



Universidad Tecnológica Nacional
Facultad Regional Santa Fe

Informe de Proyecto Final de Carrera
Ingeniería en Sistemas de Información

*Corrección de inestabilidades numéricas en
simulaciones de fenómenos de transporte reactivos
discontinuos*

Alumnos

García Aguirre, Octavio
LU: 21613

Harispe, David Gabriel
LU: 21502

Director

Dr. Pablo A. Kler

Co-Director

Ing. Gabriel S. Gerlero

Año: 2020

ÍNDICE

I. Introducción	5
1. Objetivos del proyecto	5
1.1. Objetivos generales	6
1.2. Objetivos específicos	6
2. Estructura del informe	6
II. Marco Teórico	8
1. Sistemas reactivos-difusivos-advectivos	8
2. Dinámica de fluidos computacional	9
3. Método de volúmenes finitos	9
4. Patrones de Liesegang	10
4.1. Modelos de sol-coagulación	11
5. Leyes de los patrones de Liesegang	12
5.1. Ley de Espaciado	12
5.2. Ley de Ancho	13
5.3. Ley de Tiempo	13
5.4. Ley de Matalon-Packter	13
6. Terminología	14
III. Herramientas y tecnologías utilizadas	16
1. OpenFOAM	16
2. porousMultiphaseFoam y porousScalarTransportFoam	16
3. ParaView	17
4. C++	17
5. Gmsh	17
6. Repositorios GitHub y GitLab	17
7. Herramientas Google Workspace (Drive, Docs, Sheets, Hangouts)	18
8. Python	18
9. Ubuntu y WSL	18
10. glogg	19
11. Editores de código Notepad++ y Visual Studio Code	19
12. Bash	19
IV. Metodologías utilizadas	20
1. Metodología para investigación	20
2. Metodología para desarrollo	20
V. Desarrollo del proyecto	23
1. Etapa de estudio e investigación del dominio del problema y las herramientas para abordarlo	25
Sprint 1	25
2. Etapa de desarrollo, investigación y experimentación en casos 1D	27

Sprint 2	27
Sprint 3	29
Sprint 4	32
3. Etapa de desarrollo, investigación y experimentación en casos 2D	35
Sprint 5	35
Sprint 6	36
Sprint 7	38
4. Etapa de pruebas y experimentación con casos 2D de la bibliografía	40
Sprint 8	40
5. Generación de la documentación	42
6. Riesgos no específicos a una iteración.	43
VI. Arquitectura de la aplicación	45
1. Estructura y archivos de casos simulados	45
2. Parámetros de una simulación	46
3. Estructura general de la aplicación	48
4. Diagrama de clases	49
5. Precálculo de vecinos de celdas	51
6. Graduación de celdas - Expansión de celdas	53
7. Alternativa a función escalón Heaviside	54
8. Algoritmo de actualización de paso de tiempo de la simulación	55
8.1. Cálculo de tiempo máximo τ por difusión en celdas	59
8.1.1. Derivación expresión 1D	59
8.1.2. Generalización de la expresión a 2D	60
VII. Funcionamiento de la aplicación	62
1. Dominios Unidimensionales - 1D	62
1.1. Configuración de los casos de simulación	62
1.2. Estructura del dominio	63
1.3. Funcionamiento de la implementación del modelo inicial	64
1.4. Convergencia temporal	65
1.5. Convergencia de malla	68
2. Dominios Bidimensionales - 2D	72
2.1. Pruebas Híbridas 1D/2D	72
2.2. Configuración de los casos de simulación	74
2.3. Estructura del dominio - Pruebas de convergencia temporal	74
2.4. Funcionamiento del nuevo modelo generalizado a dos dimensiones	75
2.5. Convergencia temporal	76
2.6. Convergencia de malla	77
2.7. Evaluación de independencia de malla	78

2.7.1.	Malla Circular	78
2.7.2.	Malla Combinada	79
2.7.3.	Malla V-circular	81
2.7.4.	Malla V	82
2.7.5.	Mallas Gmsh	83
2.7.6.	Conclusión de convergencia de malla	86
3.	Reproducción de resultados de bibliografía específica	86
3.1.	Determinación de parámetros - Configuración casos	86
3.2.	Estructura del dominio	88
3.3.	Resultados y comparativa	88
4.	Validación de leyes en resultados	89
VIII.	Conclusiones	91
1.	Trabajos futuros	91
	BIBLIOGRAFÍA	93
	Anexos	96
	Anexo A - Repositorio con código fuente	96
	Anexo B - Plan de gestión de riesgos	97
	Anexo C - Uso de aplicativo Python	101
	Anexo D - Efectos de baja precisión en la formación de bandas	104
	Anexo E - Archivos de un caso	105
	Anexo F - Instrucciones visualización de casos en ParaView	115

I. Introducción

La generación de nuevos materiales funcionales diseñados desde la micro y nano escala es un área de intensa actividad actual. En este marco, son de especial interés los sistemas reactivos que producen precipitaciones complejas con patrones autoorganizados. Si bien se han realizado grandes progresos en los últimos años, la dinámica de dichos procesos no ha sido bien descrita aún. Para ello es necesario comprender cómo los fenómenos de transporte afectan la síntesis en microsistemas. Este conocimiento es crucial para realizar un diseño racional de las estructuras y regular los parámetros de operación durante la síntesis.

La simulación computacional, es una herramienta muy útil para acelerar el estudio y la predicción de los sistemas mencionados. Según Pristker [1], la simulación computacional, se puede definir como el proceso de diseñar un modelo matemático-lógico de un sistema del mundo real y experimentar con el modelo en una computadora. Esto nos permite entender el comportamiento del sistema y/o evaluar estrategias para la operación de éste. Por otro lado, simular sistemas se ha impuesto como una herramienta valiosa particularmente al momento de estudiar procesos que son dependientes de recursos dado que provee una forma rápida y económica para experimentar con diferentes alternativas y enfoques.

No obstante, la implementación numérica de modelos que describen estos sistemas reactivos es muy desafiante, dado que en general se trata de sistemas de ecuaciones no-lineales acopladas en dominios multi-escala.

Dentro de los sistemas reactivos, existen los sistemas reactivos discontinuos como los que dan origen a los denominados patrones de Liesegang [2], [3]. Estos se forman cuando dos compuestos A y B reaccionan, y a partir de determinada concentración crítica, el producto AB precipita en regiones definidas del espacio, formando bandas periódicas. Al presente, existen modelos computacionales 1D que describen este proceso mediante el acople de las ecuaciones de reacción-difusión de cada compuesto, con un término auto catalítico de cristalización y/o precipitación. Sin embargo, actualmente, no se conocen otros modelos numéricos 2D o 3D capaces de simular adecuadamente el fenómeno de generación de patrones de Liesegang.

1. Objetivos del proyecto

Debido a la situación descrita previamente y en el marco del PID 4475 “Modelado y simulación de fenómenos de transporte en la micro y nano escala”, se desarrolló un modelo difusivo-reactivo-advectivo 3D utilizando la plataforma de simulación en volúmenes finitos OpenFOAM [4], para predecir la formación de patrones mediante el encuentro de dos frentes de infiltración en un sustrato poroso. Este modelo tiene el defecto de generar patrones indeseados en los resultados cuando se intenta simular estos procesos en dominios bidimensionales (2D). Es de ahí que surge este proyecto, que pretende realizar en primera instancia un estudio e investigación del fenómeno que se requiere simular, sus causas y luego desarrollar un algoritmo capaz de generar una representación adecuada (y fisicoquímicamente válida) de los patrones en cuestión en un dominio bidimensional.

Originalmente, la intención con este proyecto era intentar corregir estos “patrones indeseados” (patrones de checkerboard) que se originan a partir del modelo difusivo-reactivo-advectivo 3D elaborado previamente. Esto, implicaba comenzar el trabajo a partir de esta base creando algoritmos adicionales que, en tiempo de ejecución, fueran capaces de detectar y corregir los errores en la simulación para permitir la obtención de resultados correctos.

Luego de una primera fase de estudio del problema, se decidió que el enfoque correcto sería el de desarrollar un nuevo modelo difusivo-reactivo-advectivo. Para esto, se comenzaría trabajando con dominios unidimensionales (1D), y se aseguraría la obtención de resultados válidos en cada etapa atravesada, hasta alcanzar resultados satisfactorios en dominios bidimensionales (2D). Esto, se haría a partir de un conjunto de herramientas de código abierto para simulación de flujo multifásico¹ en medios porosos, desarrollado para OpenFOAM.

¹ Fluido multifásico: mezcla que consta de más de dos fases, que pueden ser líquidas o gaseosas, inmiscibles entre sí.

Con el nuevo enfoque, existía la posibilidad de que los patrones indeseados surgieran nuevamente al trabajar con dominios bidimensionales, lo que implicaría la necesidad de, además, detectarlos y corregirlos como se planeó originalmente, pero al mismo tiempo, permitía disponer de un modelo elaborado desde los cimientos con la idea de simulaciones en 2D en mente, otorgando posiblemente mayor robustez.

Debido a este cambio de enfoque, los objetivos originales fueron ligeramente modificados para que fuesen más amplios, enfocados más en la robustez del algoritmo ante diferentes condiciones de simulación y no específicamente en los patrones indeseados (de checkerboard), contando con el aval del director y codirector propuesto (quienes toman el rol de ser los *product owner*).

1.1. Objetivos generales

Con el proyecto se propuso el siguiente objetivo general: desarrollar e implementar un algoritmo robusto en la plataforma OpenFOAM, que permitía simular correctamente la formación de patrones de Liesegang en dominios 2D.

1.2. Objetivos específicos

En el marco del proyecto a presentar, se pretendió alcanzar los siguientes objetivos específicos:

- a. Diseñar e implementar un algoritmo robusto capaz de recrear los casos 1D de los patrones de Liesegang observados en la bibliografía de referencia, a modo de validación del código, dentro de la plataforma OpenFOAM.
- b. Diseñar e implementar un algoritmo robusto capaz de recrear los casos experimentales 2D de los patrones de Liesegang observados en la bibliografía de referencia, dentro de la plataforma OpenFOAM.
- c. Testear los algoritmos obtenidos en casos 2D más complejos, donde el comportamiento del algoritmo y la simulación, en general, sea estudiado y puesto a prueba.
- d. Publicar los resultados en una revista o congreso científico de alto impacto.

2. Estructura del informe

El actual informe está compuesto de 8 capítulos. El capítulo I es el presente, cuyo objetivo es dar una introducción a la temática de trabajo de este proyecto. Por otro lado, se presentan los objetivos del informe, tanto generales como específicos.

En el capítulo II se realiza una introducción a los conceptos teóricos que fundamentan los desarrollos, investigaciones y experimentos realizados a lo largo del proyecto. Este capítulo puede ser de gran interés para el lector, debido a que establece la base necesaria para comprender el resto del informe. Entre otras cosas, se introducirán conceptos básicos sobre dinámica de fluidos computacional, el método de volúmenes finitos y los patrones de Liesegang. Al final de este capítulo, también se explica la terminología específica utilizada en el informe (y el proyecto). Los términos ahí explicados son usados consistentemente a lo largo de los demás capítulos.

En el capítulo III, se hace una revisión de las herramientas utilizadas en las diversas actividades del proyecto. Se brinda una definición breve sobre cada herramienta y se explica por qué fue seleccionada y para qué se la utilizó.

Las metodologías seguidas en el proyecto se explican en el capítulo IV. Allí, se podrá encontrar una descripción de estas, tanto para las actividades de investigación, como para las de desarrollo.

La descripción de cómo fue la evolución del proyecto en cada una de sus etapas, analizando las tareas realizadas, el cumplimiento de cronogramas previstos, riesgos encontrados y las correspondientes respuestas se describen en detalle en el capítulo V.

En el capítulo VI se explica de forma detallada cómo se estructura internamente la aplicación desarrollada. Se describe cómo y con qué trabajan sus algoritmos, los conceptos teóricos que aplican y cómo se llegó a definirlos en su forma final.

El funcionamiento de la aplicación, se expone en el capítulo VII. En dicho capítulo, se hace un recorrido por las distintas clases de pruebas realizadas durante el desarrollo del proyecto. Se exploran las debilidades de modelos previos al desarrollado y como, el actual, es capaz de mejorar los resultados obtenidos de forma remarcable. Se hace un estudio de los resultados en todos los casos y se explica cómo se logró asegurar la validez de lo desarrollado. Adicionalmente, se muestra como el modelo y algoritmos elaborados son capaces de recrear (con un gran nivel de precisión) los resultados de experimentos de laboratorio reales (no simulados).

En el capítulo VIII, se presentan las conclusiones obtenidas y se incluyen posibles trabajos a futuro. Luego de esto se podrá encontrar la bibliografía citada en el informe y los anexos correspondientes.

II. Marco Teórico

En este capítulo se pretende que el lector sea capaz de familiarizarse con los fenómenos de transporte, en particular en sistemas reactivos-difusivos-advectivos. Por otro lado, se introducen las definiciones básicas del método de volúmenes finitos y la dinámica de fluidos computacional. Luego se procede a explicar la base teórica que rige la formación de los patrones de Liesegang y el modelo matemático que se adoptó para las simulaciones realizadas.

En la última sección de este capítulo, se introduce la terminología específica a ser usada a lo largo del informe.

1. Sistemas reactivos-difusivos-advectivos

Se puede definir a la reacción como un proceso en el que las sustancias reactivas participan en el proceso de reacción y se convierten en sustancia producto. La reacción da lugar a la variación de la concentración de dichas sustancias a causa de las interacciones con otras sustancias (de ahora en adelante especies) y consigo misma.

La velocidad de una reacción química se puede definir como la cantidad de reacción que ocurre en unidad de tiempo. La velocidad de una reacción se mide eligiendo ciertas propiedades de la reacción que indicarán el estado de los reactivos y cuya magnitud se puede observar con el tiempo sin perturbar la reacción.

Por otro lado, la difusión puede ser definida como el movimiento de partículas de una región de alta concentración a una región de baja concentración en dirección opuesta a un gradiente de concentración. Por tanto, la difusión es el proceso en el que un gas o una sustancia, que se encuentra en un medio continuo, se expande para llenar todo el volumen por igual minimizando su gradiente hasta encontrar su equilibrio termodinámico.

Finalmente, la advección, se define como el transporte de una sustancia, por ejemplo, en respuesta a un gradiente de presión o una fuerza de cuerpo (sobre el fluido). Las propiedades de esta sustancia son conservadas.

Es entonces, que se puede describir a los sistemas de reacción-difusión-advección como sistemas en los que los constituyentes se transforman localmente entre sí por reacciones químicas y se transportan en el espacio por difusión y/o advección. Estos sistemas están presentes de forma natural no solo en la ciencia química, sino que también en las ciencias ambientales y de la vida [5].

Los sistemas de reacción-difusión-advección pueden ser descritos por modelos matemáticos formados por ecuaciones en derivadas parciales (EDP). Las EDP se pueden describir como ecuaciones diferenciales donde las incógnitas son funciones de diversas variables independientes, pero, además, en estas ecuaciones, las derivadas de estas funciones también están presentes. Las EDP, en el contexto de este proyecto, representan la evolución de una sustancia química particular (sus concentraciones) en un medio descrito por coordenadas espaciales.

Como se mencionó antes, en estos sistemas, el transporte se origina por una fuerza física (advección) o química (difusión) y, en el proceso, sucede la reacción química con los otros componentes del medio, lo cual es representado por términos fuente o pérdida en las ecuaciones. La forma general de una ecuación de reacción-difusión-advección para una sustancia C es:

$$\frac{\partial C}{\partial t} = \nabla \cdot (Cu) + \nabla \cdot (D\nabla C) + R(C) \quad (2.1)$$

En esta ecuación, denominada ecuación de transporte, u es un campo vectorial (de velocidad) que determina una dirección de transporte preferencial, D es un coeficiente de difusión que puede depender de las coordenadas espaciales y R un término fuente de reacción. El operador ∇ se denomina nabra y puede denotar el

gradiente si se aplica a un campo escalar ² (como sucede con ∇C en la ecuación (2.1) o en la sección 8.1 del capítulo VI) o la divergencia si es aplicado sobre un campo vectorial (como sucede con $\nabla \cdot (Cu)$) ³.

El término $\frac{\partial C}{\partial t}$ representa la derivada parcial respecto del tiempo de la concentración de la sustancia C. Cuando hay reacciones entre diversas especies, existirá una ecuación para cada una de las especies en el sistema. Lo que plantea un sistema de ecuaciones en derivadas parciales.

Las ecuaciones como la (2.1) poseen una amplia variedad de aplicaciones, desde reactores industriales hasta el transporte de nutrientes y gases en seres vivos, pasando por muchas otras cosas [6].

2. Dinámica de fluidos computacional

La Dinámica de fluidos computacional (CFD) es una de las ramas de la mecánica de fluidos (rama de la física que estudia el movimiento de los fluidos y las fuerzas que lo provocan) que aplica métodos numéricos y algoritmos computacionales para resolver y analizar problemas sobre el flujo de estos.

Las computadoras son utilizadas para realizar los cálculos requeridos para simular el flujo del fluido y la interacción de este con superficies que se definen por medio de condiciones de borde.

Aun realizando simplificaciones, y con el uso de equipamiento computacional especializado, las simulaciones (dependiendo de los tamaños de los problemas) suelen requerir mucho tiempo para ser realizadas y los resultados son aproximaciones (con menor o mayor exactitud). La investigación ha permitido incorporar software que mejora la velocidad de cálculo y reduce los errores en los resultados.

Para implementar la dinámica de fluidos computacional existen diversos métodos que hacen uso de distintos enfoques, pero entre estos el que destaca y juega un rol especial es el método de volúmenes finitos.

En CFD las dos ecuaciones que se resuelven de manera preponderante son la ecuación de Navier-Stokes [7] (y sus variantes) y la ecuación de transporte, presentada en la sección anterior (ecuación 2.1). Ambas, son resultado de la aplicación del teorema de Reynolds [8] sobre el momento del fluido y sobre la densidad (conservación de masa), respectivamente.

3. Método de volúmenes finitos

El Método de Volúmenes Finitos (FVM) es una técnica numérica que permite transformar ecuaciones en derivadas parciales que representan leyes de conservación (de energía, masa, momento lineal) sobre volúmenes diferenciales (porciones del espacio muy pequeñas) en ecuaciones algebraicas discretas sobre volúmenes finitos (porciones del espacio bien definidas, usualmente denominadas celdas). Los desarrollos en este método han estado estrechamente relacionados con los avances en CFD.

Para aplicar este método se discretiza el dominio geométrico en volúmenes finitos (**Fig. 2-1**), es decir, se divide el dominio que naturalmente es continuo en pequeñas porciones que se consideran indivisibles (celdas), sobre las cuales se trabaja. Las ecuaciones en derivadas parciales se transforman en sistemas de ecuaciones algebraicas luego de integrarlas sobre las celdas. El sistema de ecuaciones transformadas luego se resuelve para calcular los valores de la variable dependiente, para cada elemento.

El FVM presenta algunas características que lo hacen el método preferido para la dinámica de fluidos computacional (CFD), como ser:

- El FVM es estrictamente conservativo debido a que el flujo que ingresa a un volumen dado es idéntico al que sale de los volúmenes adyacentes.
- Se puede formular en el espacio físico sobre mallas poligonales no estructuradas.

² Definido en la sección 6 de este capítulo.

³ El operador nabla puede expresar otros conceptos matemáticos dependiendo de la forma en que es aplicado.

