y cuadratura (cuarto creciente y cuarto menguante) en mareas llenante y vaciante. Son obtenidos mediante medición manual *in-situ* o uso de la base

de datos velada por Servicio Hidrográfico y Oceanográfico de la Armada de Chile (SHOA), y se les realiza un análisis estadístico en función del tiempo para usar como datos de entrada en cada uno de los escenarios.

Para realizar el proyecto, contamos con los datos base utilizados y su implementación en *OpenFOAM* de dos modelos; uno con descarga en río modelado en dos dimensiones (llamado Modelación 1), y otro con descarga marítima tridimensional (llamado Modelación 2). Junto a esto, contamos con los informes de estudio respectivos a cada caso.

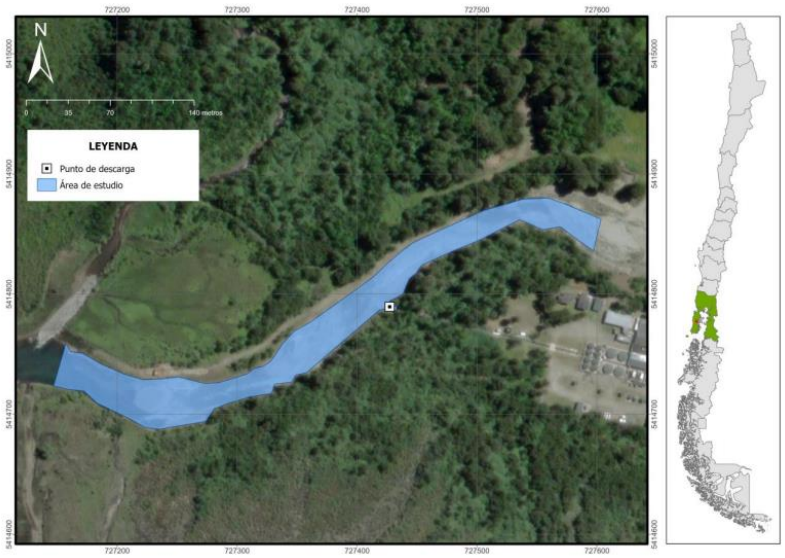

Figura 1: Área de estudio, modelo 2D. Modelación 1. Obtenido de HOLON SpA, 2024.

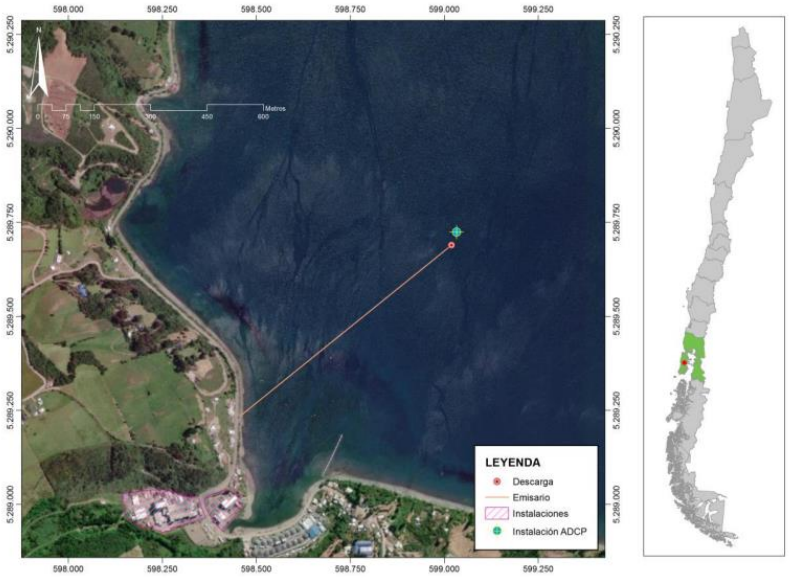

Figura 2: Área de estudio, modelo 3D. Modelación 2. Obtenido de HOLON SpA, 2022.

No contamos con datos de casos distintos a los dos anteriores para tener más iteraciones para probar el modelo. Asimismo, considerando la nula disponibilidad de

datos reales de plumas de descarga, no se podrá convalidar los resultados simulados del modelo.

Por un lado, los datos base consisten en una descripción del sistema de descarga, características del efluente y del medio receptor. En el sistema de descarga se tiene la cantidad de difusores, el diámetro externo del difusor, grosor del tubo, caudal máximo y profundidad del emisario. Las características del efluente consisten en la temperatura promedio, densidad promedio y las concentraciones máximas de sus diversos contaminantes. Las características del medio receptor incluyen temperatura promedio, densidad promedio, salinidad promedio y las concentraciones de los contaminantes. Las dinámicas de corriente se obtienen tras los análisis estadísticos de los registros de fases lunares de sicigia y cuadratura en mareas llenante y vaciante.

SISTEMA DE DESCARGA		
Número de difusores	5	
Diámetro externo difusor	292	mm
Grosor del tubo	11,5	mm
Caudal máximo	220	m^3/hora
Profundidad emisario	35,5	m

EFLUENTE		
Concentración de SST	300[a]	mg/l
Concentración de DBO_5	201[b]	mg/l
Concentración de NTK	60[b]	mg/l
Concentración de ColFec	1000[b]	NMP/100ml
Temperatura promedio	8	°C
Densidad promedio	1.008	kg/m^3

a: concentración límite del DS90 Tabla 5

MEDIO RECEPTOR		
Concentración de SST	8	mg/l
Concentración de DBO_5	2	mg/l
Concentración de NTK	0,3	mg/l
Concentración de ColFec	1,8	NMP/100mml
Temperatura promedio	11,3	°C
Salinidad promedio	33	PSU
Densidad promedio	1.025	kg/m^3

Tabla 1: Datos base para la modelación 2. Obtenido de HOLON SpA, 2022.

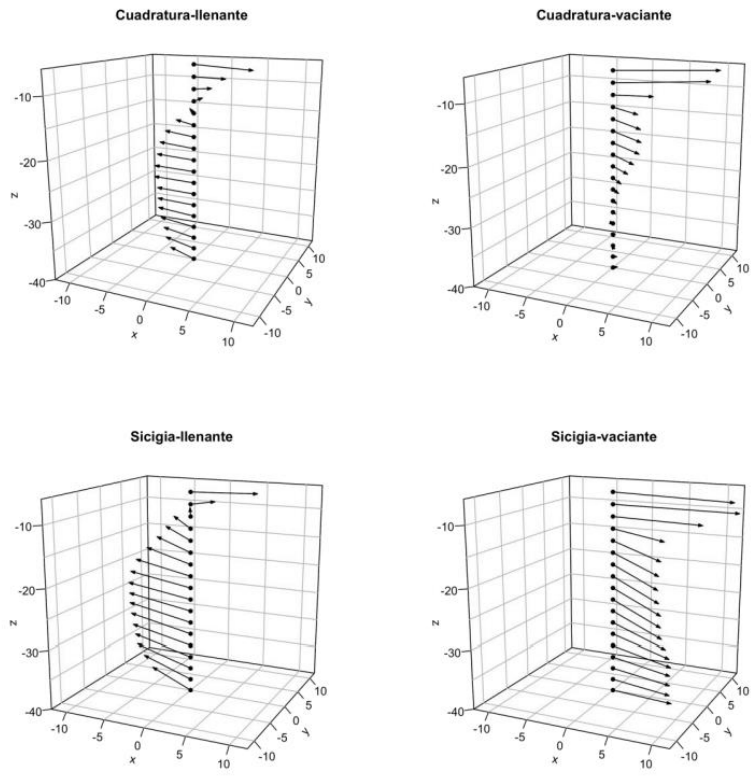

Figura 3: Espiral de Ekman de corrientes durante escenarios de llenante y vaciante, en fase lunar de sicigia y cuadratura para la modelación 2. Obtenido de HOLON SpA, 2022.

La implementación del modelo consiste en un código compilado de *OpenFOAM*, librería de código abierto utilizada principalmente en el área de CFD. Este contempla carpetas ordenadas de los cuatro escenarios a estudiar. Dentro de estas se encuentran: una carpeta *0* del estado inicial del modelo, una carpeta *constant* que contiene los datos base en formato de diccionario, una carpeta *system* que contiene especificaciones internas para los solver y una carpeta llamada *controlDict* que contiene las condiciones para detener la simulación. También, dentro de las carpetas de los escenarios existen archivos *alpha*.medio, donde medio cambia por *air*, *other* o *water*, con las variables de su respectivo medio, y un archivo de extensión *.unv* que describe la grilla del espacio ocupado.

Nombre	Tamaño	Comprimido	Tipo
..			Carpeta de archivos
0			Carpeta de archivos
constant			Carpeta de archivos
system			Carpeta de archivos
alpha.air	1.489	392	Archivo AIR
alpha.other	1.843	453	Archivo OTHER
alpha.water	1.969	436	Archivo WATER
.DS_Store	6.148	296	Archivo DS_STORE
Rauco_final4.unv	3.519.189	492.701	Archivo UNV

Figura 4: Esquema de carpetas para la implementación de OpenFOAM en uno de los escenarios del caso tridimensional (Modelación 2).

4. Metodología

Se realizó una serie de reuniones con los integrantes de HOLON Spa encargados del análisis de corrientes, modelación en *Visual Plumes* y *OpenFOAM*, Aldo Hernández y Daniela Henríquez, ambos biólogos marinos, quienes nos explicaron el contexto de la situación a resolver y el paso a paso de la modelación e implementación del problema en dichos programas, junto con proveernos de los datos, la aplicación del modelo y los resultados de este para dos casos.

Luego, se hizo un estudio sobre el estado del arte del problema a nivel teórico y computacional, donde se profundizó en los algoritmos implementados en *OpenFOAM*, los modelos matemáticos detrás de estos y las ecuaciones que los respaldan en contextos como el presentado.

Los datos entregados por la contraparte fueron utilizados para realizar simulaciones iniciales cuyos resultados posteriormente serán utilizados como referencia para la validación del trabajo realizado.

La investigación sobre *OpenFOAM* reveló que era inviable para nuestro equipo reducir significativamente los tiempos de computación de cada iteración optimizando el uso de los parámetros del programa. Por esta razón se descartó la posibilidad de abaratar el cálculo de cada iteración.

Por otro lado, el trabajo de investigación permitió plantear una hipótesis para la confección del criterio de convergencia basándose en la comparación de los momentos estadísticos de cada iteración

Con el fin de entender de mejor manera la modelación y la convergencia de este tipo de problemas, se agendó una reunión con el profesor Cristián Escauriaza, del departamento de ingeniería hidráulica y ambiental, en calidad de experto en el área. En esta reunión el profesor confirmó la validez de la hipótesis planteada para la confección del criterio de convergencia.