- Simplifica la implementación de una variedad de condiciones de borde de una manera no invasiva. Las variables desconocidas se evalúan en los centroides de las celdas (no en sus caras de contorno), y las condiciones implementadas se consideran solo en el contorno.

Estas características hacen que FVM sea bastante adecuado para la simulación numérica de una variedad de aplicaciones que involucran flujo de fluidos y transferencia de calor y masa.

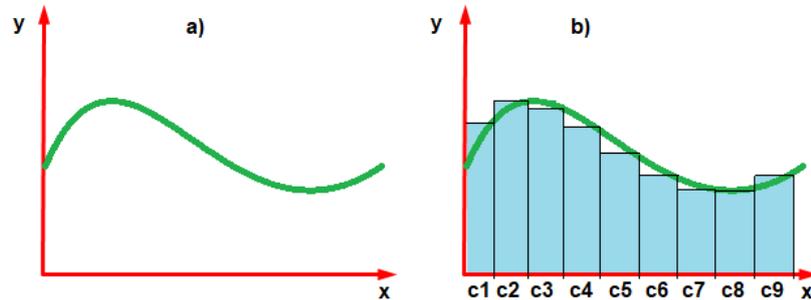


Fig. 2-1. a) Espacio continuo con una sola dimensión x donde la variable dependiente y se distribuye (puede ser la concentración de alguna especie, una velocidad, etc.). b) Discretización del espacio continuo, en donde, sobre cada celda, se tiene un valor asignado de la variable dependiente y .

4. Patrones de Liesegang

Los patrones o anillos de Liesegang (**Fig. 2-2 y 2-3**) ([2], [9], [10]) son un fenómeno observado en sistemas químicos que experimentan una reacción de precipitación (creación de un sólido insoluble a partir de una solución) en determinadas condiciones de concentración de una determinada especie.

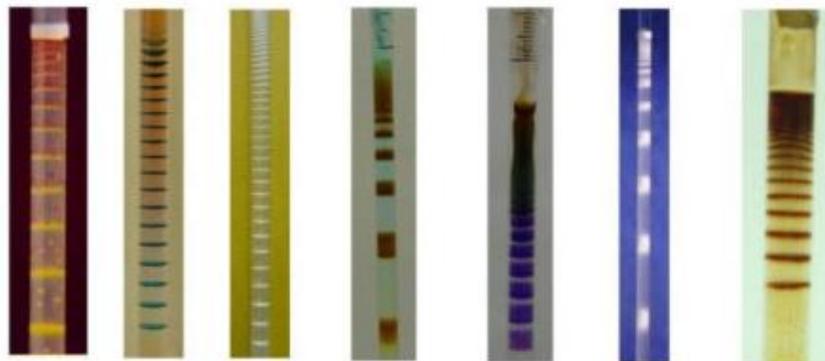


Fig. 2-2. Patrones de Liesegang en espacios aproximadamente unidimensionales (probetas). Imagen adaptada de http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1679-78252013000100006

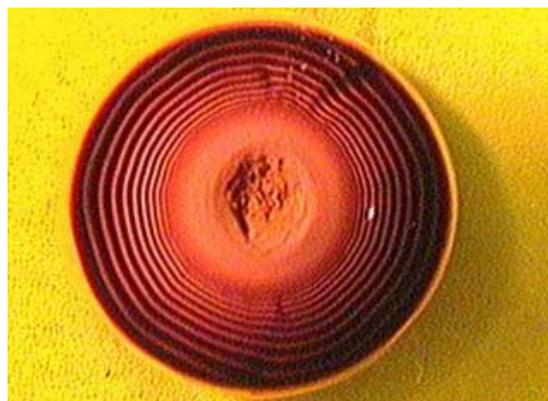


Fig. 2-3. Patrones de Liesegang en espacios bidimensionales (placa Petri). Imagen adaptada de <https://physionet.org/tutorials/e/pn/html/chp62exp1.htm>

Usualmente, la formación de los patrones ocurre a partir de distribuir homogéneamente un electrolito, denominado interno, en un gel y luego, distribuir otro electrolito externo en el mismo gel. En el caso de un dominio unidimensional (1D), el gel con electrolito interno se suele encontrar dentro de una probeta o tubo de ensayo sobre el cual se vierte el electrolito externo. Con el tiempo, se forman los patrones que en determinados casos toman la forma de bandas. En el caso bidimensional (2D), muchas veces, el gel con el electrolito interno es distribuido uniformemente sobre una base para luego aplicar el electrolito externo en forma de gota en algún punto del gel sobre el sustrato.

Estos patrones son descritos por modelos de reacción-difusión-advección basados en la difusión de especies, las reacciones de precipitación y propiedades del gel y los reactivos, que determinan la forma y posición de los patrones. Existen diversos modelos, que plantean diferentes hipótesis explicando cómo estos elementos se combinan para dar lugar a los patrones, como es el de sol-coagulación [11]. En este proyecto, se optó por seguir un modelo de sol-coagulación modificado propuesto por Lagzi [12], que será explicado a continuación, luego de explicar el original. Ambos modelos, son similares al modelo diferencial presentado en la sección 1 de este capítulo, pero anulando el término advectivo (terminó con campo vectorial de velocidad u), ya que en los sistemas estudiados se consideró una velocidad nula del solvente.

4.1. Modelos de sol-coagulación

En el contexto del modelo de sol-coagulación original, si llamamos a los electrolitos externos e internos como especies A y B respectivamente, especie C al producto intermedio soluble (generado por la reacción entre A y B) y especie D a la sustancia insoluble que se genera al precipitarse C (y compone las bandas), el mecanismo químico que genera el patrón se puede escribir como la siguiente reacción química:



La formación de las bandas está gobernada por el frente de difusión de la especie A. C se forma por detrás de este frente químico al producirse la reacción entre A y B. Una vez alcanzado un umbral de coagulación para la especie C (punto de floculación, según la teoría de sol-coagulación), esta especie se precipita como una especie D. La región precipitada de D luego crece más por medio de un proceso auto catalítico (cataliza su misma reacción), que además agota la especie C en las proximidades de la zona de precipitación. Como este proceso auto catalítico es mucho más rápido que la difusión de las especies químicas, se forman bandas separadas de especie D.

Los procesos antes mencionados, siguiendo el modelo propuesto, pueden describirse bajo el siguiente conjunto de ecuaciones en derivadas parciales:

$$\frac{\partial a}{\partial t} = D_a \nabla^2 a - kab \quad (2.4)$$

$$\frac{\partial b}{\partial t} = D_b \nabla^2 b - kab \quad (2.5)$$

$$\frac{\partial c}{\partial t} = D_c \nabla^2 c + kab - k_1 c \theta(c - c^*) - k_2 cd \quad (2.6)$$

$$\frac{\partial d}{\partial t} = k_1 c \theta(c - c^*) + k_2 cd \quad (2.7)$$

donde a , b , c y d son las concentraciones de las especies A, B, C y D respectivamente en cada punto del espacio. D_a , D_b y D_c son los coeficientes de difusión de las especies correspondientes. k es la constante de velocidad química para la reacción $A + B \rightarrow C$, k_1 y k_2 son las constantes de velocidad para la coagulación y la formación de precipitado autocatalítico, respectivamente. c^* constituye el punto de floculación mencionado previamente, correspondiente a la teoría de sol-coagulación. Por último, θ denota la función escalón unitario de Heaviside, definida como se ve en la **Fig. 2-4**.

$$\Theta(x - n) = \begin{cases} 0 & \text{si } x < n \\ 1 & \text{si } x > n \end{cases}$$

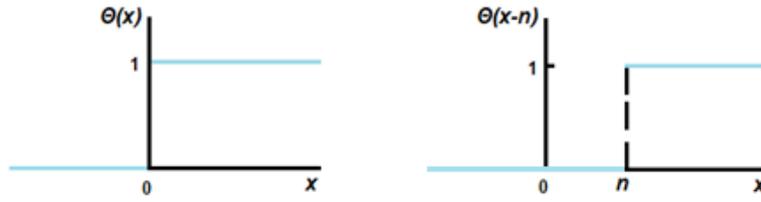


Fig. 2-4. Función escalón unitario de Heaviside

Según Lagzi, en este modelo original existen defectos que evitan la formación de patrones fieles a la situación que se da en la realidad. Como consecuencia, es necesario modificar el modelo para agregar un parámetro ρ que limita la cantidad máxima del precipitado formado en un punto determinado (limitando las concentraciones en las bandas). Si la cantidad total de precipitado alcanza este valor ρ , no se produce ningún crecimiento adicional en ese lugar ($k_1 = k_2 = 0$) y elimina el umbral de coagulación en las posiciones vecinas ($c^* = 0$), catalizando la reacción allí. Esto se traduce a las siguientes condiciones limitantes:

$$\text{si } d(t, x_n) < \rho \text{ entonces} \quad (2.8)$$

$$k_1(t, x_n) = k_1$$

$$k_2(t, x_n) = k_2$$

$$c^*(t, x_{n-1}) = c^*(t, x_{n+1}) = c^*$$

$$\text{si } d(t, x_n) \geq \rho \text{ entonces} \quad (2.9)$$

$$k_1(t, x_n) = k_2(t, x_n) = 0$$

$$c^*(t, x_{n-1}) = c^*(t, x_{n+1}) = 0$$

Donde x_n representa la n -ésima celda y t el tiempo. $k_1(t, x_n)$ representa el valor de k_1 en la celda x_n en el tiempo t ($k_2(t, x_n)$ y $c^*(t, x_n)$ representan los valores de k_2 y c^* respectivamente). $d(t, x_n)$ representa la concentración de la especie D en la celda x_n en el tiempo t .

Este modelo, es la base de la cual se partió al iniciar las actividades del proyecto. El mismo controlaría la formación de bandas tanto en su espacio, ancho y concentración. Sin embargo, a medida que se avanzó, se hizo evidente la necesidad de modificaciones en el modelo (y en consecuencia en los algoritmos que lo implementaban). Es válido mencionar aquí que el modelo propuesto por Lagzi, contempla ya en su formulación un dominio discreto, alterando la naturaleza diferencial y continua de las ecuaciones, y generando una dependencia espacial (x_{n-1} y x_{n+1}), controlada artificialmente por la discretización del problema numérico usada y no por la naturaleza topológica de los cristales insolubles formados. Esta crítica al modelo y la consecuente propuesta de modificación se expondrán en el capítulo VI.

5. Leyes de los patrones de Liesegang

Los patrones de Liesegang descritos anteriormente no se originan de forma aleatoria e impredecible. En cambio, la formación de las bandas propias del fenómeno sigue un conjunto de leyes que determinan la posición, espaciado, forma y características de las mismas [13]. Dichas leyes se describen a continuación:

5.1. Ley de Espaciado

El ratio o proporción de las posiciones de las bandas sucesivas, tiende a converger a un valor superior a 1.

$$\frac{x_{n+1}}{x_n} \rightarrow 1 + p \quad (2.10)$$

Donde x_n representa la posición de la n-ésima banda.

5.2. Ley de Ancho

El ratio o proporción del ancho de las bandas sucesivas, tiende a converger a un valor constante.

Forma débil de la ley:

$$\frac{w_{n+1}}{w_n} \rightarrow q \quad (2.11)$$

Forma fuerte de la ley:

$$q = (1 + p)^\alpha \quad (2.12)$$

donde α pertenece al rango $[0.9, 0.95]$.

Ambas formas de expresar la ley están relacionadas entre sí. Dicha relación, puede derivarse partiendo de que, generalmente, las observaciones de experimentos indican que el ancho de la n-ésima banda w_n está relacionado con la posición según la siguiente ley de potencia:

$$w_n \sim x_n^\alpha \quad (2.13)$$

donde $\alpha > 0$. Luego:

$$w_n = r x_n^\alpha \quad (2.14)$$

(con r como factor de proporción)

Si en (1) se asume n muy grande:

$$\left(\frac{x_{n+1}}{x_n}\right)^\alpha = (1 + p)^\alpha \Rightarrow \left(\frac{x_{n+1}}{1 + p}\right)^\alpha = x_n^\alpha \quad (2.15)$$

Con (15) en (14)

$$w_n = r x_{n+1}^\alpha (1 + p)^{-\alpha} \quad (2.16)$$

Según (14):

$$w_n = w_{n+1} (1 + p)^{-\alpha} \quad (2.17)$$

$$\frac{w_n}{w_{n+1}} = (1 + p)^{-\alpha} \quad (2.18)$$

$$\frac{w_{n+1}}{w_n} = (1 + p)^\alpha = q \quad (2.19)$$

5.3. Ley de Tiempo

El ratio entre la posición de una banda y la raíz cuadrada del tiempo de su aparición es constante.

$$\frac{x_n}{\sqrt{t_n}} = \text{cte} \quad (2.20)$$

Existen otras formas de representar esta ley, como las siguientes:

$$\frac{x_n^2}{t_n} = \text{cte} \quad \text{o} \quad x_n \sim \sqrt{t_n} \quad (2.21)$$

5.4. Ley de Matalon-Packter

La Ley de Matalon-Packter establece que el coeficiente de espaciado p es dependiente de las concentraciones originales de A y B (a_0 y b_0) y sigue la forma de:

$$p = F(b_0) + \frac{G(b_0)}{a_0} \quad (2.22)$$

donde F y G son funciones decrecientes.

6. Terminología

- **Caso de estudio/simulación o simplemente “caso”:** Definimos a un caso de estudio/simulación como el problema matemático conformado por el modelo diferencial (ecuaciones), dominio/geometría y condiciones de borde, además de la discretización del dominio (malla usada).

En particular, en OpenFOAM, los casos de simulación son especificados a través de un conjunto de carpetas y archivos (con una configuración de parámetros específica) que luego son usados por algoritmos de simulación para realizar la simulación propiamente dicha.

- **Solucionador/Solver:** Programa o algoritmo diseñado para la resolución de un problema matemático. Como entrada se le suministra una descripción de un problema (a veces en forma de ecuaciones) y como salida produce un resultado o solución.
- **Malla:** Llamamos malla a un conjunto de áreas o volúmenes finitos (usualmente denominado celdas) que discretizan un espacio geométrico continuo. Las celdas de una malla cubren (sin solaparse) el dominio de interés de un caso de simulación, tal que la unión de todas las celdas equivale al dominio completo.

Las mallas pueden ser estructuradas o no. En el caso de mallas estructuradas, las celdas que componen las mismas se distribuyen siguiendo un patrón uniforme.

- **Convergencia de tiempo:** Un modelo y sus correspondientes algoritmos, presentan convergencia temporal si son capaces de mostrar “estabilidad” en los resultados a partir de ciertos valores en los parámetros temporales, esto es, que ante cambios de esos parámetros (comúnmente al hacerlos más chicos), el resultado no se ve afectado (o se ve afectado de forma progresivamente menor).
- **Convergencia de malla:** Un modelo y sus correspondientes algoritmos, presentan convergencia de malla si muestran una aproximación (convergencia) hacia un resultado único a medida que se aumenta la resolución del dominio que se está simulando. Más específicamente, al incrementar la cantidad de celdas en el dominio, la variación en los resultados obtenidos debería ser cada vez menor hasta llegar al punto donde no cambie (o cambie insignificadamente).
- **Número adimensional:** Número que no tiene asignadas unidades físicas propias, por lo que representa un número “puro”. Surgen de cantidades que, si tienen unidades asignadas, pero que, al ser multiplicadas o divididas, “pierden” dichas unidades. Los valores de estos números adimensionales representan propiedades físicas o químicas de un sistema.
- **Campo escalar:** Asocia un valor numérico escalar para cada punto en el espacio. En nuestro caso, asigna un valor para cada una de las celdas que pertenecen al dominio. Este valor escalar puede ser un número adimensional o una cantidad física, por ejemplo, puede representar una temperatura, concentración, etc.
- **Función Sigmoidea:** Una función sigmoidea es simplemente una función matemática cuya curva tiene una forma de "S" o curva sigmoidea. Se puede ver un ejemplo en la **Fig. 3-1**.

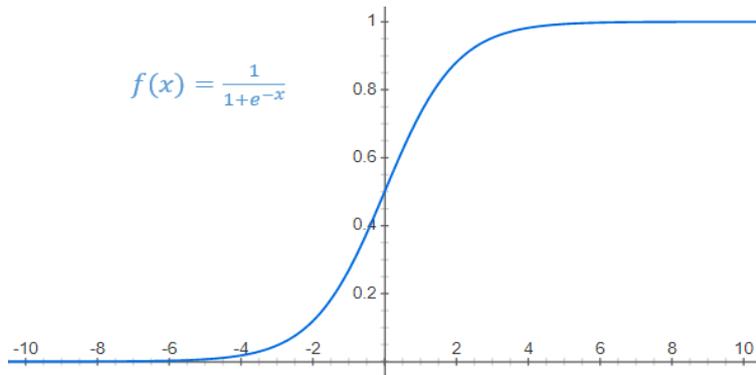


Fig. 3-1. Función sigmoidea.

- **CSV:** Un archivo de valores separados por comas (CSV) es un archivo de texto que utiliza comas para separar los valores que contiene. Las líneas del archivo representan registros de datos que pueden tener uno o más campos separados por coma.

III. Herramientas y tecnologías utilizadas

1. OpenFOAM

OpenFOAM (Open-source Field Operation And Manipulation) ([4], [14]) es un framework de código abierto (bajo la Licencia Pública General de GNU versión 3) compuesto de diversos módulos escritos en C++ para el desarrollo de solucionadores (solvers) numéricos. Este software permite la resolución de problemas de diversos tipos desde complejos flujos de fluidos que involucran reacciones químicas, turbulencias y transferencia de calor, hasta acústica, mecánica sólida y electromagnetismo. OpenFOAM permite resolver los problemas, regidos por ecuaciones en derivadas parciales, en dominios arbitrarios de hasta tres dimensiones, mediante la aplicación del método de volúmenes finitos.

La base de usuarios de este software es realmente amplia, contando con usuarios en la mayoría de las áreas de ingeniería y ciencias, en organizaciones comerciales y académicas.

Este framework fue la herramienta utilizada para realizar las simulaciones durante el proyecto porque además de las herramientas específicas que brinda para resolver los problemas de fluidos, existen herramientas adicionales con las cuales puede combinarse como Gmsh o ParaView.

Por otro lado, OpenFOAM cuenta con características que resultan deseables para el desarrollo del proyecto, como ser una interfaz flexible completamente orientada a objetos, la disponibilidad del código fuente para realizar las implementaciones deseadas, una sintaxis amigable para representar ecuaciones en derivadas parciales a ser resueltas, la posibilidad de ejecutar en mallas de diversos tipos (incluyendo mallas no estructuradas), la carga sencilla de parámetros en los casos de simulaciones a ejecutar y la salida de datos de forma apta para realizar un análisis posterior.

En el capítulo VI, se explicará en mayor detalle cómo se componen los casos a ejecutar en OpenFOAM y cuáles fueron las herramientas provistas por este que fueron utilizadas

2. porousMultiphaseFoam y porousScalarTransportFoam

porousMultiphaseFoam ([15], [16]) es un proyecto de código abierto que consta de un conjunto de herramientas para modelar el flujo multifásico en medios porosos con OpenFOAM. El flujo multifásico consiste en el flujo de fluidos donde existen más de dos fases inmiscibles. El hecho de que sea en medios porosos implica medios/materiales donde existen poros (huecos). Se considera que las fases de interés sólo pueden ocupar/fluir a través del espacio de poros del material poroso.