5. Marco Teórico

La situación que estudiamos es la de la mezcla de agua y un contaminante, cada uno con sus propiedades físicas como la densidad y viscosidad. Dada una región en el que se tienen ciertas condiciones acerca del campo de velocidad del fluido y la inyección de contaminante, queremos predecir cómo evoluciona cada líquido mientras se mezclan entre sí. Dada una región infinitesimal se calcula la concentración de contaminante como la cantidad de volumen de contaminante dividido por el volumen total de la región.

Nuestro objetivo es estudiar la pluma del contaminante, la región que cumple que la concentración del contaminante es mayor a cierta cota. Esta pluma va a cambiar mediante pasa el tiempo, hasta que se estabiliza y deja de evolucionar significativamente. Para modelar esta situación matemáticamente, usaremos el esquema de Huang et al., 2019. En particular, acoplamos las ecuaciones de Navier Stokes y una de conservación de fases. Resolvemos para $u,\ p,\ f$

$$\partial_t \rho + \nabla \cdot (\rho u) = 0$$

$$\partial_t (\rho u) + \nabla \cdot (\rho u u^{\mathsf{T}}) = -\nabla p + \nabla \cdot (\mu \nabla u) + \rho g$$

$$\partial_t (\rho f) + \nabla \cdot (\rho f u) = \nabla \cdot (\rho D \nabla f)$$

Donde ρ es la densidad del fluido, p la presión, u la velocidad, μ la viscosidad, g la aceleración gravitatoria, f la proporción de masa local de agua y el total, y D es el coeficiente de difusión de masa.

Se tiene que la proporción de contaminante es $1-f$, y, por lo tanto, $\mu = f\mu_1 + (1-f)\mu_2$, con μ_1 la viscosidad del agua y μ_2 la del contaminante. Además, $\rho = \left(\frac{f}{\rho_1} + \frac{(1-f)}{\rho_2}\right)^{-1}$, con ρ_1 la densidad del agua y ρ_2 la del contaminante. De manera que el campo de concentración de agua es $C_1 = \frac{f\rho}{\rho_1}$ y la del contaminante es $C_2 = \frac{(1-f)\rho}{\rho_2}$.

Para definir la pluma de contaminante, fijaremos una cota $\alpha \in [0,1]$. De manera que la pluma es la región del espacio que cumple que $C_2 \geq \alpha$.

6. Marco Computacional

6.1. El *Solver*

El solucionador utilizado actualmente es *interMixingFoam*, este contempla la aplicación conjunta (llamada PIMPLE) de los algoritmos Semi-*Implicit Method for Pressure-Linked Equations* (SIMPLE) y *Pressure-Implicit Split-Operator* (PISO) (Figura 5). Este está

diseñado específicamente para simulaciones del comportamiento de un sistema con tres fluidos incompresibles e isotérmicos, dos de los cuales son miscibles, e inmiscibles a un tercero, el cual usualmente es el aire en una mezcla abierta. Este puede adaptarse a flujos laminares y turbulentos, y fluidos newtonianos y no newtonianos, a través del enfoque de volumen de fluido (VoF). El enfoque principal de este solver está dirigido a resolver problemas donde es relevante conocer con precisión el comportamiento de la interfaz entre dos fluidos. Se nota que en el caso que no exista un efecto significativo del tercer fluido inmiscible, es posible anular su efecto, como es el caso cubierto por el solucionador twoLiquidMixing*Foam*, cuyas capacidades son ideales para analizar la mezcla de dos fluidos en tanques abiertos y se utiliza comúnmente para analizar la mezcla de dos fases, omitiendo la presencia de una tercera fase inmiscible.

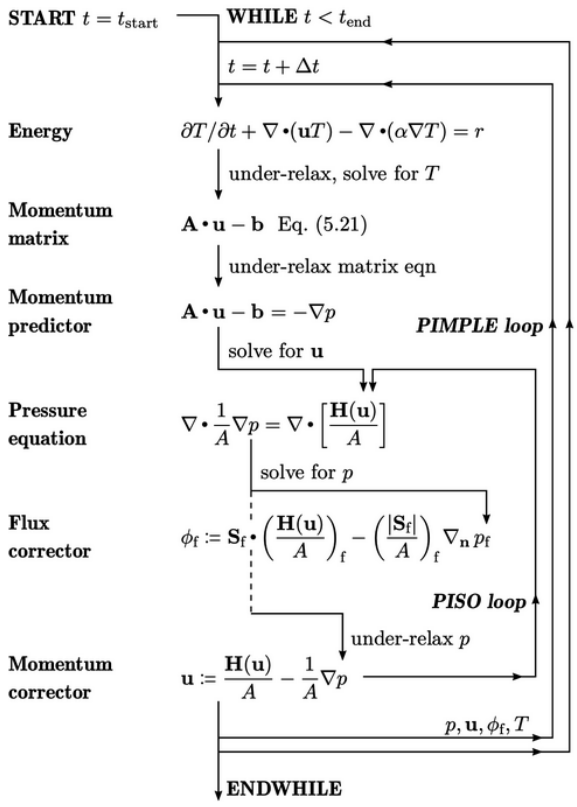

Figura 5: Flujo lógico del algoritmo PIMPLE. Obtenido de Greenshields & Weller, 2022.

Se nota dentro del método aplicado que las simulaciones realizadas son pseudo-transientes, lo que significa que relaja el problema para converger más rápido, sin embargo, es necesario que el problema al cual es aplicado debe ser sobre estado estacionario.