El flujo multifásico puede ser aplicado en numerosas situaciones, que van desde estudio y entendimiento del ecosistema (acuíferos, contaminación de suelos) a mejoras de procesos industriales (producción de petróleo, manejo de desechos). El modelado de estos flujos requiere de herramientas específicas de dinámica de fluidos computacional (CFD).

porousMultiphaseFoam provee herramientas adaptables a estudios futuros para probar nuevos modelos matemáticos y métodos numéricos. Además, brinda los modelos de propiedades efectivas más comunes de la literatura (permeabilidad relativa, presión capilar) y condiciones de borde específicas al flujo en medios porosos.

Dado que el proyecto desarrollado se caracteriza por la difusión y reacción de especies en medios porosos, se consideró apto utilizar este conjunto de herramientas de OpenFOAM para realizar, a partir de este, el modelo que describe los patrones de Liesegang. Esta elección aceleró el tiempo de desarrollo evitando la necesidad de la elaboración desde cero de solucionadores de ecuaciones en derivadas parciales características de estos experimentos, al ya encontrarse implementados y validados, reduciendo la actividad a elegir el más apto y adaptarlo a nuestro caso particular.

El solucionador elegido como base para resolver las ecuaciones involucradas al fenómeno estudiado se denomina *porousScalarTransportFoam* [16] dado que se adapta a las circunstancias de transporte de propiedades en forma de escalares en el medio poroso. Otro motivo que nos empujó a seleccionarlo, es que

dicho solucionador fue desarrollado en parte por nuestro codirector (Ing. Gabriel S. Gerlero), quien tenía amplio conocimiento sobre el funcionamiento, forma de uso, beneficios y salvedades a considerar.

Por otro lado, la elección de esta biblioteca, obedece al marco del proyecto, donde en el mediano plazo, se espera asociar los fenómenos reactivos-difusivos estudiados a flujo capilar en régimen no saturado, asociado a la tecnología de materiales mencionada en la introducción de este informe.

3. ParaView

ParaView [17] es una aplicación de análisis y visualización de datos, multiplataforma y libre. Este análisis puede hacerse rápidamente con la interfaz gráfica 3D integrada o con su funcionalidad de procesamiento batch. Puede correrse desde supercomputadoras de forma distribuida hasta en computadoras portátiles para conjuntos de datos más pequeños y es una herramienta integral en laboratorios, universidades e industrias.

ParaView se instaló directamente en Windows y proporcionó herramientas que permitieron la visualización y análisis de las salidas obtenidas en las múltiples simulaciones realizadas con OpenFOAM. De esta forma, se verificó la calidad numérica de las soluciones y en otras ocasiones la validez empírica de estas.

4. C++

C++ [18] es un lenguaje de programación abierto, estandarizado por la ISO [19], compilado, de tipado fuerte y “no seguro” (espera que los programadores sepan que están haciendo), que permite el tipado inferido o manifestado, el tipado estático, multiparadigma, portable, compatible hacia atrás con C y con una gran cantidad de librerías.

OpenFOAM y el proyecto base *porousMultiphaseFoam* están desarrollados en C++, por lo que fue el lenguaje con el que se implementaron los algoritmos del proyecto. Los miembros del equipo ya tenían experiencia programando con este lenguaje, lo que hizo que el periodo de aprendizaje e investigación fuera más orientado al uso de las herramientas específicas mencionadas.

5. Gmsh

Gmsh [20] es una herramienta de código abierto que permite generar mallas de elementos finitos y volúmenes finitos en tres dimensiones. Es una herramienta rápida, liviana, de uso sencillo, capaz de funcionar con parámetros y provee un visualizador de mallas integrado. Dispone de diversos algoritmos para la generación de mallas.

Para la especificación de entradas a Gmsh se puede hacer uso de una interfaz gráfica de usuario, archivos de texto con una sintaxis específica o mediante el uso de APIs escritas en C, C++, Python o Julia.

Gmsh fue utilizado para generar las mallas a ser utilizadas en un conjunto especial de pruebas realizadas sobre dominios bidimensionales. Las características de estas mallas permitieron hacer un estudio más detallado del método y algoritmos desarrollados en este proyecto.

6. Repositorios GitHub y GitLab

Git [21] es un sistema de control de versiones distribuido libre y gratis diseñado para funcionar y escalar con proyectos pequeños a grandes de forma veloz y eficiente. Es fácil de aprender y tiene un consumo de recursos liviano.

GitHub [22] y GitLab [23] son herramientas web de gestión de configuración y desarrollo del software. Su principal funcionalidad es la de administrar un repositorio remoto Git. Con estas herramientas, es posible configurar detalles de privacidad y acceso al proyecto, de documentación, gestión de las múltiples ramas de desarrollo, gestión de tickets y tareas asignadas, integración continua con pruebas automáticas, entre otras cosas.

El código base para el desarrollo fue clonado del repositorio del proyecto original *porousMultiphaseFoam*, ubicado en GitLab. Una vez clonado el repositorio, se lo subió a un repositorio privado en la plataforma GitHub,

por existir preferencia por esta herramienta entre los miembros del equipo de desarrollo, quienes consideraban más fácil su uso. Desde allí, se trabajó sobre múltiples ramas agregando o probando distintas funcionalidades, relativas a las actividades del proyecto.

7. Herramientas Google Workspace (Drive, Docs, Sheets, Hangouts)

El juego de herramientas Google Workspace [24] son un conjunto de aplicaciones web integradas caracterizadas por ser seguras, nativas a la nube y de alta productividad. Dentro de este conjunto de apps existen algunas muy populares como son el servicio de correo Gmail, el de almacenamiento Drive, el de videoconferencia grupal Hangouts y múltiples aplicativos para la creación de documentos como son Docs para los documentos de texto enriquecidos y Sheets para las planillas de cálculo.

La documentación realizada, como los análisis intermedios para reportar al director y codirector, o planillas de seguimiento de experimentación y otros subproductos fueron almacenados en una carpeta compartida de Google Drive. Para realizar los documentos en cuestión, se hizo uso de las herramientas Docs y Sheets.

Las reuniones realizadas con el director y codirector fueron llevadas a cabo a través de Google Hangouts debido a su robustez, facilidad de uso y por las funcionalidades provistas de compartir pantalla, además de ser posible utilizarlo directamente desde el navegador sin necesidad de instalar software adicional.

8. Python

Python [25] es un lenguaje orientado a objetos con múltiples características que lo diferencian y lo hacen un lenguaje ideal para prototipos y otras tareas de programación ad-hoc sin comprometer la mantenibilidad. Posee una sintaxis elegante que hace a los programas fácil de leer y fácil de usar simplificando el tener un programa funcionando y además viene con una amplia librería estándar que facilita muchas tareas comunes en la programación como son la conexión a servidores web, búsqueda por expresiones regulares, lectura y modificación de archivos. Es fácil de extender con módulos implementados en lenguajes compilados como C/C++, y a su vez puede ser embebido en una aplicación para permitir una interfaz programable. Además, corre en los principales sistemas operativos y su uso es libre y gratuito.

En el proyecto, originalmente se pensaba utilizar Python para la lectura y transformación de datos experimentales. Sin embargo, fue utilizado para la lectura e interpretación de salidas de simulaciones, automatizando el análisis de los patrones estudiados según las leyes que los rigen.

9. Ubuntu y WSL

Ubuntu [26] es un sistema operativo de código abierto basado en Linux, de libre disponibilidad con soporte comunitario y profesional. Su uso es gratuito, así como también la mayoría de las herramientas destinadas al mismo.

Ubuntu está basado en la arquitectura Debian y es capaz de funcionar desde equipos de escritorios hasta servidores. Es altamente personalizable y modificable de acuerdo a las necesidades de los usuarios. El enfoque en el software proporciona un amplio abanico de posibilidades en forma de aplicaciones que los usuarios pueden aprovechar.

Windows Subsystem for Linux (WSL) [27] es una capa de compatibilidad presente a partir de Windows 10 y Windows Server 2019.

WSL proporciona un mapeo de las llamadas al sistema del kernel de Windows a las llamadas al sistema del kernel de Linux. De esta forma, permite a los desarrolladores correr un entorno GNU/Linux (incluyendo la mayoría de las herramientas de líneas de comando, utilidades y aplicaciones) directamente en Windows, sin modificaciones y sin el consumo superfluo de recursos de una máquina virtual o de una configuración multi arranque.

WSL, por otro lado, también asigna los servicios de Windows, como el sistema de archivos y las redes, como dispositivos a los que puede acceder Linux.

Para el uso de OpenFOAM y herramientas relacionadas, es recomendable en entornos Linux, de lo contrario en sistemas Windows debe hacerse uso de virtualizaciones o contenedores con entornos Linux. Debido a este motivo se decidió utilizar Ubuntu como el sistema donde OpenFOAM fuese instalado.

Debido a necesidades externas al proyecto, los integrantes debían seguir disponiendo de sistemas Windows en sus equipos principales de trabajo. Ante esta situación, se probó el *Windows Subsystem for Linux* que permite la ejecución de aplicativos desarrollados para distintas distribuciones de Linux en entornos Windows. No se encontraron problemas mayores en el proyecto y pudieron hacerse todas las actividades desde este subsistema.

10. glogg

glogg [28] es una aplicación gráfica multiplataforma que permite abrir, modificar y explorar archivos de “log” de gran tamaño.

En la depuración del simulador se encontró necesaria esta herramienta porque en las primeras etapas, los logs generados a veces alcanzaban tamaños de 3 gigabytes o más. Esos archivos de gran tamaño no son manejables con editores de texto convencionales debido a que la mayoría (al menos en Windows) precargan el archivo completo en memoria antes de permitir la visualización de los mismos. Con glogg, en cambio, solamente se carga en memoria la porción del archivo que se está visualizando, lo que permite analizar el contenido de los archivos, además de la búsqueda y corrección de errores.

11. Editores de código Notepad++ y Visual Studio Code

Al editarse distintos tipos de archivos, en múltiples carpetas de forma simultánea, a veces con distintos lenguajes y/o sintaxis, existía la necesidad de editores de textos con características avanzadas.

Notepad++ [29] es un editor de texto y código libre y gratuito desarrollado con el objetivo de reemplazar el editor estándar de Windows “Notepad”. Soporta múltiples lenguajes de programación. Está desarrollado en C++ y usa directamente la API de Windows, lo que le da una alta performance. Admite la edición con pestañas, lo que permite trabajar con varios archivos abiertos en una sola ventana.

Visual Studio Code [30] es un editor de texto gratis y de código abierto desarrollado por Microsoft, que funciona en múltiples plataformas como son Windows, Linux y macOS. Entre las características que Visual Studio Code incluye se puede mencionar resaltado de sintaxis, autocompletado de código, refactorización de código y Git integrado. Admite la edición con pestañas, lo que permite trabajar con varios archivos abiertos en una sola ventana.

12. Bash

Bash [31] es la shell del proyecto GNU, compatible con la shell estándar Unix (sh), que ofrece mejoras sobre esta para la programación y su uso interactivo. Entre estas podemos enumerar la edición de líneas de comandos, un historial de comandos sin límite, control de trabajos, funciones y alias, indexación de arreglos de tamaño ilimitado y la aritmética de enteros desde base 2 a base 64.

Por la gran cantidad de factores a variar en las simulaciones, se necesitó de automatizar pruebas para evitar el error humano de cambio de parámetros manualmente. Con la codificación en Bash integrada en la línea de comandos del WSL, se eliminaron varios pasos manuales en la preparación y ejecución de casos de prueba, agilizando el trabajo y estudio de las simulaciones.

IV. Metodologías utilizadas

El proyecto desarrollado fue llevado a cabo dividiendo el trabajo realizado en 4 etapas principales. Las mismas fueron:

1. Estudio e investigación del dominio del problema y las herramientas para abordarlo.
2. Desarrollo, investigación y experimentación en casos 1D.
3. Desarrollo, investigación y experimentación en casos 2D.
4. Pruebas y reproducción de casos empíricos bibliográficos.

A su vez, las etapas contaron con una metodología específica para realizar la investigación necesaria y los desarrollos requeridos (en la primera y la última, por ejemplo, no fueron requeridos desarrollos).

1. Metodología para investigación

La investigación realizada en cada una de las etapas pretendía recopilar tanta información como fuese necesaria para dos cuestiones básicas: comprender el problema enfrentado y saber qué tipo de estrategias desarrollar para llegar a una solución suficientemente robusta.

En muchos casos, los obstáculos que se encontraban al realizar la simulación (que producían resultados no deseados) estaban relacionados a las propiedades fisicoquímicas, numéricas o, incluso, de precisión asociadas al problema a resolver. Esto obligaba a realizar investigaciones sobre el comportamiento del sistema bajo determinadas condiciones para así poder generar modelos matemáticos que pudiesen ajustarse de forma válida al problema. La metodología para la investigación involucró la lectura de bibliografía específica ([32], [12], [33]), consulta con especialistas en el tema y hasta el estudio del comportamiento de los sistemas simulados frente a diferentes estados.

En otros casos, los obstáculos se debían al desconocimiento y falta de experiencia con el framework OpenFOAM, utilizado para la resolución de los problemas de flujo (resolver ecuaciones diferenciales asociadas a la simulación). Esto requirió de extensa investigación entre la documentación relacionada al framework ([34], [35]), foros de comunidades dedicadas a la CFD [36] y la búsqueda de posibles documentos de cursos relacionados. Por lo tanto, la metodología en este tipo de investigación, consistió en buscar entre la documentación disponible, información sobre el funcionamiento y herramientas del framework que permitieran generar los algoritmos deseados.

2. Metodología para desarrollo

Para el proceso de desarrollo llevado a cabo en las etapas del proyecto, se decidió que la mejor opción era hacer uso de una metodología basada en los principios ágiles del desarrollo de software. Esto era necesario porque se buscaba reducir los riesgos presentes en el proyecto (debido a su naturaleza). Por otro lado, se requería una constante retroalimentación a medida que se avanzaba en el mismo para evaluar el avance del proyecto y disminuir potenciales incertidumbres. Esto último se logró a través de una constante interacción con el director y codirector del proyecto, quienes tienen amplios conocimientos y una gran experiencia en el dominio de los fenómenos de este tipo.

Otro aspecto importante, que era requisito en la metodología a usar, era que debía ser suficientemente flexible para permitir tener tareas y actividades más dinámicas, sujetas a las variaciones propias de una investigación. A medida que se produjeran avances en el proyecto, se esperaba mejorar el entendimiento del problema y encontrar circunstancias que requerirían variar las actividades y tareas inicialmente definidas.

La metodología ágil elegida fue Scrum, la cual fue sometida a modificaciones/adaptaciones para adecuarla a las características de un proyecto orientado a la investigación como el presentado, haciendo uso sólo de herramientas y conceptos necesarios u aplicables.

El desarrollo iterativo e incremental que Scrum propone sería ideal para alcanzar una adaptabilidad suficientemente alta dentro del proyecto, para hacer frente a la necesidad de realizar ajustes donde sea necesario. El avance incremental permitiría alcanzar pequeños resultados o hitos mientras se desarrollaban los algoritmos que resuelvan el problema planteado. Las iteraciones cumplirían con la necesidad de realizar numerosas pruebas por parte del equipo de desarrollo y de los interesados en el producto final para asegurar la validez del producto obtenido.

Scrum provee herramientas y conceptos que facilitan la definición y descripción de tareas a llevar a cabo durante el desarrollo. Entre las herramientas y conceptos utilizados se puede mencionar *product backlog*, *sprint*, *sprint backlog* y *spikes*. A continuación, se describe cada uno y las modificaciones efectuadas sobre cada uno (si es que hubo):

- **Product backlog:** Es una lista de las tareas a realizarse dentro del proyecto expresadas como objetivos. Las tareas pueden ser *user stories* (historias de usuario) o no y, a cada una de ellas, se le establece una estimación de esfuerzo y la prioridad de desarrollo. La estimación de esfuerzo se basa en *story points* (puntos de historia), para los cuales importa el valor relativo entre sí, y no el valor asignado a cada historia individual. El *product backlog* sirve para determinar el esfuerzo y duración del proyecto.
- **Sprint:** Se puede definir como un esfuerzo para realizar un desarrollo, en un tiempo finito y con un objetivo específico. Dentro de un *sprint* existen las siguientes actividades:
 - **Planificación:** Consiste en tomar, del *product backlog*, las historias de usuario que van a ser implementadas como parte de este *sprint*. Como salida se genera un *sprint backlog* con un orden de tareas específico.
 - **Ejecución:** Se llevan a cabo las tareas pertenecientes al *sprint backlog* correspondiente al *sprint* actual.
 - **Presentación de resultados y retrospectiva:** Una vez completadas las tareas del *sprint backlog*, se presentan los resultados al/a los *product owner*. Luego, se analizan los resultados y se hace una reflexión del trabajo realizado con el propósito de planificar el próximo *sprint*.
- **Sprint backlog:** Representa el conjunto de tareas a ser realizadas en cada iteración. En este proyecto se adoptó una variación de este concepto, asignando un esfuerzo estimado en *story points* también a las tareas de investigación. El esfuerzo estimado representa cuánto esfuerzo se dedicó a realizar la actividad, lo que se traduce en un periodo de tiempo. En el caso de tareas de investigación, se estima que el periodo de tiempo es suficiente para efectuar la investigación requerida.
- **Spike:** Las tareas que corresponden a la investigación y estudio de bibliografía, así como también tareas de configuración, desarrollo y prueba de herramientas/algoritmos pueden ser enmarcadas bajo este término. Se lo puede definir formalmente como “Una historia o tarea destinada a responder una pregunta o recopilar información, en lugar de producir un producto entregable” o, también, se puede decir que “un *spike* es una actividad breve y enmarcada en el tiempo que se utiliza para descubrir el trabajo necesario para lograr una tarea o historia grande y/o ambigua” [37].

Los *spikes* cuentan con un periodo de tiempo en el cual deben realizarse, una vez cumplido este periodo se debe continuar con las demás tareas.

Según las buenas prácticas, no debería agregarse tareas *spikes* en un *product backlog*, sino más bien en un *sprint backlog* [38]. Por definición, los *spikes*, ayudan a resolver problemas o despejar dudas existentes para poder continuar con la realización de otras tareas.

Scrum también hace hincapié en la asignación y uso de roles. En ese aspecto, en el proyecto se asignó el rol de *product owner* (dueño del producto) tanto al director como el codirector del proyecto. En este proyecto el *product owner* es aquel que evalúa los resultados alcanzados en los incrementos y solicitando cambios de ser necesario. Por otro lado, los *product owner* participaron en la planificación de los *sprint*, asignando prioridades y determinando tareas a realizar. En consecuencia, los *product owner* definieron las tareas a incluir en el *product backlog*.

Finalmente, el rol de *desarrollador* fue ocupado por los alumnos involucrados en la realización del proyecto. Estos, estuvieron a cargo de realizar las tareas correspondientes a cada etapa e iteración, tanto de desarrollo como investigación.

Es importante destacar que en esta adaptación de Scrum no se contaba con un rol de *scrum master*, y tampoco se hizo uso de actividades como las reuniones *daily scrum*, debido a que carecían de sentido en un proyecto de estas características. Al ser solo dos integrantes en el equipo de desarrollo, existía una forma de trabajo muy dinámica entre ambos, estando uno informado de cualquier avance o cambio realizado por el otro.