Evaluando los beneficios de *interMixingFoam*, se encuentra que su implementación es sencilla para una simulación y posee herramientas, actualmente no implementadas, que son deseables para el caso presente. Entre estas se encuentran la incorporación de turbulencias, un desarrollo de mallado dinámico, y trabajo con múltiples puntos de descarga de efluentes.

Sin embargo, este mismo solucionador restringe las aplicaciones del sistema dado que sigue una estructura definida que requiere una adaptación al código mismo para poder aplicarse sobre otro tipo de problemas. Así mismo, el solver está diseñado para operar en el problema general, de modo que, al igual que la mayoría de los solucionadores estándar de *OpenFOAM*, no implementa criterios de convergencia directos, lo cual es necesario en el contexto actual dado que se busca el estado estable en un tiempo finito.

6.2. Versiones de OpenFOAM

La implementación original del algoritmo se realiza en la versión 8 del código abierto, curada por OpenFOAM Foundation Inc., lanzada el año 2020.

Se plantea transferir la implementación a la versión actual 2406, curada por OpenCFD Ltd., lanzada en junio del año 2024. Esto dado la comparación en la Tabla 2,

Versión	Ventajas	Desventajas
v8.0	• Versión actual utilizada por HOLON	• Poco control de convergencia • Bajo número de *utilities* • Menor compatibilidad con código y librerías externas
v2406	• Funciones como *runtimeControl* • *topoSet* mejorado • Ampliamente utilizado por desarrolladores	• Necesidad de migrar el código • Validación de los resultados

Tabla 2: Comparación entre *OpenFOAM* versión v8.0 y v2406.

7. Optimización del Algoritmo

En la implementación …

La migración a la versión 2406 de OpenFOAM nos proporciona una versión optimizada y funcional de forma nativa.

7.1. Número de Courant

El número de Courant es un coeficiente adimensional utilizado en CFD para medir el cuociente entre la distancia que recorre una parte del fluido en el delta de tiempo simulado y el tamaño de la celda de la malla. Se utiliza en simulaciones de CFD establecer como cota superior 1, dado que un valor mayor a este representaría que el

fluido está atravesando más de una celda en el delta de tiempo, comportamiento que puede llevar a simulaciones incorrectas o resultados inestables. De esta forma, definir una mayor cota nos da una menor cantidad de pasos de ejecución del *solver*, así el número de Courant es una condición de convergencia, pero muy limitada en función del intervalo de tiempo definido. Se nota que el número de Courant está asociado a las condiciones convergencia de Courant–Friedrichs–Lewy (CFL) que también son utilizadas para definir requisitos de estabilidad, aunque no son suficientes.

A nivel del *solver*, este calcula el intervalo de tiempo máximo de su iteración actual tal que el número de Courant de cada celda sea menor o igual a la cota máxima dada. Iteramos la simulación con valores menores a 1 y no se obtuvo un cambio significativo en el tiempo de cómputo y los resultados. Concluimos que el número de Courant actual es óptimo dado el contexto del problema presente.

7.2. Turbulencia

La turbulencia es un aspecto importante a la hora de analizar un flujo de fluido, sin embargo, en nuestra reunión con el profesor Cristian Escauriaza, se nos afirmó que en la escala a la que se está trabajando, el efecto de la implementación de la turbulencia no presenta mayores diferencias en la convergencia y/o tiempos de cálculo. Es decir, la turbulencia no es significativa en el presente contexto.

7.3. *Custom solver*

Se analizó la posibilidad de crear un método propio y un *solver* que resuelva directamente el estado estable del problema sin tener que simular las iteraciones intermedias, sin embargo, se optó por no seguir esta línea de trabajo dado que el método actual se adapta de manera adecuada a la naturaleza del problema, y por ser nativa de *OpenFOAM*, es más fácil de implementar.

7.4. Implementación de tolerancia y relajación

La implementación nativa de los *solvers* contempla los siguientes parámetros de tolerancia sobre las soluciones que permiten abaratar parte del costo computacional en cada iteración sobre cada campo:

- *tolerance*: el residuo es menor que el parámetro.
- *relTol*: la relación entre el residuo inicial y actual es menor que el parámetro.
- *maxIter*: se alcanza un máximo de iteraciones.

De la misma manera, se agrega en el archivo *fvSolution* los factores de relajación en la función *relaxationFactors*, a aplicar sobre los campos y ecuaciones seleccionadas,

De forma más específica para el algoritmo PIMPLE, se implementa la función *residualControl* que busca implementar una tolerancia para las iteraciones a realizar por los correctores, tales como *nCorrectors*, *nNonOrthogonalCorrectors*, *nOuterCorrectors*.

Se iteró varias veces sobre los valores de los parámetros (Holzmann, 2019), sin embargo, no se logró obtener soluciones con costos computacional y temporal significativos.

8. Convergencia

Uno de nuestros principales objetivos consistía en crear e implementar un criterio de convergencia que permitiera determinar el momento en que la simulación alcanza un estado es estabilidad. Esto con el fin de evitar que el programa realice iteraciones innecesarias, además de poder asegurar la estabilidad del sistema con parámetros medibles.

OpenFOAM no cuenta con implementaciones nativas de criterios de convergencia de las simulaciones en sí, sin embargo, es posible detener la ejecución arbitrariamente mediante la manipulación del archivo *controlDict*. Así, buscamos verificar la convergencia comparando el estado del modelo actual con los pasos anteriores mediante la diferencia de los momentos de las últimas iteraciones del sistema.