V. Desarrollo del proyecto

En este capítulo, se describe cómo fue la evolución del proyecto en cada una de las 4 etapas principales entre las cuales se repartieron las actividades de desarrollo, investigación, experimentación, etc. realizadas. Sin embargo, primero es importante definir cuál fue el *product backlog* al cual estuvo sujeto el proyecto, dado que este sufrió modificaciones en relación a lo que se definió en la planificación inicial del proyecto (como fue descrito en la sección 1.A), debido a las necesidades surgidas durante la realización.

El *product backlog*, según el cual fue guiado el proyecto, se puede observar en la **Tabla 6-1** donde se asignó una estimación y prioridad de esfuerzo a cada una de las tareas fundamentales del proyecto.

ID	Tarea	Prioridad de desarrollo	Estimación esfuerzo (<i>story points</i>)
1	Configuración del entorno del trabajo.	Baja	2
2	Estudiar el funcionamiento y características de herramientas.	Baja	2
3	Implementación modelo matemático 1D inicial.	Media	5
4	Lograr convergencia temporal para dominios 1D.	Media	5
5	Lograr convergencia de malla para dominios 1D.	Media	3
6	Implementación modelo matemático 2D inicial.	Media	7
7	Lograr convergencia temporal para dominios 2D.	Alta	3
8	Lograr convergencia de malla para dominios 2D.	Alta	4
9	Pruebas/validación del caso 2D con casos de bibliografía.	Alta	3
10	Generación de la documentación.	Alta	8
		TOTAL	50

Tabla 6-1. Product backlog.

Es importante notar que la estimación de esfuerzo (*story points*) realizada para todas las tareas del *product backlog* no suma el total indicado al final de la **Tabla 6-1**. Esto es así, porque se toma en consideración también los *story points* asignados a tareas de tipo *spikes*. Debido a las buenas prácticas mencionadas al introducir el concepto de *spike* (capítulo IV, sección 2), no debería agregarse dichas tareas en el mencionado artefacto. Los *spikes* sí se incluyeron dentro de los *sprint backlogs*.

En la **Tabla 6-2** se muestra la distribución de tareas del *product backlog* en *sprints* realizada, generando lo que se conoce en SCRUM como *sprint backlogs* para cada una. En los *sprint backlogs* para cada *sprint* se indica el orden en el cual serían realizadas las tareas.

Para los *spikes* (incluidos en la tabla en color azul) también se realizaron estimaciones de esfuerzo mediante *story points*, para poder traducir este esfuerzo a periodos de tiempo. Se estimaba que dichos períodos de tiempo serían suficientes para realizar la/el investigación/estudio requerida/o. Tanto para los *sprints* como para los *spikes* se determinaron fechas tentativas de inicio y fin.

Como puede observarse en la **Tabla 6-2**, no se consideró un periodo de suspensión de las actividades del proyecto, para rendir exámenes relacionados a materias de la carrera. Se tomó esta decisión dado que solo un alumno tendrá dichas obligaciones, por lo que es posible continuar con las actividades, aunque a un menor ritmo, que luego sería compensado.

Desde	Hasta	Sprint	Tarea	SP ⁴ por tarea	SP ⁵ por sprint
19/12/19	02/01/20	1	Estudio e investigación teórica del dominio del problema.	4	8
02/01/20	09/01/20		Configuración del entorno del trabajo.	2	
09/01/20	16/01/20		Estudiar el funcionamiento y características de herramientas.	2	
16/01/20	20/01/20	2	Familiarización Código de <i>porousScalarTransportFoam</i> .	1	8
20/01/20	06/02/20		Implementación modelo matemático 1D inicial en <i>porousScalarTransportFoam</i> .	5	
06/02/20	13/02/20		Pruebas y estudio convergencia tiempo y malla.	2	
13/02/20	02/03/20	3	Desarrollo algoritmos para lograr convergencia temporal para dominios 1D.	5	5
			Pruebas de convergencia temporal sobre dominios 1D.		
02/03/20	12/03/20	4	Desarrollo algoritmos para lograr convergencia de malla para dominios 1D.	3	3
			Pruebas de convergencia de malla sobre dominios 1D.		
12/03/20	06/04/20	5	Implementación modelo matemático 2D inicial.	7	8
06/04/20	09/04/20		Pruebas y ajustes de compatibilidad hacia atrás con dominios 1D	1	
09/04/20	20/04/20	6	Desarrollo algoritmos para lograr convergencia temporal para dominios 2D.	3	3
			Pruebas convergencia temporal sobre dominios 2D.		
20/04/20	04/05/20	7	Desarrollo algoritmos para lograr convergencia de malla para dominios 2D.	4	4
			Pruebas de convergencia de malla sobre dominios 2D.		
04/05/20	14/05/20	8	Estudio de la bibliografía relacionada a casos 2D.	3	3
			Pruebas del caso 2D para recrear resultados de bibliografía.		
14/05/20	14/06/20		Generación de la documentación.	8	

Tabla 6-2. *Sprint backlog* para cada *sprint*.

Por otro lado, puede observarse que no se incluyeron períodos de vacaciones. Esto se debe a que ninguno de los alumnos involucrados gozará de éstas en el periodo de tiempo donde se llevará a cabo el proyecto. Para el caso del director y codirector, no fue necesario incluir este tipo de periodo, dado que, en todo momento, se iban a poder realizar reuniones o comunicar con al menos uno de ellos.

La **Tabla 6-2** muestra el conjunto de *sprint backlogs* en una sola tabla, donde de forma global, se puede observar cómo se planeaba proceder con del proyecto.

Esta actualización de la planificación se realizó en etapas muy tempranas del proyecto, previas al comienzo formal de las actividades involucradas, por lo que no fue necesario ningún tipo de revisión sobre la fecha de inicio.

⁴ *Story points*

⁵ *Story points*

Como en la planificación original, se definió a un *story point* como 10 horas de trabajo, a partir de los 50 *story point* totales y de la dedicación semanal de 20 horas totales del equipo de desarrollo, se obtuvo una duración estimada inicial de 25 semanas de trabajo o, en otras palabras, un poco más de 6 meses de trabajo.

Es importante mencionar que se optó por no incluir la tarea relativa a generación de la documentación dentro de un *sprint* específico debido a que no se considera como parte del proceso de desarrollo e investigación relacionado a la realización del proyecto presentado.

A continuación, se presentará el detalle de cómo fueron llevadas a cabo las tareas de los *sprints* en el marco de las etapas principales correspondientes. Se realiza un análisis de la planificación original del *sprint*, como fue ejecutado el mismo y, finalmente, se presenta la gestión de riesgos correspondiente, en base al plan de gestión de riesgos utilizado⁶. La generación de la documentación será analizada en una sección separada luego de las etapas centrales del proyecto. Finalmente, se procederá a hacer mención de algunos riesgos que se considera que, en gran parte, estuvieron activos a lo largo de todo el proyecto.

1. Etapa de estudio e investigación del dominio del problema y las herramientas para abordarlo

Sprint 1

Planificación

Este *sprint* estaba constituido por las tareas mostradas en el *sprint backlog* mostrado en la **Tabla 6-3**. En la misma tabla, se puede observar cuál fue la planificación inicial en cuanto a tiempos y esfuerzo total.

Desde	Hasta	Tarea	SP por tarea	SP por iteración
19/12/19	02/01/20	Estudio e investigación teórica del dominio del problema.	4	8
02/01/20	09/01/20	Configuración del entorno del trabajo.	2	
09/01/20	16/01/20	Estudiar el funcionamiento y características de herramientas.	2	

Tabla 6-3. Planificación del *sprint* 1.

En este *sprint*, lo primero que se deseaba realizar era introducirse y profundizar en los conceptos teóricos que estaban relacionados al dominio del problema a ser resuelto. Tanto en el aspecto computacional de la solución que se pretendía desarrollar como en el aspecto fisicoquímico del fenómeno que iba a ser estudiado y se pretende simular correctamente. Para esto se acudió a bibliografía específica suministrada por el director y codirector del proyecto.

Una vez obtenida la base teórica necesaria, era necesario realizar todas las configuraciones necesarias sobre el entorno de trabajo, es decir, las herramientas y archivos que iban a ser utilizados durante el proyecto, es decir OpenFOAM, ParaView, etc.

Finalmente, se pretendía estudiar el funcionamiento y características de las herramientas mencionadas, que iban a ser usadas para realizar las simulaciones correspondientes y estudiar sus resultados.

Estas dos últimas tareas también fueron realizadas basándose en bibliografía específica facilitada por el director y codirector del proyecto.

Ejecución

La ejecución real del *sprint* se dio como se ilustra en la **Tabla 6-4**.

⁶ El plan de gestión de riesgos utilizado (con la lista de todos los riesgos identificados y considerados, así como también las acciones/respuestas consecuentes) se incluye en el Anexo A de este documento

Desde	Hasta	Tarea	SP por tarea	SP por iteración
20/12/19	10/01/20	Estudio e investigación teórica del dominio del problema.	6	8
10/01/20	13/01/20	Configuración del entorno del trabajo.	1	
13/01/20	16/01/20	Estudiar el funcionamiento y características de herramientas.	1	

Tabla 6-4. Ejecución del *sprint* 1.

Como se puede observar en la **Tabla 6-4**, la duración del *sprint* fue aproximadamente la estimada originalmente. Sin embargo, la duración de la primera tarea del *sprint* se extendió más de lo estimado originalmente. Esto se dio de esta manera debido a que fue necesaria más investigación y aprendizaje relacionado al dominio del problema de lo pensado, es decir, investigar y aprender sobre los fundamentos teóricos de CFD, FVM, etc.

Por otro lado, la segunda y tercera tarea del *sprint* demostraron ser más sencillas de lo pensado inicialmente.

Finalizada el *sprint* 1 se procedió a actualizar el *product backlog* Inicial eliminando aquellas tareas ya realizadas. El resultado se observa en la **Tabla 6-5**.

ID	Tarea	Prioridad de desarrollo	Estimación esfuerzo (<i>story points</i>)
3	Implementación modelo matemático 1D inicial.	Media	5
4	Lograr convergencia temporal para dominios 1D.	Media	5
5	Lograr convergencia de malla para dominios 1D.	Media	3
6	Implementación modelo matemático 2D inicial.	Media	7
7	Lograr convergencia temporal para dominios 2D.	Alta	3
8	Lograr convergencia de malla para dominios 2D.	Alta	4
9	Pruebas/validación del caso 2D con casos de bibliografía.	Alta	3
10	Generación de la documentación.	Alta	8
		TOTAL	50

Tabla 6-5. *Product backlog* actualizado luego del *sprint* 1.

Gestión de Riesgos

En base al plan de gestión de riesgos utilizado, se puede decir que durante la ejecución de este *sprint* el riesgo que se materializó fue R8, que se muestra en la **Tabla 6-6**.

Ref.	Riesgo	Probabilidad	Impacto
R8	Falta de conocimiento teórico sobre la problemática para desarrollar la aplicación	Muy Alta	Alto

Tabla 6-6. Riesgos activados durante el *sprint* 1.

El riesgo R8 se activó en la primera tarea del *sprint*. Frente a este riesgo, la respuesta especificada originalmente en el plan de gestión de riesgos fue la mostrada en la **Tabla 6-7**.

Ref.	Acción	Respuesta
R8	Asumir	Se crearon etapas de investigación de manera iterativa considerando este riesgo.

Tabla 6-7. Respuestas especificadas en la planificación para los riesgos activados durante el *sprint* 1.

Debido a que la activación de este riesgo se dio justamente en una etapa de investigación creada como parte de la respuesta adoptada para asumir la existencia de dicho riesgo, no hubo mucho que pueda hacerse en cuanto a acciones correctivas. Lo único que se pudo hacer fue extender la duración de esta tarea dentro del *sprint* para evitar mayores complicaciones a futuro.

2. Etapa de desarrollo, investigación y experimentación en casos 1D

Sprint 2

Planificación

En la **Tabla 6-8** se puede observar cuales eran las tareas asignadas a este *sprint*. Además, se puede observar la planificación inicial en cuanto a tiempo y esfuerzo total.

Desde	Hasta	Tarea	SP por tarea	SP por iteración
16/01/20	20/01/20	Familiarización Código de <i>porousScalarTransportFoam</i> .	1	8
20/01/20	06/02/20	Implementación modelo matemático 1D inicial en <i>porousScalarTransportFoam</i> .	5	
06/02/20	13/02/20	Pruebas y estudio convergencia tiempo y malla.	2	

Tabla 6-8. Planificación del *sprint* 2.

En esta iteración, la primera tarea consistió en que los integrantes del equipo de desarrollo se familiaricen con el código de *porousScalarTransportFoam* y el uso que éste hacía del framework de OpenFOAM. Este código base era donde se iban a desarrollar los algoritmos (usando C++) que implementarían el modelo matemático propuesto que describiría la formación de patrones de Liesegang. La segunda tarea de la iteración consistió en efectivamente implementar el mencionado modelo matemático.

La tercera tarea consistió en realizar pruebas sobre el modelo finalmente implementado para analizar y estudiar la convergencia en tiempo y malla (conceptos explicados en el capítulo II, sección 6) sobre ese modelo inicial. Era probable que los resultados presentados no tuviesen una adecuada convergencia que luego debía ser corregida en actividades posteriores.

Ejecución

La ejecución real del *sprint* se dio como se ilustra en la **Tabla 6-9**.

Desde	Hasta	Tarea	SP por tarea	SP por iteración
16/01/20	22/01/20	Familiarización Código de <i>porousScalarTransportFoam</i> .	2	11
22/01/20	31/01/20	Implementación modelo matemático 1D inicial en <i>porousScalarTransportFoam</i> .	5	
31/01/20	16/02/20	Pruebas y estudio convergencia tiempo y malla.	4	

Tabla 6-9. Ejecución del *sprint* 2.

Como se muestra en la **Tabla 6-9**, la duración del *sprint* fue más larga de lo estimado originalmente. Si bien la segunda tarea resultó ser más corta de lo esperado, la primera y última técnicamente duplicaron su duración.

La primera tarea resultó ser más extensa de lo esperado debido a que el código de *porousScalarTransportFoam* contenía una complejidad matemático-algorítmica mayor de lo esperada, relacionada al uso de OpenFOAM y las herramientas que provee. Esta complejidad hizo necesario un estudio más detenido de cómo eran realizadas las operaciones involucradas.

La segunda tarea, relacionada a implementar el modelo matemático básico presentado en el artículo usado como referencia [12], pudo realizarse en el tiempo estimado dado a que las herramientas provistas por OpenFOAM ofrecían una forma sencilla de implementar ciertas operaciones matemáticas y, por otro lado, la base de *porousScalarTransportFoam* sugirió una forma intuitiva de organizar el código. La mayor dificultad encontrada en esta tarea se originó en entender cómo usar las herramientas de OpenFOAM, dado que, en ciertos casos particulares, la documentación disponible era escasa.

Finalmente, la tercera tarea requirió de un tiempo mayor al inicialmente estimado debido a la gran cantidad de parámetros fisicoquímicos y computacionales que podían establecerse (cuestión que podrá visualizarse en los capítulos VI y VII). Se requirieron de numerosas pruebas (con distintos conjuntos de parámetros) para estudiar el comportamiento relacionado a convergencia temporal y de dominio del modelo implementado. Estas pruebas buscaban determinar si el modelo matemático inicialmente planteado en el artículo de referencia contaba o no con dichas características. De no contar con ellas, debía lograrse para poder proseguir con las demás tareas de los siguientes *sprints*.

Finalizado el *sprint 2* se procedió a actualizar el *product backlog* Inicial eliminando aquellas tareas ya realizadas. El resultado se observa en la **Tabla 6-10**.

ID	Tarea	Prioridad de desarrollo	Estimación esfuerzo (story points)
4	Lograr convergencia temporal para dominios 1D.	Media	5
5	Lograr convergencia de malla para dominios 1D.	Media	3
6	Implementación modelo matemático 2D inicial.	Media	7
7	Lograr convergencia temporal para dominios 2D.	Alta	3
8	Lograr convergencia de malla para dominios 2D.	Alta	4
9	Pruebas/validación del caso 2D con casos de bibliografía.	Alta	3
10	Generación de la documentación.	Alta	8
		TOTAL	50

Tabla 6-9. *Product backlog* actualizado luego del *sprint 2*.

Gestión de Riesgos

En base al plan de gestión de riesgos utilizado, se detectó que durante la ejecución de este *sprint* los riesgos que se materializaron fueron R8, R12, R14 y R15, que se muestran en la **Tabla 6-10**.

El riesgo R8 se activó en la primera tarea del *sprint*. El riesgo R12 se activó en la primera y segunda, debido a la activación de R8 y a la falta de conocimiento de la herramienta OpenFOAM (una variante del riesgo R8). El riesgo R12, en este caso, se refiere a que no fue trivial aprender a utilizar correctamente OpenFOAM y las herramientas que este provee.

Además, el riesgo R14 se activó sobre la última tarea y el riesgo R15 sobre la segunda tarea.

Ref.	Riesgo	Probabilidad	Impacto
R8	Falta de conocimiento teórico sobre la problemática para desarrollar la aplicación.	Muy Alta	Alto
R12	Problemas de ejecución del software utilizado para realizar el proyecto	Bajo	Alto
R14	Las validaciones al final de cada sprint demoran más de lo esperado.	Medio	Medio
R15	Las actividades de investigación se extienden por dificultad de comprensión del material.	Medio	Alto

Tabla 6-10. Riesgos activados durante el *sprint 2*.

Los riesgos R8 y R14, que fueron catalogados con un nivel de “Riesgoso” o “Muy riesgoso” debían contar con una estrategia a llevar a cabo si se materializaban. Las respuestas especificadas originalmente para estos riesgos en el plan de gestión de riesgos fueron las mostradas en la **Tabla 6-11**.

Ref.	Acción	Respuesta
R8	Asumir	Se crearon etapas de investigación de manera iterativa considerando este riesgo.
R15	Reducir	Ya se obtuvo y estudió material inicial relacionado al proyecto de investigación. Se considera que el acceso a material nuevo se puede realizar de manera progresiva para facilitar la comprensión. Además, el director del proyecto tiene experiencia en la temática a tratar.

Tabla 6-11. Respuestas especificadas en la planificación para los riesgos activados durante el *sprint 2*.

En relación al riesgo R8, debido a que se activó en la primera tarea del *sprint*, lo único que se pudo hacer fue realizar una investigación y estudio más en profundidad del funcionamiento de *porousScalarTransportFoam*.

En cuanto al segundo riesgo, R15, lo que se hizo fue aumentar la interacción con el director y codirector del proyecto, quienes cuentan con experiencia trabajando con temáticas relacionadas y en especial implementando códigos que hacen uso de lo que OpenFOAM provee. De esta forma, se pudo solventar el problema generado por la complejidad o incompletitud del material/bibliografía relacionado a OpenFOAM y también obtener información de primera mano sobre el funcionamiento de *porousScalarTransportFoam* (debido a que el codirector del proyecto estuvo involucrado en su desarrollo).

Sprint 3

Planificación

Las tareas asignadas a este *sprint*, al igual que la planificación inicial en cuanto a tiempo y esfuerzo total pueden visualizarse en la **Tabla 6-12**.

Desde	Hasta	Tarea	SP por tarea	SP por iteración
13/02/20	02/03/20	Desarrollo algoritmos para lograr convergencia temporal para dominios 1D.	5	5
		Pruebas de convergencia temporal sobre dominios 1D.		