Nuestra reunión con el profesor Escauriaza nos permitió reafirmar la hipótesis planteada, además nos aseguró que es suficiente comparar los primeros dos momentos del sistema, el promedio y la varianza. Estos momentos se pueden obtener a través de la función *fieldAverage,* donde se calcula el promedio (*mean*) y la varianza (*prime2Mean*) en función de las iteraciones temporales (*base: time*) de cada campo vectorial o escalar, dependiendo del caso. Para obtener mejor precisión al comparar los momentos, se calculan estos sobre algunas zonas delimitadas en el archivo *topoSetDict*, igualmente ubicado en la carpeta *system*.

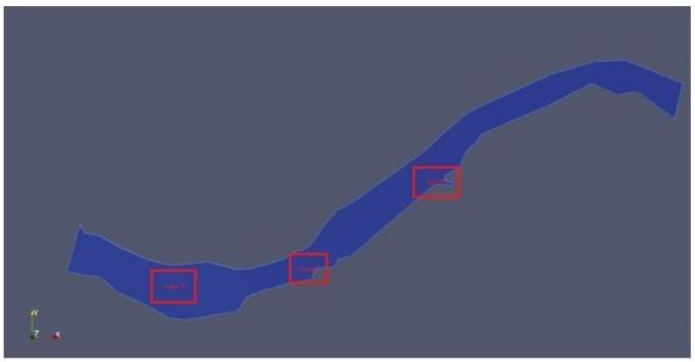

Figura 6: Zonas delimitadas en la Modelación 1, 02 Vaciante. De derecha a izquierda: zona 1, zona 2 y zona 3.

```
// ************************************************************************* //

functions //funciones en controlDict
{
       fieldAverage1 //se calculan los momentos para cada punto del dominio de un campo con
respecto al tiempo
       {
type                 fieldAverage; //tipo de formato
libs                 (“libfieldFunctionObjects.so”);
fields
(
                     alpha.other //campo sobre el cual calcular los momentos
                     {
                            mean        yes; //promedio
                            prime2Mean  yes; //varianza
                            base        time;
                     }
);
log                  true;
executeControl       writeTime;
writeControl         writeTime;
       }
...
};
```

Código 1: *fieldAverage* para el campo completo en Modelación 1, 01 Llenante.

```
// ************************************************************************* //

functions //funciones en controlDict
{
...
       fieldAverageCelda1 //se calculan los momentos para cada punto del dominio de un campo
con respecto al tiempo
       {
type                 volFieldValue; //tipo de formato
libs                 (“libfieldFunctionObjects.so”);
log                  true;
writeControl         writeTime;
writeFields          false;
regionType           cellZone;
name                 zona1;
operation            volAverage;
executeControl       writeTime;
writeToFile          true;

fields
(
                     alpha.other //campo sobre el cual calcular los momentos
);
       }
...
};
```

Código 2: *fieldAverage* en la zona 1 del campo en Modelación 1, 01 Llenante.

Así, se implementa en el archivo *controlDict* la función *runTimeControl*, donde se instancian condiciones para detener la ejecución completa de la simulación. Estas condiciones comparan los momentos de cada campo, cuyos valores en cada iteración

se guardan en el archivo *functionObjectProperties*. Se nota que las funciones encargadas de calcular los momentos y almacenar dichos valores también se definen dentro de *controlDict*.

```
// ************************************************************************* //

functions //funciones en controlDict
{
...    //aquí van las funciones de fieldAverage para cada campo
runTimeControl1
{
type            runTimeControl;
libs            ("libutilityFunctionObjects.so");
       timeStart       200;
       conditions
{
                           condition0
                           {
                                  type            equationInitialResidual;
                                  fields          (U);
                                  value           0.7;
                                  mode            maximum;
                           }
                           condition1
                           {
                                  type            equationMaxIter;
                                  fields          (U);
                                  threshold       100;
                           }
                           condition2
                           {
                                  type            average;
                                  functionObject  fieldAverageCelda1;
                                  fields          (volAverage(zona1,alpha.other));
                                  tolerance       1e-3;
                                  window          5;
windowType      exact;
                           }
                           condition3
                           {
                                  type            average;
                                  functionObject  fieldAverageCelda2;
                                  fields          (volAverage(zona2,alpha.other));
                                  tolerance       1e-3;
                                  window          5;
windowType      exact;
                           }
                           condition4
                           {
                                  type            equationInitialResidual;
                                  fields          (U);
                                  value           1e-04;
                                  mode            minimum;
                           }
}
}
};
```

Código 3: Condiciones en *runTimeControl* en Modelación 1, 01 Llenante.

Además, para asegurar la precisión de la convergencia, analizamos los siguientes elementos:

- *Residual values*: cuantificar errores, esto se controla a través de las tolerancias definidas.
- *Solution imbalances*: asegurar conservación de masa, *momentum*, energía, entre otros, a través de las iteraciones. Esto se realiza mediante el uso del método de volúmenes finitos.
- Verificar que las cantidades de interés converjan en el rango esperado. Esto lo confirmamos mediante la comparación de varias simulaciones.

El error numérico se puede controlar mediante una elección de malla del problema lo suficientemente fina tal que el resultado final sea independiente de este, no obstante, fue acordado con la contraparte, HOLON SpA., que no manipularemos la malla actual.

9. Resultados

Se obtiene exitosamente la implementación del criterio de convergencia donde se puede apreciar en las simulaciones, la mínima cantidad de iteraciones hasta lograr el estado estacionario. Esto se puede ver en los siguientes gráficos, en los cuales se eligen zonas de muestreo en base a regiones en que se sabe que pasará contaminante.