Tabla 6-12. Planificación del *sprint 3*.

En este *sprint* (y de algunos de los siguientes) están contenidas tareas de desarrollo, que los desarrolladores consideran compuestas de dos tareas estrechamente relacionadas, como son el desarrollo propiamente dicho y las pruebas de lo desarrollado.

Las tareas de desarrollo de algoritmos computacionales usualmente van acompañadas de pruebas que determinan la efectividad y eficiencia de estos para saber si existe la necesidad de realizar cambios/ajustes sobre los algoritmos. Esto es un proceso iterativo que puede repetirse una cantidad indeterminada de veces, que depende del algoritmo, metodología de trabajo, del entorno, de la aplicación de los algoritmos, etc. Debido a la naturaleza de la relación entre estas dos tareas se consideraron a ambas simultáneamente al asignar tiempo y esfuerzo.

El propósito de este *sprint* era el de hacer las modificaciones necesarias para lograr que el modelo matemático implementado con OpenFOAM tuviera convergencia temporal al simular patrones de Liesegang (lo cual se mostró necesario en el *sprint* anterior). El concepto de convergencia temporal se describió en el capítulo II, en la sección 6.

Este *sprint*, tal vez podría haber sido modificado para tener un conjunto de tareas más “pequeñas” que otorgaran mayor granularidad y, de esa forma, permitir un control más detallado sobre el progreso del mismo. Sin embargo, debido a la naturaleza de la tarea y del proyecto, no era posible dividirla en tareas más pequeñas dado que al momento de planificar no existía suficiente certeza del camino a ser tomado para lograr el propósito del *sprint*. Se podrá observar que esta situación relacionada a la granularidad de tareas y la consideración realizada para la asignación de tiempo y esfuerzo (a tareas muy relacionadas), se repiten en los próximos *sprints*.

Por otro lado, este *sprint* podría haber sido no necesario, debido a que existía la posibilidad de que el modelo implementado hubiese mostrado convergencia temporal desde un principio, sin embargo, ese no fue el caso.

Ejecución

Los tiempos de ejecución reales del *sprint* se dieron como se ilustra en la **Tabla 6-13**.

Desde	Hasta	Tarea	SP por tarea	SP por iteración
16/02/20	11/03/20	Desarrollo algoritmos para lograr convergencia temporal para dominios 1D.	7	7
		Pruebas de convergencia temporal sobre dominios 1D.		

Tabla 6-13. Ejecución del *sprint* 3.

En la **Tabla 6-13**, se puede ver que las tareas del *sprint* resultaron ser aún más largas en duración que lo estimado inicialmente. Esto sucedió porque se debieron probar distintos enfoques en el intento de lograr la convergencia temporal. Inicialmente se intentó modificaciones menores (pero significativas) a la forma en que el avance de tiempo de la simulación era controlado en `porousScalarTransportFoam`. Sin embargo, la solución que demostró ser la correcta fue la de generar un nuevo mecanismo (basado en conceptos químicos y matemáticos) que controlase el avance de tiempo en la simulación de forma adecuada al tipo de fenómeno con el cual se estaba trabajando. Este mecanismo y su fundamentación está explicado en la sección 8 del capítulo VI.

Otro motivo por el cual el *sprint* tuvo una duración mayor es debido al hecho de que las simulaciones que debían ser ejecutadas como forma de prueba del algoritmo muchas veces debían ser configuradas con parámetros de tiempo que producían un avance muy lento del tiempo de simulación, lo que se traducía a mayores tiempos de espera para poder ver los resultados.

Finalizado el *sprint* 3, se procedió a actualizar el *product backlog* Inicial eliminando aquellas tareas ya realizadas. El resultado se observa en la **Tabla 6-14**.

ID	Tarea	Prioridad de desarrollo	Estimación esfuerzo (<i>story points</i>)
5	Lograr convergencia de malla para dominios 1D.	Media	3
6	Implementación modelo matemático 2D inicial.	Media	7
7	Lograr convergencia temporal para dominios 2D.	Alta	3
8	Lograr convergencia de malla para dominios 2D.	Alta	4
9	Pruebas/validación del caso 2D con casos de bibliografía.	Alta	3
10	Generación de la documentación.	Alta	8
		TOTAL	50

Tabla 6-14. *Product backlog* actualizado luego del *sprint* 3.

Gestión de Riesgos

Durante este *sprint*, se detectó que los riesgos que se materializaron fueron R8, R12, R15 y R16, que se muestran en la **Tabla 6-15**.

Ref.	Riesgo	Probabilidad	Impacto
R8	Falta de conocimiento teórico sobre la problemática para desarrollar la aplicación.	Muy Alta	Alto
R12	Problemas de ejecución del software utilizado para realizar el proyecto.	Baja	Alto
R15	Las actividades de investigación se extienden por dificultad de comprensión del material.	Media	Alto
R16	El desarrollo fue subestimado.	Media	Alto

Tabla 6-15. Riesgos activados durante el *sprint* 3.

El riesgo R8 se activó en la primera y segunda tarea del *sprint*. El riesgo R12 se activó sobre la primera tarea, debido a la falta de experiencia con OpenFOAM que dio origen a problemas para hacer uso del mismo y las herramientas que provee.

Los riesgos R8, R15 y R16, fueron catalogados con un nivel de “Riesgoso” o “Muy riesgoso” por lo que debían contar con una estrategia a llevar a cabo si se materializaban. Las respuestas especificadas originalmente para estos riesgos en el plan de gestión de riesgos fueron las mostradas en la **Tabla 6-16**.

Ref.	Acción	Respuesta
R8	Asumir	Se crearon etapas de investigación de manera iterativa considerando este riesgo.
R15	Reducir	Ya se obtuvo y estudió material inicial relacionado al proyecto de investigación. Se considera que el acceso a material nuevo se puede realizar de manera progresiva para facilitar la comprensión. Además, el director del proyecto tiene experiencia en la temática a tratar.
R16	Asumir	Es posible que las estimaciones iniciales de tiempo de desarrollo se vean afectadas por desconocimiento de la temática o falta de información.

Tabla 6-16. Respuestas especificadas en la planificación para los riesgos activados durante el *sprint* 3.

En relación al riesgo R8, debido a que se activó en la primera tarea del *sprint*, lo único que se pudo hacer fue realizar una investigación y estudio en mayor profundidad del funcionamiento de *porousScalarTransportFoam* y OpenFOAM.

En cuanto al riesgo R15, al igual que en el *sprint* anterior, lo que se hizo fue aumentar la interacción con el director y codirector del proyecto a fin de reducir el impacto generado por la complejidad o incompletitud del material/bibliografía relacionada a OpenFOAM.

Finalmente, para el riesgo R16, lo único que pudo hacerse fue extender la duración de la iteración para poder realizar los desarrollos mencionados.

Sprint 4

Planificación

En la **Tabla 6-17** se listan las tareas relacionadas a este *sprint* y como fue la planificación inicial del mismo.

Desde	Hasta	Tarea	SP por tarea	SP por iteración
02/03/20	12/03/20	Desarrollo algoritmos para lograr convergencia de malla para dominios 1D.	3	3
		Pruebas de convergencia de malla sobre dominios 1D.		

Tabla 6-17. Planificación del *sprint* 4.

Similarmente al *sprint* anterior, este *sprint* tenía como propósito alcanzar niveles de convergencia que el modelo matemático inicialmente implementado no tenía. En este caso, se buscó alcanzar un nivel de convergencia de malla aceptable.

En este *sprint* se vuelve a presentar la situación mencionada en el *sprint* 3 sobre la granularidad de tareas y la asignación de tiempos y esfuerzos.

Finalmente, este *sprint* también podría haber resultado innecesario si en el modelo implementado hubiese existido una convergencia de malla aceptable.

Ejecución

El *sprint* 4 se desarrolló en los tiempos y con el esfuerzo indicado en la **Tabla 6-18**.

Desde	Hasta	Tarea	SP por tarea	SP por iteración
11/03/20	22/04/20	Desarrollo algoritmos para lograr convergencia de malla para dominios 1D.	12	12
		Pruebas de convergencia de malla sobre dominios 1D.		

Tabla 6-18. Ejecución del *sprint* 4.

De acuerdo a los datos mostrados en la **Tabla 6-18** se puede ver que este *sprint* fue ampliamente subestimado al momento de planificarlo. Este *sprint* se mostró de especial complejidad debido a un conjunto de situaciones que terminaron por alargar su duración de manera excesiva (al menos en comparación a lo estimado inicialmente). A continuación, se listan las razones principales de esto:

- La convergencia de malla o espacial presenta una relación con la convergencia de tiempo. El problema estudiado posee la característica de ser un sistema acoplado en el hecho de que las variables temporales están estrechamente relacionadas a las variables espaciales. Esta situación se puede observar en las ecuaciones diferenciales que describen el problema mostradas en la sección 4 del capítulo II. El acoplamiento, produce una dependencia entre variables temporales y de dominio, que a su vez originan dependencia entre la convergencia temporal y la convergencia de malla. Si bien un algoritmo y

mínimos de valores de parámetros pueden funcionar bien en cuanto a convergencia temporal, puede que se necesiten modificaciones (en el algoritmo o en los parámetros) para lograr convergencia espacial. Es posible que los parámetros deban estar por debajo de cierto umbral menor que el anterior o que el algoritmo deba ser modificado para que el modelo matemático que aplica funcione de manera distinta.

- Mayor cantidad de parámetros a considerar al configurar la simulación. Debido al acoplamiento de variables, la cantidad de parámetros de la simulación a considerar al intentar obtener convergencia de malla aumentó, en relación a lo que se consideraba previamente. Esto implicó, en consecuencia, realizar una mayor cantidad de pruebas en esta etapa para conocer los efectos de los parámetros, lo que implicaba más consideraciones a ser realizadas en los algoritmos. Para todos estos parámetros se debía hallar los valores mínimos necesarios para que no se genere una divergencia en los resultados.
- Para probar la convergencia de malla, las mallas deben cambiar en densidad (cantidad de celdas del dominio). Cuando uno desea averiguar si existe convergencia de malla, las mallas deben variar su cantidad de celdas (usualmente aumentando en número) para encontrar convergencia (se profundiza esta idea en el capítulo VII) por lo que las simulaciones tienden rápidamente a ser más costosas a nivel computacional, lo que a su vez requiere de mucho más tiempo para que sean completadas y poder observar los resultados.

Inicialmente existieron dos enfoques distintos a nivel de algoritmos para intentar obtener convergencia de malla (ambos con resultados prometedores). Sin embargo, al final, se eligió el que se consideraba mejor, pero implicaba modificaciones sobre el algoritmo relacionado a la convergencia temporal (para permitir mayor flexibilidad sobre mallas más complejas que aumentan la convergencia espacial).

Finalizado el *sprint 4* se procedió a actualizar el *product backlog* Inicial eliminando aquellas tareas ya realizadas. El resultado se observa en la **Tabla 6-19**.

ID	Tarea	Prioridad de desarrollo	Estimación esfuerzo (story points)
6	Implementación modelo matemático 2D inicial.	Media	7
7	Lograr convergencia temporal para dominios 2D.	Alta	3
8	Lograr convergencia de malla para dominios 2D.	Alta	4
9	Pruebas/validación del caso 2D con casos de bibliografía.	Alta	3
10	Generación de la documentación.	Alta	8
		TOTAL	50

Tabla 6-19. *Product backlog* actualizado luego del *sprint 4*.

Gestión de Riesgos

A lo largo de esta iteración, los riesgos que se materializaron son R5, R14, R16 y R20. Sus detalles se muestran en la **Tabla 6-20**.

Los riesgos R5 y R19 se activaron en la tarea de desarrollo, donde en un primer momento se diseñó un algoritmo que aportaba a la convergencia de malla pero que era innecesariamente complejo en su funcionamiento y era muy difícil de mantener. En una refactorización posterior, este código fue reformulado obteniendo los mismos resultados y de una forma más sencilla.

El riesgo R14, por otro lado, se activó durante la segunda tarea, la relacionada a pruebas, debido a que se requirió mucho más tiempo del estimado en un principio para poder hacer validaciones del funcionamiento de los algoritmos desarrollados. Esto fue en parte porque las pruebas requirieron mucho tiempo para finalizar y porque se requirieron muchas pruebas diferentes.

Ref.	Riesgo	Probabilidad	Impacto
R5	Pérdida de tiempo y aumento de complejidad por sobre-ingeniería.	Baja	Alto
R14	Las validaciones al final de cada sprint demoran más de lo esperado.	Media	Medio
R16	El desarrollo fue subestimado.	Media	Alto
R19	Deterioro de la calidad del código debido a la falta de comprensión del problema impactando negativamente en la mantenibilidad.	Media	Medio
R20	Algoritmos desarrollados no se aplican a todos los casos posibles.	Alta	Alto

Tabla 6-20. Riesgos activados durante el *sprint* 4.

Los riesgos R16 y R20 se activaron durante la primera tarea. El desarrollo llevado a cabo fue subestimado en su complejidad y duración, al realizar la planificación previa al inicio del proyecto. Por otro lado, los primeros algoritmos desarrollados, durante la primera tarea de este *sprint*, probaron no ser aplicables a todos los casos posibles de simulaciones durante la fase de pruebas (segunda tarea). Esta incompatibilidad se debía a que las mallas usadas tenían propiedades no constantes a lo largo de las mismas (el tamaño de las celdas).

Debido a que los riesgos R14, R16 y R20 fueron catalogados con un nivel de “Riesgoso” o “Muy riesgoso”, debían contar con una estrategia a llevar a cabo si se materializaban. Las respuestas especificadas originalmente para estos riesgos en el plan de gestión de riesgos fueron las mostradas en la **Tabla 6-21**.

Ref.	Acción	Respuesta
R14	Asumir	Se determinaron tiempos de validación que consideran este riesgo, por lo que, si se materializa, será de un impacto menor.
R16	Asumir	Es posible que las estimaciones iniciales de tiempo de desarrollo se vean afectadas por desconocimiento de la temática o falta de información.
R19	Asumir	Las características del problema plantean un escenario donde un enfoque inicial no permite un avance claro en la estructuración del código.
R20	Reducir	Se pasará por múltiples etapas de validaciones considerando este riesgo. Se realizará una batería de pruebas para reducir la probabilidad.

Tabla 6-21. Respuestas especificadas en la planificación para los riesgos activados durante el *sprint* 4.

En relación a las acciones llevadas a cabo para los distintos riesgos se debe mencionar que en el primer y segundo caso (R14 y R16), no hubo mucho que pudiese hacerse cuando se volvieron una realidad debido a que la única acción que se podía tomar era prolongar la duración del *sprint*. Evidentemente, el impacto por el suceso de R14 no fue menor, como se supuso inicialmente, dado que los tiempos de validación especificados fueron insuficientes y se requirió de más tiempo.

Como se mencionó antes, frente a la calidad deteriorada del código debido al suceso del riesgo R19, se hizo una refactorización de una porción del algoritmo para lograr convergencia de malla para elevar la calidad del mismo, facilitando mantenibilidad a futuro.

Finalmente, para el caso de R20, se realizaron una gran cantidad de pruebas durante el *sprint* para validar, que demostraron ser valiosas para encontrar casos especiales donde el algoritmo mostraba falencias que debían corregirse. Estas falencias fueron corregidas a través de la repetición de la tarea de desarrollo (y luego de pruebas) del *sprint*.

3. Etapa de desarrollo, investigación y experimentación en casos 2D

Sprint 5

Planificación

Las tareas, así como también la estimación de tiempos y esfuerzos, para las tareas relacionadas a este *sprint* se muestran en la **Tabla 6-22**.

Desde	Hasta	Tarea	SP por tarea	SP por iteración
12/03/20	06/04/20	Implementación modelo matemático 2D inicial.	7	8
06/04/20	09/04/20	Pruebas y ajustes de compatibilidad hacia atrás con dominios 1D	1	

Tabla 6-22. Planificación del *sprint* 5.

En este *sprint* se realizó la implementación de un modelo matemático 2D, que se entiende como una extensión de lo que se había logrado hasta entonces en 1D a un dominio generalizado. Inicialmente no se sabía si esto implicaría demasiados cambios sobre los algoritmos generados, sin embargo, se esperaba que, al haber hecho las etapas anteriores con la idea de alcanzar simulaciones válidas a nivel físico químico en 2D, no existiría un problema en generalizar a 2D. Aun así, se optó por estimar el esfuerzo de esta tarea de forma conservadora por posibles problemas que surgieran en el transcurso de este u otros *sprints*.

Por otro lado, una vez implementado el modelo matemático para dominios bidimensionales, se procedería a realizar pruebas de compatibilidad hacia atrás con dominios unidimensionales. Esto, implicaba comprobar si los cambios realizados sobre los algoritmos para generalizar su comportamiento a dominios bidimensionales, alteraban de forma apreciable o no los resultados obtenidos para los dominios unidimensionales en etapas y *sprints* previos. Si existían cambios significativos en los resultados, entonces debía reverse la implementación del modelo y realizar ajustes de generalización ya que debía existir compatibilidad hacia atrás.

En este *sprint* se realizaron las mismas consideraciones sobre la granularidad de tareas y asignación de tiempos/esfuerzo del *sprint* 3.

Ejecución

La ejecución definitiva del *sprint* en cuestión, se dio como se ilustra en la **Tabla 6-23**.

Desde	Hasta	Tarea	SP por tarea	SP por iteración
22/04/20	15/05/20	Implementación modelo matemático 2D inicial.	6	7
15/05/20	19/05/20	Pruebas y ajustes de compatibilidad hacia atrás con dominios 1D	1	

Tabla 6-23. Ejecución del *sprint* 5.

Como se aprecia en la **Tabla 6-23**, el tiempo en el cual se llevó a cabo las tareas de este *sprint* resultaron ser similares a lo que se estimó inicialmente durante la planificación de los *sprints*, incluso menores en esfuerzo por 1 SP. Esto se debió a que se cumplió la expectativa de que, al haber llevado durante todo el proyecto el objetivo de obtener resultados válidos en 2D, cuando se debiese implementar el modelo y algoritmo bidimensionales se encontraría cierta facilidad en hacerlo.

Al finalizar el *sprint* 4 se actualizó el *product backlog* inicial para eliminar las tareas ya realizadas. El resultado se observa en la **Tabla 6-24**.

ID	Tarea	Prioridad de desarrollo	Estimación esfuerzo (<i>story points</i>)
7	Lograr convergencia temporal para dominios 2D.	Alta	3
8	Lograr convergencia de malla para dominios 2D.	Alta	4
9	Pruebas/validación del caso 2D con casos de bibliografía.	Alta	3
10	Generación de la documentación.	Alta	8
		TOTAL	50

Tabla 6-24. *Product backlog* actualizado luego del *sprint* 5.

Gestión de Riesgos

En este *sprint* no se detectó la activación de ningún riesgo. Sin embargo, es importante mencionar que la segunda tarea de este *sprint*, fue establecida por la posibilidad de la activación riesgo R20 - “Algoritmos desarrollados no se aplican a todos los casos posibles” como consecuencia de los desarrollos de la primera tarea.

Sprint 6

Planificación

Este *sprint* consistió de las tareas que se muestran en la **Tabla 6-25**, donde se indica, también, la estimación de tiempos y esfuerzos.

Desde	Hasta	Tarea	SP por tarea	SP por iteración
09/04/20	20/04/20	Desarrollo algoritmos para lograr convergencia temporal para dominios 2D.	3	3
		Pruebas convergencia temporal sobre dominios 2D.		

Tabla 6-25. Planificación del *sprint* 6.

Las tareas asignadas a este *sprint* tenían como objetivo lograr la convergencia temporal en las simulaciones para dominios 2D (concepto explicado en la sección 6 del capítulo II y estudiado en el capítulo VII).

La situación existente en cuanto a la asignación de tiempo y esfuerzo para las tareas (relacionadas entre sí), y la granularidad de estas, es análoga a la observada en el *sprint* 3.

Este *sprint* podría haber resultado innecesario si el modelo implementado inicialmente para dominios bidimensionales hubiese contado con una convergencia de malla aceptable.

Ejecución

Los tiempos y esfuerzos destinados a completar el *sprint* 6 son los que se muestran en la **Tabla 6-26**.

Desde	Hasta	Tarea	SP por tarea	SP por iteración
19/05/20	06/06/20	Desarrollo algoritmos para lograr convergencia temporal para dominios 2D.	5	5
		Pruebas convergencia temporal sobre dominios 2D.		

Tabla 6-26. Ejecución del *sprint* 6.

Para esta iteración se puede observar que fue requerido mayor tiempo y esfuerzo que el estimado originalmente. Esto se dio de esta forma porque fue necesario implementar una forma generalizada a dominios bidimensionales del algoritmo de control de avance de tiempo de la simulación (implementando una fórmula matemática más general desarrollada para ser válida en dominios bidimensionales). Esto requirió de un tiempo de desarrollo y pruebas adicionales que no podría haber sido previsto de antemano durante la planificación. Este algoritmo y su fundamento teórico se explican en el capítulo VI, en la sección 8.

Una vez finalizadas las tareas relativas al *sprint* se procedió a actualizar el *product backlog* del proyecto eliminando las tareas ya realizadas. El resultado se muestra en la **Tabla 6-27**.

ID	Tarea	Prioridad de desarrollo	Estimación esfuerzo (<i>story points</i>)
8	Lograr convergencia de malla para dominios 2D.	Alta	4
9	Pruebas/validación del caso 2D con casos de bibliografía.	Alta	3
10	Generación de la documentación.	Alta	8
		TOTAL	50 *

Tabla 6-27. *Product backlog* actualizado luego del *sprint* 6.

Gestión de Riesgos

A lo largo de esta iteración, el riesgo que se materializó fue R20. Sus detalles se pueden ver en la **Tabla 6-28**.

Ref.	Riesgo	Probabilidad	Impacto
R20	Algoritmos desarrollados no se aplican a todos los casos posibles.	Alta	Alto

Tabla 6-28. Riesgos activados durante el *sprint* 6.

El riesgo R20 se activó durante la primera tarea. En un principio se diseñó un algoritmo para conseguir convergencia temporal en dominios bidimensionales que no hacía uso de las herramientas provistas por OpenFOAM. Sin embargo, el mayor problema fue que los algoritmos generados no eran correctos, en el sentido de que no aplicaban a todos los casos posibles (R20), donde mallas con distintas características afectan la convergencia temporal (se recuerda que, como se mencionó en el *sprint* 4 y, como se observará en el capítulo VII, este es un problema con variables de tiempo y espacio acopladas, donde la convergencia de malla y de tiempo están relacionadas).

Debido a que el riesgo R20 fue catalogado con un nivel de “Riesgoso” o “Muy riesgoso”, debía contar con una estrategia a llevar a cabo si se materializaba. La respuesta especificada para este riesgo en el plan de gestión de riesgos fue la mostrada en la **Tabla 6-29**.

Ref.	Acción	Respuesta
R20	Reducir	Se pasará por múltiples etapas de validaciones considerando este riesgo. Se realizará una batería de pruebas para reducir la probabilidad.

Tabla 6-29. Respuestas especificadas en la planificación para los riesgos activados durante el *sprint* 6.

Para poder reducir el riesgo de R20, múltiples pruebas fueron realizadas. De esta forma, se encontró la falta de generalidad del algoritmo. Una vez advertido este hecho, se repitió la tarea de desarrollo (y luego de pruebas) del *sprint* con un enfoque más general y a la vez más sencillo (a nivel de código) en mente, que hacía uso de las herramientas que OpenFOAM proporciona, por lo que incluso estaba más optimizado.

Sprint 7

Planificación

Las tareas asignadas a este *sprint* durante la planificación se muestran en la **Tabla 6-30**.

Desde	Hasta	Tarea	SP por tarea	SP por iteración
20/04/20	04/05/20	Desarrollo algoritmos para lograr convergencia de malla para dominios 2D.	4	4
		Pruebas convergencia de malla sobre 2D		

Tabla 6-30. Planificación del *sprint* 7.

Con este *sprint* se pretendía hacer modificaciones necesarias sobre el conjunto de algoritmos desarrollados hasta el momento para que, luego de probarlo satisfactoriamente, se pudiese concluir que las simulaciones presentaban convergencia de malla.

Nuevamente, como en los demás *sprints* enfocados en desarrollo y pruebas, se presenta la situación mencionada en el *sprint* 3 sobre la granularidad de tareas y la asignación de tiempos y esfuerzos.

Cómo en otros casos previos, existía la posibilidad de que este *sprint* hubiese sido innecesario si en el modelo implementado en dominios bidimensionales hubiese existido una convergencia de malla aceptable.

Ejecución

Para este *sprint* fue necesario destinar el tiempo y esfuerzo mostrado en la **Tabla 6-31**.

Desde	Hasta	Tarea	SP por tarea	SP por iteración
06/06/20	20/06/20	Desarrollo algoritmos para lograr convergencia de malla para dominios 2D.	4	4
		Pruebas convergencia de malla sobre 2D		

Tabla 6-31. Ejecución del *sprint* 7. La primera tarea se encuentra tachada debido a que no fue necesario llevarla a cabo.

Como se muestra en la **Tabla 6-31**, para esta iteración se requirió el esfuerzo que se había planificado originalmente, aunque las tareas fueron realizadas en un periodo de tiempo distinto al planificado por las demoras originadas en *sprints* previos.

Un detalle muy importante que puede notarse es que la primera tarea no fue necesario realizarla. Esto fue así porque la convergencia de malla fue alcanzada para dominios unidimensionales y se esperaba que dicha convergencia no sufriera cambios (debido a la implementación y mayormente a las pruebas realizadas en el *sprint* 5). Luego, las pruebas de este *sprint* terminaron por confirmar esta hipótesis. En este *sprint* se hicieron numerosas pruebas usando mallas con topologías totalmente distintas y con cantidades de celdas muy distintas, obteniendo resultados muy similares en todos los casos. Este conjunto de pruebas y el análisis de resultados pueden verse en el capítulo VII.

Cuando se dio por finalizado este *sprint* se procedió a actualizar el *product backlog* removiendo las tareas que ya se habían terminado. El resultado se muestra en la **Tabla 6-32**.

ID	Tarea	Prioridad de desarrollo	Estimación esfuerzo (<i>story points</i>)
9	Pruebas/validación del caso 2D con casos de bibliografía.	Alta	3
10	Generación de la documentación.	Alta	8
		TOTAL	50 *

Tabla 6-32. *Product backlog* actualizado luego del *sprint* 7.

Gestión de Riesgos

Si bien este *sprint* duró exactamente lo especificado en la planificación inicial y no fue necesario llevar a cabo la tarea de desarrollo (por motivos explicados en párrafos anteriores), aun así, se dio la activación de determinados riesgos, que evitaron que la duración del *sprint* fuese menor a lo estimado. Durante este *sprint* los riesgos que se activaron fueron R10 y R14. Los detalles de dichos riesgos se muestran en la **Tabla 6-33**.

Ref.	Riesgo	Probabilidad	Impacto
R10	Impacto de obligaciones laborales y académicas del alumno investigador.	Media	Medio
R14	Las validaciones al final de cada <i>sprint</i> demoran más de lo esperado.	Media	Medio

Tabla 6-33. Riesgos activados durante el *sprint* 7.

Si bien hasta este punto del desarrollo del proyecto ambos miembros del equipo de desarrollo estuvieron realizando pasantías en distintos lugares, a partir de este *sprint*, uno de los miembros empezó a trabajar tiempo completo (9 hs.). Esto, significó un impacto por obligaciones laborales al alumno investigador (R10) que redujo su capacidad de cumplir las cuotas horarias semanales dedicadas al avance del proyecto.

Por otro lado, en esta etapa fue necesario realizar una gran cantidad de pruebas con mallas distintas para corroborar que tan bien la convergencia de malla se mantenía en dominios bidimensionales. Estas pruebas tenían la característica de poseer un número muy grande de celdas en el dominio simulado (porque de otra forma la simulación en 2D tendría demasiado error, sería demasiado imprecisa) lo que se tradujo a tiempos de simulación excesivamente largos (llegando incluso a 3 días). Solo una vez terminadas las simulaciones se podía saber que tan bien funcionaba el algoritmo en dicha malla. Esto dio lugar a la activación del riesgo R14.

Los riesgos R10 y R14 fueron catalogados con un nivel de “Riesgoso” o “Muy riesgoso”, por lo tanto, en el plan de gestión de riesgos especificado originalmente, debía especificarse una estrategia a llevar a cabo si se materializaban. Las respuestas especificadas fueron las mostradas en la **Tabla 6-34**.

Ref.	Acción	Respuesta
R10	Asumir	La carga horaria semanal ya fue diseñada para que se tenga en cuenta estos imprevistos personales.
R14	Asumir	Se determinó tiempos de validación que consideran este riesgo, por lo que, si se materializa, será de un impacto menor.

Tabla 6-34. Respuestas especificadas en la planificación para los riesgos activados durante el *sprint* 7.

En el caso de la activación del riesgo R10, evidentemente se subestimó el impacto de contraer obligaciones laborales al momento de establecer una respuesta adecuada. Sin embargo, en este *sprint*, el riesgo individualmente no tuvo como efecto un retraso mayor por lo que no fue necesario adoptar acciones correctivas adicionales.

Para el riesgo R14 el impacto en este *sprint* también fue menor, como se esperaba a partir de adoptar la respuesta especificada en el plan de gestión de riesgos.

La activación de los riesgos mencionados no tuvo tanto impacto individualmente en cuanto a tiempo, pero sumados generó una extensión de la duración de las tareas que no era menor. Afortunadamente, la duración del *sprint* se mantuvo acorde a lo estimado inicialmente.

4. Etapa de pruebas y experimentación con casos 2D de la bibliografía

Sprint 8

Planificación

En este sprint las tareas que serían llevadas a cabo se listan en la **Tabla 6-35** junto con la estimación inicial de tiempo y esfuerzo.

Desde	Hasta	Tarea	SP por tarea	SP por iteración
04/05/20	14/05/20	Estudio de la bibliografía relacionada a casos 2D.	3	3
		Pruebas del caso 2D para recrear resultados de bibliografía.		

Tabla 6-35. Planificación del *sprint* 8.

En esta última iteración el objetivo era intentar recrear, a través del uso de la simulación desarrollada a lo largo del proyecto, los resultados presentados en un artículo científico específico donde fueron expuestos resultados de experimentos reales (no simulados) sobre un dominio aproximadamente bidimensional (ultrafino) [33]. Para poder hacerlo, primero era requerido un estudio previo sobre la bibliografía en cuestión. Con este estudio se pretendía obtener las condiciones fisicoquímicas de realización de los experimentos (y de sus resultados). Estos datos, se iban a cargar como parámetros de la simulación para poder comparar sus resultados con el experimento real, a fin de determinar la calidad del modelo y los algoritmos producidos.

En este *sprint*, se decidió que la mejor manera de asignar tiempo y esfuerzo a las tareas sería considerarlas como una sola, ya que ambas serían realizadas casi paralelamente.

Ejecución

La ejecución de este último sprint se llevó a cabo durante el periodo y con la asignación de esfuerzos en SP mostrada en la **Tabla 6-36**.

Desde	Hasta	Tarea	SP por tarea	SP por iteración
20/06/20	30/07/20	Estudio de la bibliografía relacionada a casos 2D.	12	12
		Pruebas del caso 2D para recrear resultados de bibliografía.		

Tabla 6-36. Ejecución del *sprint* 8.

Como se puede observar, los tiempos y esfuerzos estimados inicialmente para esta iteración resultaron estar muy por debajo de lo que realmente fue requerido. Esto se debió (además de lo explicado en la sección de gestión de riesgos) al hecho de que, inicialmente, se creyó que la bibliografía a utilizar disponía de suficiente información como para recrear los experimentos allí expuestos. Sin embargo, una vez alcanzada esta etapa se descubrió que existía mucha información del tipo empírica relacionada al procedimiento con el cual se llevó a cabo el experimento, pero existía muy poca información relacionada a las características de los resultados que pudiesen ser usada en los algoritmos desarrollados (y por consiguiente en los modelos matemáticos). Esta información, aunque no es difícil de obtener cuando se hace el experimento químico, no fue central al momento de realización del artículo científico utilizado, por lo que no se contaba con suficiente información para hacer pruebas directas con valores de este como se esperaba inicialmente.

Para resolver el problema de la falta de información, que limitaba la capacidad de asignar valores a los parámetros, se recurrió a utilizar los valores que eran conocidos a partir del artículo y, luego, a aproximar los demás de forma empírica (numérica) buscando obtener los resultados más próximos posibles (sin modificar aquellos valores que sí se pudieron extraer).

Al existir varios parámetros para los cuales no se tenía indicios de qué valores usar, existían muchas combinaciones posibles para probar buscando resultados satisfactorios. Esto, dio lugar a realizar una gran cantidad de pruebas para poder conseguir los resultados deseados, y fue así como la duración del *sprint* se extendió mucho más de lo anticipado durante la planificación.

Una vez finalizadas las tareas de este último *sprint*, se actualizó el *product backlog*, resultando como se muestra en la **Tabla 6-37**.

ID	Tarea	Prioridad de desarrollo	Estimación esfuerzo (<i>story points</i>)
10	Generación de la documentación.	Alta	8
		TOTAL	50

Tabla 6-37. *Product backlog* actualizado luego del *sprint* 8.

Gestión de Riesgos

Durante esta iteración, los riesgos que se materializaron fueron R10, R14, R15 y R22. Los detalles de estos riesgos se muestran en la **Tabla 6-38**.

Ref.	Riesgo	Probabilidad	Impacto
R10	Impacto de obligaciones laborales y académicas del alumno investigador.	Media	Medio
R14	Las validaciones al final de cada <i>sprint</i> demoran más de lo esperado.	Media	Medio
R15	Las actividades de investigación se extienden por dificultad de comprensión del material.	Media	Alto
R22	Imposibilidad de obtener artículos científicos y libros en la etapa de investigación.	Medio	Alto

Tabla 6-38. Riesgos activados durante el *sprint* 8.

Similarmente al *sprint* anterior, el riesgo R10 se activó durante este *sprint*. La activación se produjo en ambas tareas del *sprint*. El impacto de las obligaciones laborales de uno de los alumnos afectó notablemente los tiempos requeridos para llevar a cabo las tareas.

Por otro lado, el riesgo R22 se activó durante la primera tarea. Este riesgo se debió a que, en el artículo científico que contenía los experimentos de contraste, faltaban datos requeridos y/o estaban indicados parcialmente. Estos datos, eran necesarios para configurar los parámetros de la simulación, y no fue posible obtenerlos a partir de otros artículos científicos y/o libros, impidiendo conocer las condiciones específicas de los experimentos presentados.

Para el caso de los parámetros faltantes, fueron usados dos enfoques distintos. En primer lugar, se intentó estimar dichos parámetros a partir de la información que si estaba disponible. La dificultad presente en hacer esto activó el riesgo R15.

En segundo lugar, se procedió a ajustar los parámetros desconocidos de forma empírica a través de numerosas simulaciones, buscando aproximar los resultados lo más posible haciendo uso de fundamentos teóricos sobre el comportamiento del fenómeno simulado. Estas actividades empíricas, extendieron las validaciones del *sprint* activando el riesgo R14.

Dado que los riesgos R14 y R15 tenían asignado un nivel de riesgo de “Riesgoso” o “Muy riesgoso”, se debía contar con una estrategia a llevar a cabo si se materializaban. Las respuestas específicas para estos riesgos en el plan de gestión de riesgos fueron la mostrada en la **Tabla 6-39**.

Ref.	Acción	Respuesta
R10	Asumir	La carga horaria semanal ya fue diseñada para que se tenga en cuenta estos imprevistos personales.
R14	Asumir	Se determinó tiempos de validación que consideran este riesgo, por lo que, si se materializa, será de un impacto menor.
R15	Reducir	Ya se obtuvo y estudió material inicial relacionado al proyecto de investigación. Se considera que el acceso a material nuevo se puede realizar de manera progresiva para facilitar la comprensión. Además, el director del proyecto tiene experiencia en la temática a tratar.
R22	Reducir	Se cuenta con la experiencia del director propuesto y con contactos involucrados en el tema.

Tabla 6-39. Respuestas especificadas en la planificación para los riesgos activados durante el *sprint* 8.

Con la activación del riesgo R10, no pudieron adoptarse acciones correctivas durante la iteración porque simplemente se experimentó una reducción en la cuota horaria diaria (y en consecuencia semanal) de uno de los alumnos involucrados. La respuesta especificada en la planificación inicial mostró no ser suficiente.

En cuanto a los riesgos R14 y R15, no hubo demasiado que pudiese hacerse para evitar el impacto de los riesgos. En el primer caso, claramente en este *sprint*, el tiempo de validación considerado inicialmente no fue suficiente dado que no se tuvo en cuenta el problema encontrado con los datos faltantes en el artículo, por lo que la única opción posible fue la de extender la duración del *sprint*. Para el riesgo R15, lo único que se pudo hacer frente a la activación, fue continuar el estudio de la bibliografía disponible con apoyo del conocimiento del director y codirector del proyecto, extendiendo en consecuencia la duración del *sprint*.

Finalmente, dada la activación del riesgo R22, se procedió a consultar con el director y codirector propuestos para realizar una estimación (a partir de los datos proporcionados en el artículo) de los valores a utilizar en las simulaciones que no estaban claros. Con esto se logró realizar una serie de pruebas mejor encaminada hacia resultados que fuesen acordes a lo mostrado en el artículo.

5. Generación de la documentación

Planificación

La planificación inicial para esta tarea fue la que se muestra en la **Tabla 6-40**.

Desde	Hasta	Tarea	SP por tarea	SP por iteración
14/05/20	14/06/20	Generación de la documentación	8	

Tabla 6-40. Planificación para la generación de documentación.

En esta tarea se pretendía generar la documentación presente, solicitada para ser entregada al momento de finalización del proyecto.

Ejecución

La generación de la documentación requirió de los tiempos y el esfuerzo indicado en la **Tabla 6-41**.

Desde	Hasta	Tarea	SP por tarea	SP por iteración
30/07/20	12/10/20	Generación de la documentación	21	

Tabla 6-41. Ejecución de la generación de documentación.

Siguiendo lo que se muestra en la tabla, se puede decir que el esfuerzo y tiempo requerido para completar la presente documentación fue ampliamente subestimado, llegando a casi triplicar las cantidades originalmente estimadas.

En esta tarea, se volvió a evidenciar el efecto de las obligaciones laborales contraídas por los alumnos (en especial uno de ellos). Este hecho tuvo un efecto muy importante en los tiempos y esfuerzos destinados a la producción de documentación.