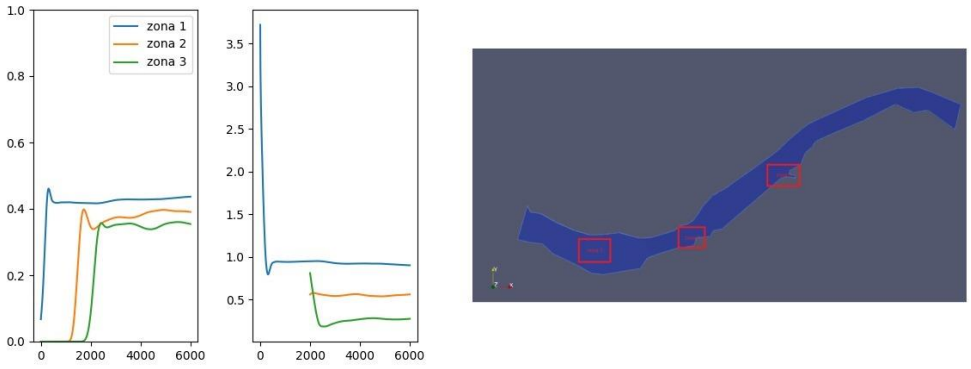

Figura 7: Convergencia del promedio (izquierda) y coeficiente de variación (medio) de la variable alpha.other en función de cantidad de iteraciones en las zonas de muestreo escogidas (derecha). Modelación 1, 02 Vaciante.

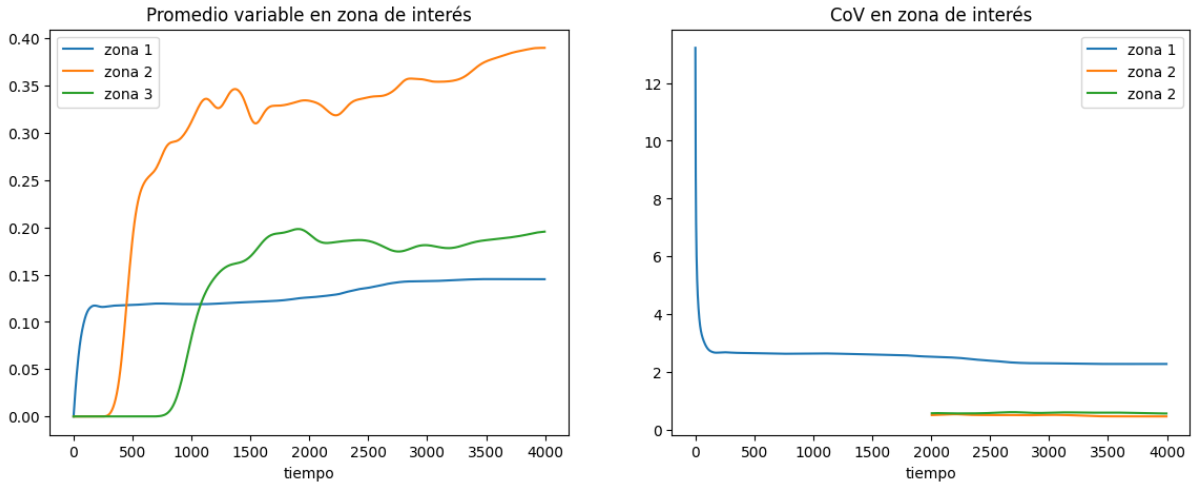

Figura 8: Convergencia del promedio y coeficiente de variación de la variable alpha.other en función de cantidad de iteraciones. Modelación 1, 01 Llenante.

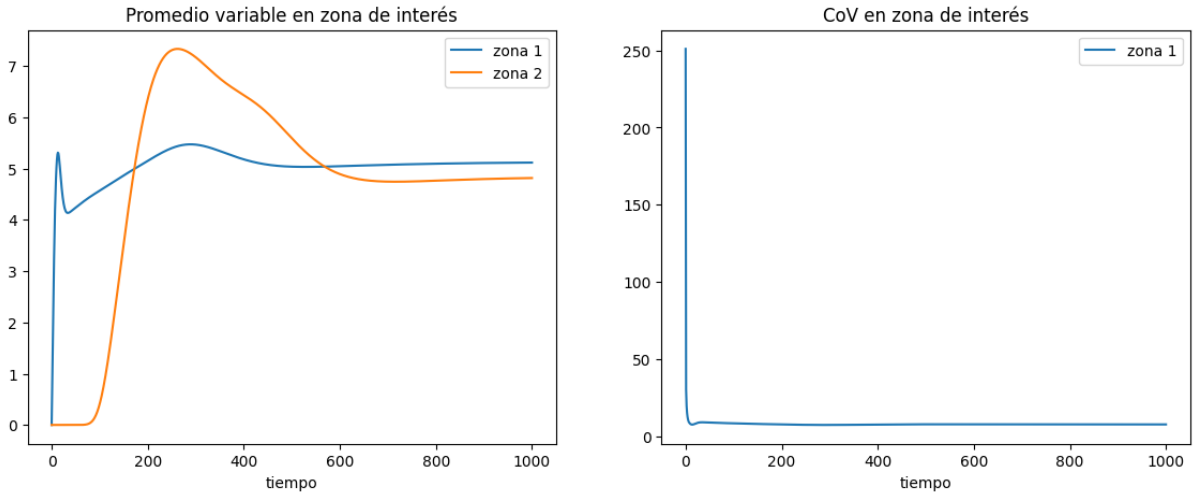

Figura 9: Convergencia del promedio y coeficiente de variación de la variable ColFec en función de cantidad de iteraciones. Modelación 2, 01 Sicigia Vaciante.