Por otro lado, la temática tratada en el proyecto y su inherente complejidad, hicieron necesario que en la documentación sea explicado cuidadosamente, y en un nivel de detalle considerable, el funcionamiento del modelo y algoritmos desarrollados.

Gestión de Riesgos

Como se mencionó antes, durante la generación de la documentación, el riesgo que se activó (al igual que en los *sprint* 7 y 8) fue el R10. Los detalles de dicho riesgo se muestran en la **Tabla 6-42**.

Ref.	Riesgo	Probabilidad	Impacto
R10	Impacto de obligaciones laborales y académicas del alumno investigador.	Media	Medio

Tabla 6-42. Riesgos activados durante la generación de la documentación.

El hecho de que uno de los integrantes de este proyecto comenzará a trabajar a tiempo completo (9 hs), causó una gran reducción en la capacidad de cumplir las cuotas horarias semanales, dedicadas a la generación de la documentación.

Frente al riesgo R10, en el plan de gestión de riesgos especificado originalmente, se especificó la respuesta mostrada en la **Tabla 6-43**.

Ref.	Acción	Respuesta
R10	Asumir	La carga horaria semanal ya fue diseñada para que se tenga en cuenta estos imprevistos personales.

Tabla 6-43. Respuestas especificadas en la planificación para los riesgos activados durante la generación de la documentación.

Frente a la activación del riesgo R10, claramente se subestimó el impacto de las obligaciones laborales al establecer una respuesta adecuada. Al trabajar sobre la documentación, la diferencia en disponibilidad se hizo muy evidente, alargando los tiempos necesarios en gran medida.

6. Riesgos no específicos a una iteración.

Se considera oportuno mencionar y explicar riesgos que estuvieron activos durante todo el desarrollo de este proyecto y que no pueden ser enmarcados dentro de un solo *sprint*. Los riesgos en cuestión son:

- **R8 - “Falta de conocimiento teórico sobre la problemática para desarrollar la aplicación”:** Si bien este riesgo estuvo activo mayormente en los primeros *sprints*, fue una cuestión de importancia durante todo el desarrollo. Debido a las características del proyecto y a la base de conocimiento con las cuales contaban los alumnos al momento de comenzar el proyecto, existió una curva de aprendizaje muy alargada que generó que durante el desarrollo se requiriese mucho estudio e investigación para adquirir los conceptos relevantes al problema a resolver. Esto se dio por la activación del riesgo. Frente a este riesgo, la acción tomada fue la de continuar las tareas de investigación el tiempo suficiente para obtener la información necesaria para seguir o consultar con el director y codirector.

- R18 - “Cambios en las tareas en una iteración”: Como se explicó en la sección 1.A, la planificación original realizada respecto al proyecto sufrió cambios previos al comienzo de actividades, que modificaron las tareas de los *sprints* a llevar a cabo, produciendo así la activación del riesgo. Como este riesgo se activó en etapas muy tempranas del proyecto fue posible revisar y corregir la planificación de tareas, manteniendo el tiempo total del proyecto originalmente estimado.
- Aislamiento Social Preventivo y Obligatorio: Debido a la pandemia relacionada al Covid-19 que sufrió y sufre el mundo durante el periodo 2019-2020 y las medidas de aislamiento, el avance del proyecto encontró problemas al momento de realizar encuentros con el director y codirector del proyecto, así como también entre los alumnos involucrados. Esto afortunadamente se pudo solventar en mayor parte a través del uso de herramientas/servicios para realizar video-llamadas, pero la imposibilidad de realizar reuniones presenciales marcó una diferencia en el progreso de todas maneras. Este riesgo no fue considerado durante ninguna etapa de planificación previa al inicio del proyecto debido a la impredecibilidad característica de la situación.

VI. Arquitectura de la aplicación

1. Estructura y archivos de casos simulados

Los casos de simulaciones deben contar de una estructura de directorios específica para que puedan ser ejecutados. En la **Fig. 7-1** se puede observar esta estructura en un caso de simulación.

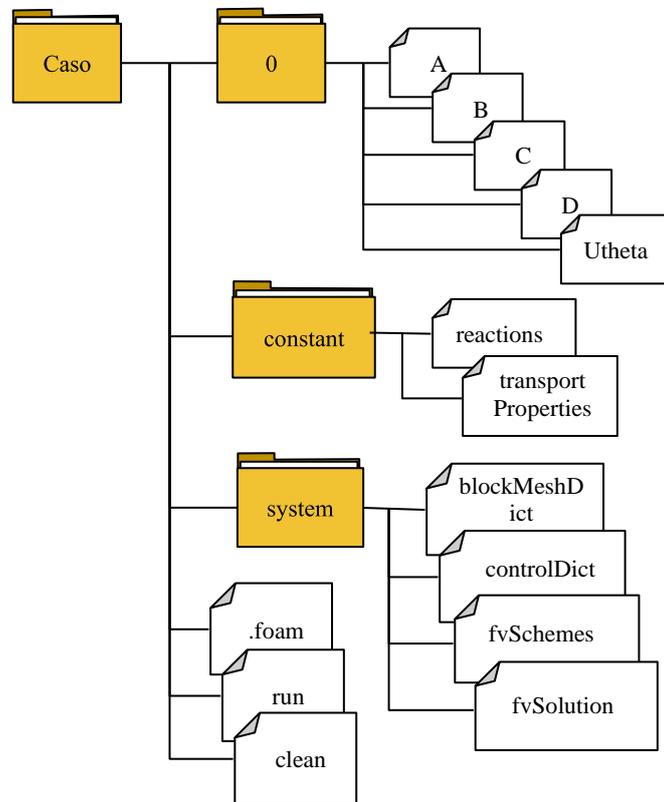


Fig. 7-1. Carpeta estándar de un caso

Los archivos que se encuentran en la carpeta principal (*.foam*, *run*, *clean*) son de carácter auxiliar. En su interior se encuentran simples comandos shell (a excepción de *.foam* que está vacío). La función de estos, es la de permitir la visualización desde ParaView (*.foam*), la de ejecutar todos los comandos para llevar a cabo la simulación (*run*) y la de limpiar las salidas de la ejecución para volver a simular (*clean*).

Para los demás archivos, la forma de especificar la información en la mayoría de ellos, es decir, los parámetros de la simulación, sigue el estándar definido por OpenFOAM ([39], [40], [41], [42], [43]). Quedan exentos del estándar los parámetros que fueron añadidos para su uso en los algoritmos diseñados en el proyecto actual, o en el proyecto *porousMultiphaseFoam* usado como base (desde donde surge también el solucionador utilizado, *porousScalarTransportFoam*). Además, algunos archivos fueron modificados ampliamente o creados en dicho proyecto base.

En la carpeta *system*, se encuentran los archivos de configuración de la simulación:

- *blockMeshDict*: Archivo de entrada para la herramienta *blockMesh*. A partir de la información en este archivo se genera la malla del dominio.

De forma general, en este archivo se especifican vértices (siempre en 3 dimensiones) y bloques con estos vértices (estos pueden tener distintas formas y discretizaciones en celdas). Además, se detallan los bordes del dominio, definiendo cuales son y si van a ser simulados o no. A partir de todo esto, la simulación será 1D, 2D o 3D según corresponda.

- *controlDict*: En este archivo no se definen los parámetros específicos a algún objeto de la simulación, sino los que están relacionados al control de esta, como pasos de tiempo, control de escritura, entre otros.
- *fvSchemes*: Definición de qué algoritmos matemáticos se van a utilizar para las distintas operaciones matemáticas relacionadas a los procesos de discretización de los operadores diferenciales. Por ejemplo, en la resolución de ecuaciones en derivadas parciales, algunos algoritmos brindan menos precisión a cambio de menor costo computacional o viceversa.
- *fvSolution*: Contiene parámetros de índole numérica para el solucionador a ser utilizado junto a la selección de algoritmos internos. Se especifican algunas constantes como, por ejemplo, las tolerancias admitidas para ciertos algoritmos iterativos.

En la carpeta *constant* se definen las propiedades fisicoquímicas de las especies involucradas en la simulación:

- *reactions*: Este archivo es específico del proyecto *porousMultiphaseFoam* (se originó en el contexto de su desarrollo). Con los datos definidos aquí, se realiza la selección del modelo de reacciones y se especifican las distintas reacciones que ocurren en la simulación. En estas reacciones, se define la ecuación de la reacción propiamente dicha (de donde se obtiene el orden de la reacción), las especies involucradas, los valores de las constantes de reacción y cualquier otra variable específica del modelo de reacción.

En nuestro caso, el modelo de reacciones usado es *liesegangReaction*, que fue desarrollado en el proyecto para la simulación de los patrones de Liesegang. Las ecuaciones de las reacciones son tomadas del artículo de Lagzi, mencionado en la sección 4 del capítulo II.

- *transportProperties*: En este archivo se definen los valores de difusividades de las distintas especies y otros parámetros que definen la interacción entre las especies, los fluidos y el material poroso. Su estructura, si bien está definida en el estándar de OpenFOAM, cuenta con grandes modificaciones que surgen del proyecto *PorousMultiphaseFoam*

Finalmente, en la carpeta *0*, se especifican las condiciones de las especies al tiempo cero de la simulación, en otras palabras, las condiciones iniciales. Se especifican las unidades/dimensiones de la especie, la distribución en el dominio y las condiciones de borde del dominio para cada una.

A medida que se ejecuta la simulación, nuevas carpetas se crearán a medida que el tiempo avance. Estas carpetas tendrán como nombre el tiempo de la simulación en el cual fueron creadas. Dentro de estas carpetas se podrá encontrar archivos correspondientes a cada especie, que indican las concentraciones de estas en cada celda del dominio (en otras palabras, el estado del dominio), en ese punto en el tiempo de la simulación.

El archivo *Utheta* corresponde al campo de velocidad de la fase theta, que en términos químicos es un solvente (las especies constituyen los solutos). Este es necesario para el funcionamiento del solucionador *porousScalarTransportFoam*, pero en este proyecto se asume que el campo *Utheta* es nulo dado que no se consideraron efectos advectivos en las simulaciones (como se explicó en el capítulo II). Por este motivo, el archivo correspondiente a este campo sólo existe en la carpeta de tiempo inicial (carpeta 0 a menos que se decida comenzar desde otro tiempo).

2. Parámetros de una simulación

Para poder ejecutar las simulaciones, además de los algoritmos desarrollados, se deben especificar los parámetros adecuados en los archivos correspondientes y en la forma estandarizada por OpenFOAM, o la forma esperada por la implementación realizada en este proyecto.

Por otro lado, los parámetros, se pueden dividir en tres grupos conceptuales, aquellos que son de relevancia físico-química (por ejemplo, pertinentes a las propiedades de las sustancias), parámetros de control de la simulación y aquellos relacionados a la forma de expresar las salidas. A dichos parámetros, se los ubica en diferentes archivos de acuerdo a que representan y en qué grupo se los ubica.

En la **Tabla 7-1** se puede observar una lista de los parámetros de interés utilizados en este proyecto, especificando la función que cumplen y en qué archivo se los debe especificar.

Parámetro/Variable [Unidades ⁷]	Archivo	Observaciones
Parámetros heredados de las herramientas usadas		
Concentración especie A [mol/m^3] ⁸	A	Concentración de la especie A. Se puede especificar en distintas partes del dominio. No tiene nombre de variable.
Concentración especie B [mol/m^3]	B	Concentración de la especie B. Se puede especificar en distintas partes del dominio. No tiene nombre de variable.
Concentración especie C [mol/m^3]	C	Concentración de la especie C. Se puede especificar en distintas partes del dominio. No tiene nombre de variable.
Concentración especie D [mol/m^3]	D	Concentración de la especie D. Se puede especificar en distintas partes del dominio. No tiene nombre de variable.
kf [$molar/seg$] (reacción AB, k)	reactions	Constante que rige la velocidad de la reacción $A + B \rightarrow C$. Debe asignarse a la reacción correspondiente en el archivo.
kf [$1/seg$] (reacción CD, k_1)	reactions	Constante que rige la velocidad de coagulación. k_1 de ecuación (2.6) y (2.7) (capítulo II). Debe asignarse a la reacción correspondiente en el archivo.
kf [$molar/seg$] (reacción CD, k_2)	reactions	Constante que rige la velocidad de formación de precipitado. k_2 de ecuación (2.6) y (2.7) (capítulo II). Debe asignarse a la reacción correspondiente en el archivo.
Dm [m^2/seg] (especie A)	transportProperties	Valor de difusividad de la especie A. Debe asignarse a la especie correspondiente en el archivo.
Dm [m^2/seg] (especie B)	transportProperties	Valor de difusividad de la especie B. Debe asignarse a la especie correspondiente en el archivo.
Dm [m^2/seg] (especie C)	transportProperties	Valor de difusividad de la especie C. Debe asignarse a la especie correspondiente en el archivo.
Dm [m^2/seg] (especie D)	transportProperties	Valor de difusividad de la especie D. Debe asignarse a la especie correspondiente en el archivo.
startTime [seg]	controlDict	Tiempo de inicio de la simulación.
endTime [seg]	controlDict	Tiempo de fin de la simulación
deltaT [seg]	controlDict	Paso de tiempo inicial para la simulación.
maxDeltaT [seg]	controlDict	Mayor paso de tiempo posiblemente usado en la simulación.
writeInterval [seg]	controlDict	Controla el intervalo de tiempo entre escrituras de archivos y carpetas de salida (escritura estado), con la información de dominio. Son los archivos que se cargan y visualizan en ParaView. Las carpetas tienen de nombre el tiempo de la simulación al momento de la escritura.
Extensión del dominio [m]	blockMeshDict	Extensión del dominio simulado. Compuesto por un conjunto de coordenadas de puntos. No tiene nombre de variable.

⁷ Unidades del SI de unidades. La unidad no siempre se debe escribir en el archivo. En los casos que se hace, no se escribe directamente. Remitirse a ejemplos de Anexo E para ver cómo se especifica.

⁸ Unidad equivalente a milimolar (mM).

Cantidad de celdas	blockMeshDict	Número de celdas en el dominio. Compuesto por uno o más valores (dependiendo de las dimensiones del dominio). No tiene nombre de variable.
Relación de expansión de celdas	blockMeshDict	Valor que establece una relación de aumento de tamaño de celdas a medida que se las recorre a lo largo de algún eje particular. No tiene nombre de variable.
Parámetros introducidos en este proyecto		
cradius [<i>m</i>]	reactions	Este valor, establece un radio de cristalización, es decir una distancia radial desde el centro de una celda de chequeo de saturación.
rho [<i>molar</i>]	reactions	Este valor especifica la concentración máxima que puede alcanzar la especie D en las bandas.
cs [<i>molar</i>]	reactions	Punto de floculación. Concentración que debe alcanzar la especie C para producirse precipitación en forma de especie D.
variationMax [%]	controlDict	Variación máxima admitida de la especie D entre puntos en el tiempo.
kValueTauD	controlDict	Valor de la constante k para fórmula de ajuste de tiempo.
refValA [<i>molar</i>]	controlDict	Especifica la concentración de referencia de la especie A a ser usada en la fórmula de ajuste de tiempo.
refValB [<i>molar</i>]	controlDict	Especifica la concentración de referencia de la especie B a ser usada en la fórmula de ajuste de tiempo.
refValC [<i>molar</i>]	controlDict	Especifica la concentración de referencia de la especie C a ser usada en la fórmula de ajuste de tiempo.

Tabla 7-1. Parámetros usados en las simulaciones de este proyecto.

Como se puede observar en la **Tabla 7-1**, muchos valores de parámetros no tienen un nombre de variable específico. Esto significa que en los archivos está simplemente indicado con un valor numérico en un lugar determinado.

Por otro lado, en la **Tabla 7-1**, algunos parámetros tienen el mismo nombre de variable, esto es así porque dentro del archivo correspondiente se especifica su valor para distintas especies o reacciones, por lo que se debe asignar el mismo para cada entidad.

Los parámetros refValA, refValB y refValC son completamente opcionales. De no ser especificados, se utilizarán valores por defecto. El único efecto será que el tiempo de la simulación podría avanzar más lentamente.

La funcionalidad de los parámetros/variables será clarificada en las siguientes secciones de este capítulo y en el capítulo VII.

En el Anexo E de este informe, se puede encontrar especificado el contenido de los archivos para un caso sencillo de ejemplo, en un dominio bidimensional.

3. Estructura general de la aplicación

La estructura de la aplicación desarrollada durante el proyecto estuvo dividida en dos partes mayores. Por un lado, el modelo de reacción que es donde se realizan los cálculos pertinentes al modelo de sol-coagulación, y por el otro, el solucionador, que es donde se realizan la carga de parámetros de archivos de configuración, inicialización del framework, resolución de ecuaciones diferenciales, generación de salidas y el control general del flujo del programa. En la **Fig. 7-2** se puede observar en detalle los procedimientos que componen cada parte de dicha estructura.

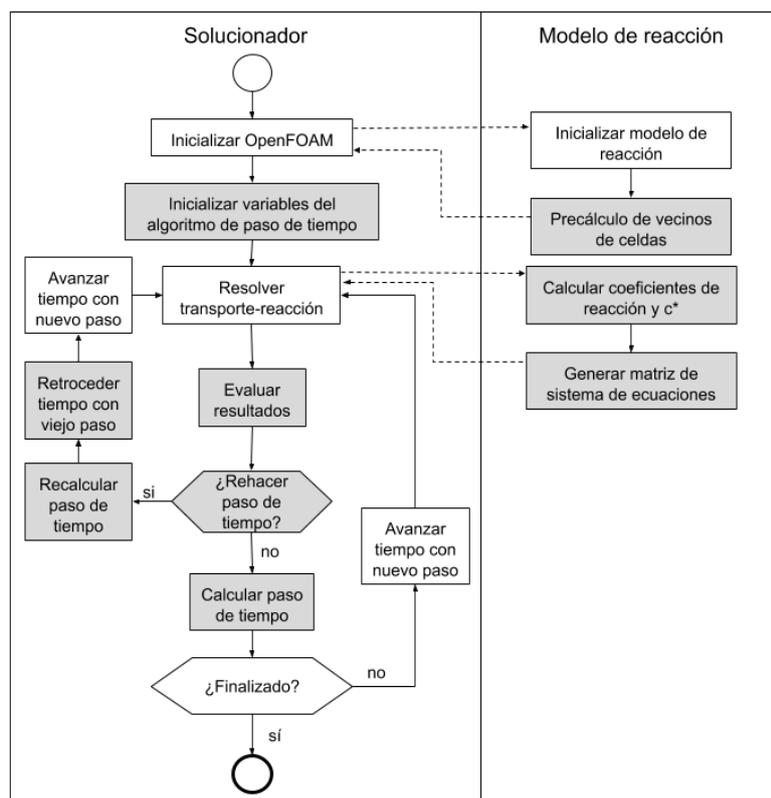


Fig. 7-2. Estructura de la aplicación.

En la Fig. 7-2 se distinguen algunos procedimientos en color gris y otros en color blanco. En blanco, se indican procedimientos heredados de lo existente en el solucionador original *porousScalarTransportFoam*. Indicado en gris, se identifican los procedimientos donde se desarrollaron nuevos algoritmos o se realizaron grandes modificaciones sobre lo existente en *porousScalarTransportFoam*.