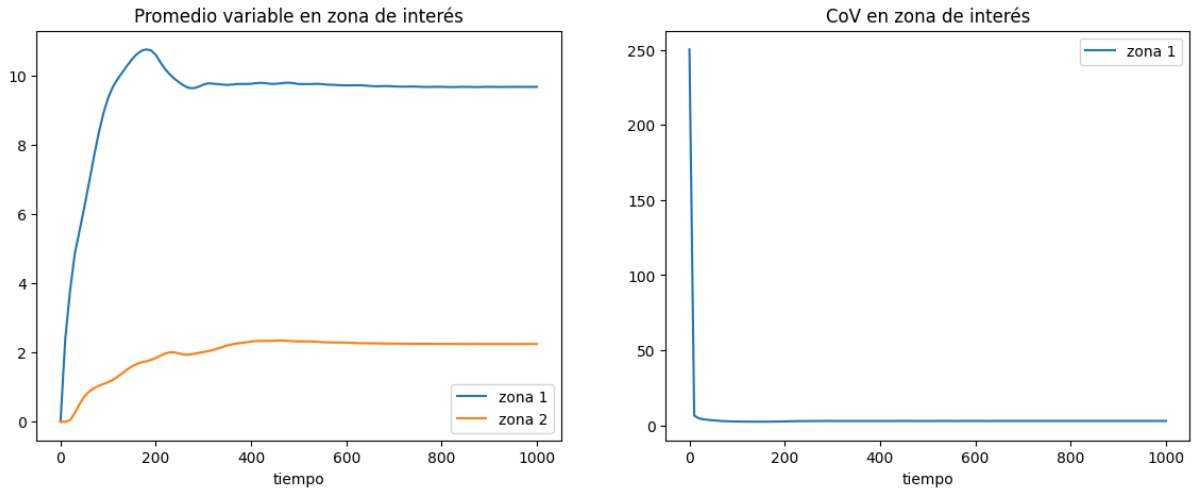

Figura 10: Convergencia del promedio y coeficiente de variación de la variable ColFec en función de cantidad de iteraciones. Modelación 2, 02 Sicigia Llenante.

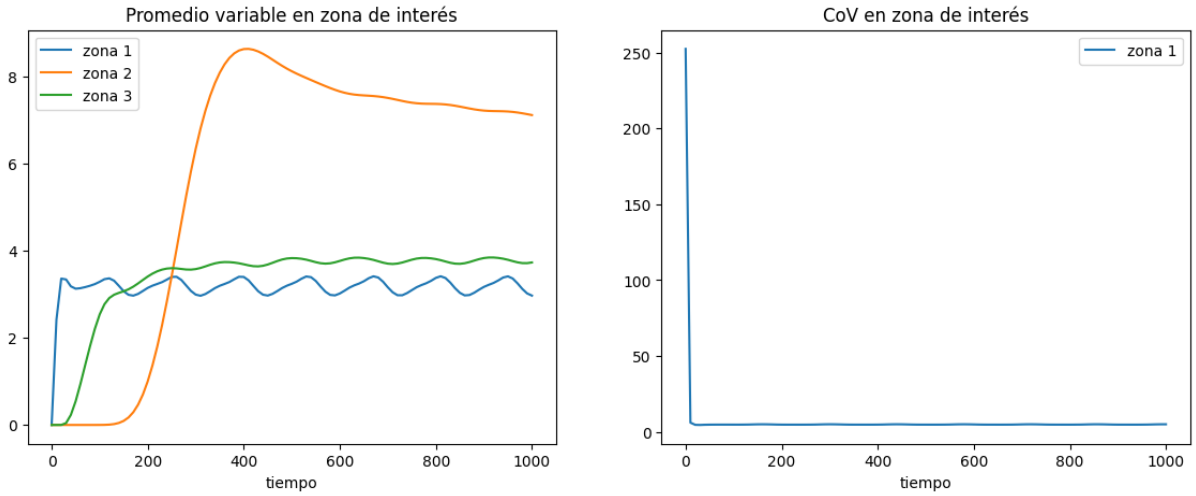

Figura 11: Convergencia del promedio y coeficiente de variación la variable ColFec en función de cantidad de iteraciones. Modelación 2, 03 Cuadratura Vaciante.

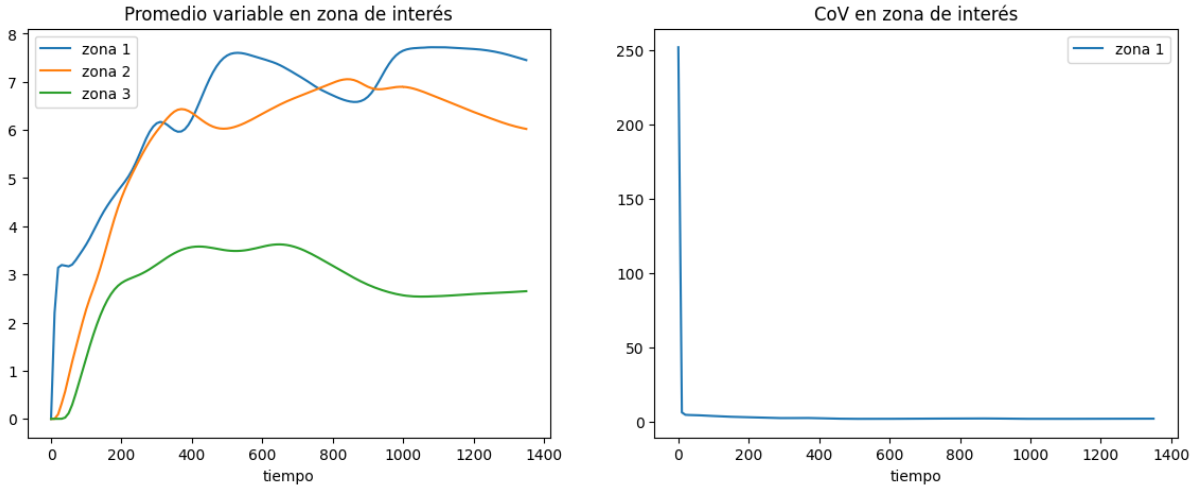

Figura 12: Convergencia del promedio y coeficiente de variación de la variable ColFec en función de cantidad de iteraciones. Modelación 2, 04 Cuadratura Llenante.

Se puede observar la cantidad de iteraciones realizadas resumidas en la Tabla 2, donde hay una clara diferencia del orden de 10E3. Del mismo modo, esto se ve reflejado en los tiempos de cómputo de las simulaciones.

Modelo	**Iteraciones Originales**	**Iteraciones Convergentes**
1: Llenante	4000	4000
1: Vaciante	4000	6000
2: Sicigia Vaciante	4000	<1000
2: Sicigia Llenante	4000	<1000
2: Cuadratura Vaciante	4000	~1000
2: Cuadratura Llenante	4000	~1000

Tabla 2: Comparación de iteraciones realizadas.

10. Conclusión

De los objetivos principales planteados para este proyecto, no se logró reducir los costos computacionales de cada iteración del modelo. Dado que este tipo de optimización necesita más tiempo y/o herramientas de las que tenemos a disposición para la realización de este trabajo.

Sin embargo, la implementación del criterio de convergencia basado en la comparación de momentos entrega a la contraparte un parámetro medible para determinar que el sistema ha alcanzado la estabilidad. De esta manera, cada caso a simular tiene el potencial de reducir significativamente el número de iteraciones que le toma alcanzar la estabilidad. Por otra parte, en caso contrario, este mismo criterio de convergencia le permite a la contraparte asegurar la estabilidad de la pluma y no cometer el error de detener la simulación de manera precoz.

Finalmente, en cuanto a la complejización del modelo, se determinó que a la escala en que se está trabajando efectos como la turbulencia no debiesen generar diferencias importantes tanto en los costos computacionales como en la precisión de las soluciones.

11. Referencias Bibliográficas

Greenshields, C. (2024). *OpenFOAM v12 User Guide*. The OpenFOAM Foundation. https://doc.cfd.direct/openfoam/user-guide-v12

Greenshields, C., Weller, H. (2022). *Notes on Computational Fluid Dynamics: General Principles*. CFD Direct Ltd. https://doc.cfd.direct/notes/cfd-general-principles

HOLON SpA. (2022). *Estudio de dinámica costera y modelación de la pluma de descarga. Proyecto "Ampliación de la planta de congelados de Choritos", sector Rauco, Región de Los Lagos. Camanchaca Cultivos Sur S.A.*

HOLON SpA. (2024). *Informe Modelación pluma de descarga Proyecto "Piscicultura Río del Este". Salmones Camanchaca, Región de Los Lagos*.

Holzmann, T. (2019). *Mathematics, Numerics, Derivations and OpenFOAM: The Basics for Numerical Simulations*. https://holzmann-cfd.com

IdealSimulations (2020). *Courant Number*. https://www.idealsimulations.com/resources/courant-number-cfd/

Joseph, D.D. (2010). *Fluid Dynamics of Mixtures of Incompressible Miscible Liquids*. In: Fitzgibbon, W., Kuznetsov, Y., Neittaanmäki, P., Périaux, J., Pironneau, O. (eds) Applied and Numerical Partial Differential Equations. Computational Methods in Applied Sciences, vol 15. Springer, Dordrecht. https://doi.org/10.1007/978-90-481-3239-3_10

Joseph, D.D., Renardy, Y.Y. (1993). *Fluid Dynamics of Two Miscible Liquids with Diffusion and Gradient Stresses*. In: Fundamentals of Two-Fluid Dynamics. Interdisciplinary Applied Mathematics, vol 4. Springer, New York, NY. https://doi.org/10.1007/978-1-4615-7061-5_6

Krishna, R. (1976). Steady-state Mass Transport in Multicomponent Liquid Mixtures. *Letters in Heat and Mass Transfer (3)*2, p.153-162. https://doi.org/10.1016/0094-4548(76)90067-9

Huang, F., Wang, D., Li, Z., Gao, Z., Derksen, J.J. (2019). Mixing process of two miscible fluids in a lid-driven cavity, *Chemical Engineering Journal (362)*, p.229-242. https://doi.org/10.1016/j.cej.2019.01.024

OpenCFD Ltd. (2024). OpenFoam: The Open Source CFD Toolbox. User Guide (v.2406).

OpenCFD Ltd. (2024). OpenFoam: The Open Source CFD Toolbox. Programmer's Guide (v.2406).