En el recuadro del solucionador (izquierdo), se inicializan estructuras requeridas por OpenFOAM y se inicializan las variables relacionadas al algoritmo de paso de tiempo. Además, se resuelven los sistemas de ecuaciones involucrados en el cálculo de la solución para un tiempo de la simulación. Con los resultados, se determina si es necesario repetir un paso de tiempo o no, volviendo a resolver las ecuaciones en un tiempo dado (usando un paso de tiempo menor). Cualquiera sea el caso, se calcula un nuevo paso de tiempo y se lo usa para continuar la simulación (desde el tiempo actual o desde un tiempo anterior). Todo este proceso de control de tiempos está detallado en la sección 8 de este capítulo.

En el recuadro del modelo de reacción (derecho), se ubican los procedimientos relacionados al inicializado del modelo de reacción (tomando de los archivos correspondientes la información de las reacciones involucradas), el algoritmo de convergencia de malla (precálculo de vecinos de celdas) y el cálculo de los coeficientes para la resolución de las ecuaciones de reacción del modelo, mostradas en la sección 4 del capítulo II. Los algoritmos de precálculo de vecino y el cálculo de los coeficientes se explican en las secciones 5 y 7 de este capítulo. En la sección siguiente se pueden encontrar los atributos, métodos, relaciones y dependencias utilizados al implementar este modelo.

4. Diagrama de clases

En la Fig. 7-3, se puede observar el diagrama de clases correspondiente al desarrollo llevado a cabo durante este proyecto.

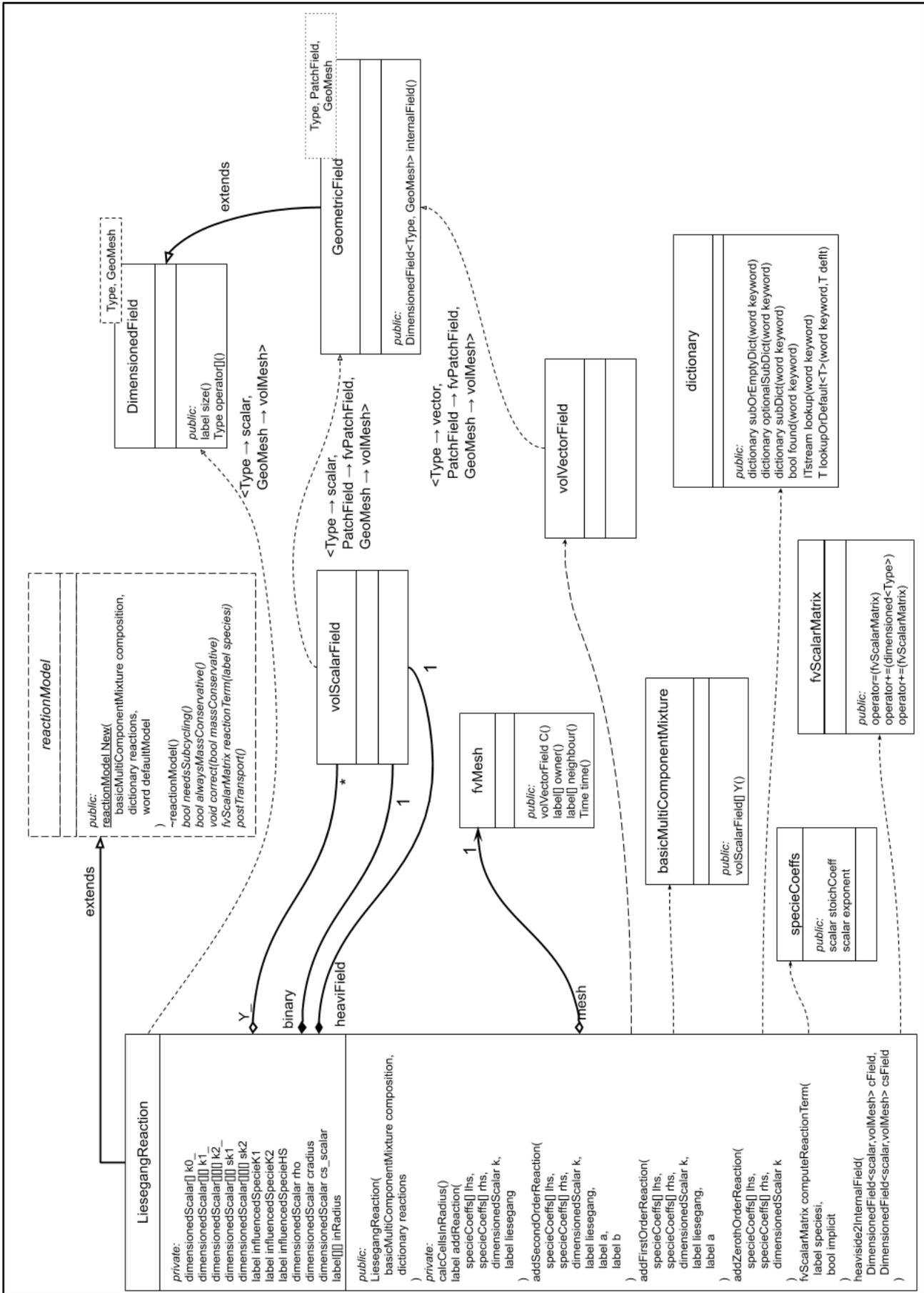


Fig. 7-3. Diagrama de clases que interactúan con el modelo de reacción LiesegangReaction.

Como se puede observar, no fue necesaria la definición de clases, ya que el desarrollo realizado estuvo constituido, en gran parte, por una serie de algoritmos autónomos no aislados. Las clases que se pueden ver en la **Fig. 7-3** son provistas por OpenFOAM, se utilizaron en los algoritmos matemáticos y están vinculadas por relaciones de dependencia con el modelo de reacción (LiesegangReaction).

Cabe destacar que el proyecto al ser desarrollado en C++, no es puramente orientado a objetos. Por este motivo, en la **Fig. 7-3**, no están presentes funciones creadas como las auxiliares en los cálculos matemáticos y (principalmente) la función *main*, que es donde se realizó gran parte del trabajo. El algoritmo que controla el paso de tiempo, es también un ejemplo de esta situación, dado que está incluido en su totalidad en la función *main*.

Se debe notar que esta forma de estructurar el código constituye la práctica esperable en aplicaciones de OpenFOAM. En este framework, la construcción de solucionadores usualmente emplea composición mediante directivas de inclusión de código en lugar de rutinas/funciones.

Por otro lado, es importante mencionar que muchos tipos de datos son heredados de las librerías de OpenFOAM. Es por esto que se encuentran tipos como *dimensionedScalar*, *label*, *volVectorField*, etc. en el diagrama de clases. Todos estos se encuentran definidos en la documentación de las API de OpenFOAM ([34], [35]) y se usan en los cálculos matemáticos involucrados en el funcionamiento de la aplicación.

Para ajustarse al estilo de diseño general del solucionador *porousScalarTransportFoam* usado como base, era necesario implementar una clase que extienda *reactionModel* y que implemente los métodos virtuales descritos en la **Fig. 7-3**. El objeto de dicha clase fue denominado *liesegangReaction* y es llamado en el inicio (instanciación) de la simulación y en cada iteración del solucionador. Este objeto está representado con el recuadro “Modelo de reacción” de la **Fig. 7-2** de la sección anterior.

En la instanciación del modelo es donde se utilizan los métodos privados *calcCellsInRadius* (detallado en la sección 5 de este capítulo), y todos los métodos “*add...Reaction*”, que son los que cargan los detalles de las reacciones involucradas. Estos detalles son leídos desde el archivo *reactions*.

De los métodos presentes en el modelo de reacción desarrollado, *correct* y *reactionTerm* destacan debido a las tareas que llevan a cabo. En el método *correct* se evalúan condiciones específicas a este modelo (ecuaciones (2.8) y (2.9)). Para esto, se calculan dos campos escalares, *binary* y *heaviField*. *binary* se utiliza para determinar (por cada celda) si se satisface la condición requerida para anular los términos en las ecuaciones (2.6) y (2.7), estableciendo k_1 y k_2 en cero. *heaviField* es utilizado para evaluar la activación de la reacción que transforma C a D generando un comportamiento catalizador en un radio de influencia desde una celda (ecuación (2.3)).

En el método *reactionTerm*, las constantes de reacción leídas del archivo *reactions* y los dos campos escalares mencionados son utilizados para generar una matriz representativa del sistema de ecuaciones de reacción que es retornada y utilizada por el solucionador para proporcionar una solución.

5. Precálculo de vecinos de celdas

El modelo de sol-coagulación modificado tomado como base (sección 4, capítulo II), al generalizarlo a dominios bidimensionales, realizaría la verificación de una sola celda vecina a los lados para la catalización del precipitado (como se puede ver para un caso 2D en la **Fig. 7-4**). Si bien este enfoque es correcto en el sentido que genera patrones de Liesegang, no es convergente en espacio ya que cualquier cambio en la discretización realizada por la malla (ya sea el tamaño de celda o la configuración de celdas) afecta el modelo matemático directamente y por ende la salida obtenida. Se necesitaba una forma de determinar la saturación del precipitado de celdas vecinas, pero sin usar el concepto de celdas, sino que usando una medida geométrica “universal”. Esta medida está basada en la formulación continua del problema y, en consecuencia, es independiente de la discretización (mallado).

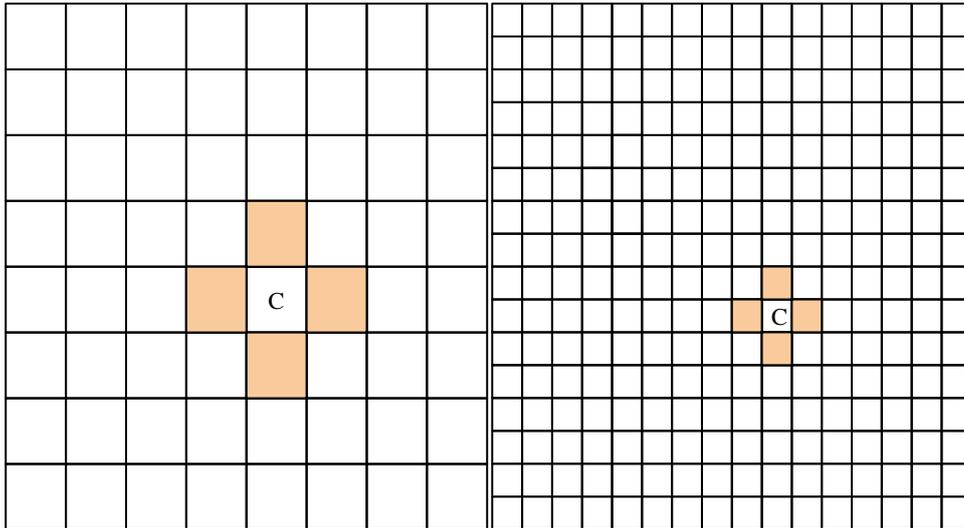


Fig. 7-4. Para un ejemplo 2D, se ve en naranja las celdas donde se cataliza la especie D. El algoritmo del modelo original es inherentemente no convergente, porque el espacio donde se cataliza es dependiente de la malla.

Como mejora al modelo y algoritmo original, se decidió agregar un parámetro geométrico con unidades de longitud denominado *cradius* (de *crystalization radius*). Este, especifica el radio desde el centro de una celda dada en el que se verificará la condición de saturación del precipitado (es decir, cuando la especie D llega al máximo posible para una celda dada).

El parámetro *cradius*, al ser independiente del mallado realizado para cada caso, prometía alcanzar convergencia en espacio. Para que esto fuese así, en cualquier caso ejecutado y para cada celda de la malla, se debía evaluar qué celdas de la vecindad estaban dentro de este radio, teniendo en cuenta que las celdas no necesariamente tienen todas el mismo tamaño.

En una dimensión, es factible y sencillo hacer esta evaluación dinámicamente durante la ejecución del caso ya que: (1) no hay tantas celdas en comparación con 2D, y (2) las celdas son enumeradas convenientemente en OpenFOAM (en el sentido que para la celda C_i , los vecinos directos son las celdas C_{i-1} y C_{i+1}) y con un ciclo es relativamente simple chequear las celdas en un radio definido.

Sin embargo, en dos dimensiones esta estrategia no es viable, porque para que la simulación tenga resultados razonables, la cantidad de celdas debe incrementarse considerablemente y esta verificación en tiempo de ejecución haría más lenta a la misma. Además, la relación entre celdas no es tan simple y constante, la forma de numerar las celdas depende de la malla utilizada en cada caso.

Utilizando las herramientas provistas por OpenFOAM, que facilitan la obtención de las celdas adyacentes a alguna otra celda, se realizó una estrategia de inundación implementada con una búsqueda en anchura (BFS - Breadth-first search) [44] de celdas partiendo desde la celda objetivo en un momento dado (se hace por cada celda del dominio). Esto puede verse representado en la **Fig. 7-5**.

Como se puede observar en la **Fig. 7-5**, la estrategia implementada determina las celdas vecinas a una celda dada, que se encuentran a una distancia (medida entre los centroides de las celdas) menor que el valor especificado por *cradius* (el valor determina así el horizonte de búsqueda). Estas celdas vecinas de la celda objetivo pueden ser contiguas o ser vecinas de vecinas.

La nueva estrategia es significativamente menos compleja, algorítmicamente hablando, que la búsqueda “naive” (ingenua), es decir, por fuerza bruta. En esta última, para una celda dada se procedería a verificar todas las demás y se decidiría si insertarla en el conjunto de celdas dentro de la distancia radial especificada.

En la búsqueda “naive”, se tendría un algoritmo cuyo costo computacional crecería de forma cuadrática en función de las celdas. En cambio, con una implementación BFS con una simple búsqueda focalizada, se reduce la cantidad de celdas a operar considerablemente y, por consiguiente, la cantidad de operaciones involucradas.

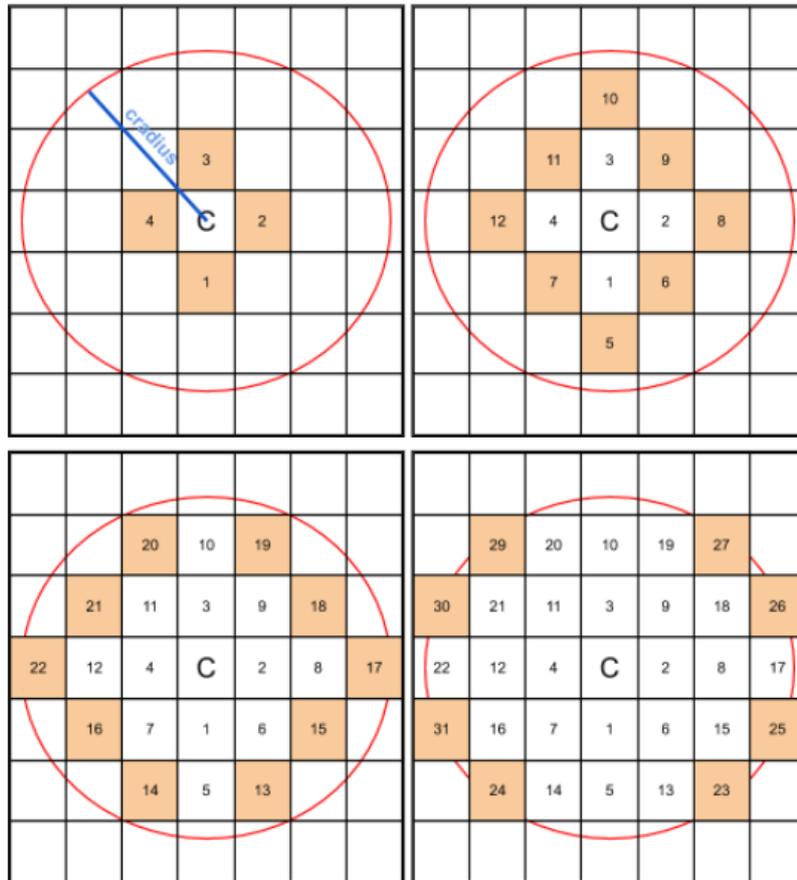


Fig. 7-5. Inundación para encontrar celdas vecinas a C. a) Primer nivel de BFS utilizando como base a las vecinas de C. b) Segundo nivel de BFS utilizando como base a las vecinas de 1,2,3,4. c) Tercer nivel de BFS utilizando como base a las vecinas de 5,6,7,8,9,10,11,12. d) Cuarto y último nivel reiterando el comportamiento anterior; luego no se pueden encontrar celdas vecinas que estén dentro del radio.

La estrategia de búsqueda implementada puede aplicarse una sola vez al iniciar la simulación. Debido a la reducción en la cantidad de celdas a operar y la cantidad de operaciones realizadas, es posible inicializar el modelo planteado, sobre una malla con grandes cantidades de celdas, en un tiempo insignificante.

6. Graduación de celdas - Expansión de celdas

El parámetro “relación de expansión de celdas”, descrito en la sección de parámetros de este capítulo, es un parámetro disponible dentro de la especificación de la utilidad *blockMesh* de OpenFOAM. Este fue usado, en conjunto con el precálculo de vecinos de celdas, para permitir alcanzar la convergencia de malla.

Como su nombre indica, el parámetro especifica la relación de expansión de celdas, es decir, establece el resultado de dividir el tamaño de la última celda a lo largo de un eje coordenado definido por el tamaño de la primera. El tamaño se mide a lo largo del mismo eje.

Para una malla de dominios bidimensionales, se pueden especificar dos valores, para los cuales su forma de cálculo se muestra en la **Fig. 7-6**.

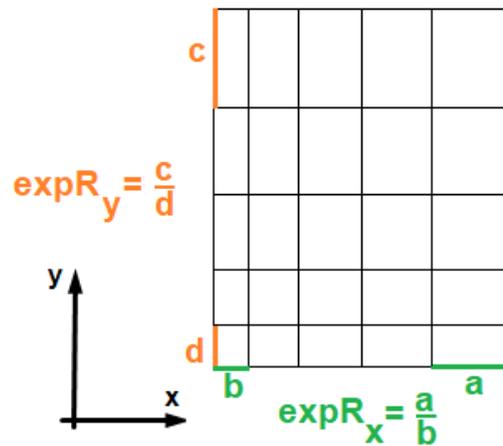
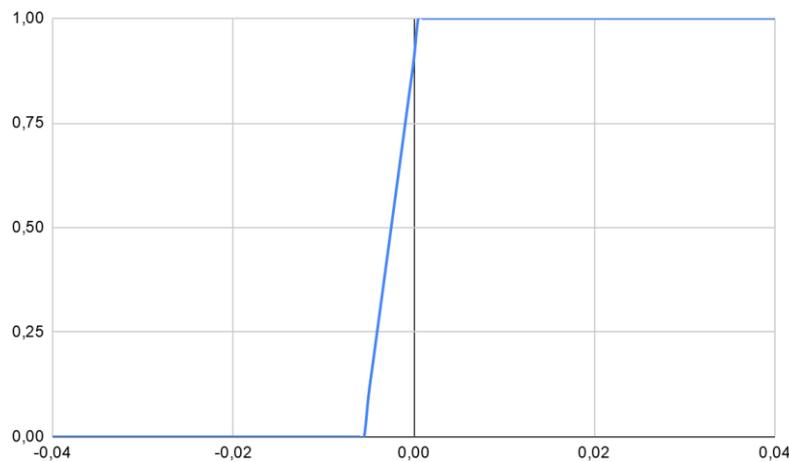


Fig. 7-6. Cálculo de las relaciones de expansión de celdas en dominios bidimensionales.

7. Alternativa a función escalón Heaviside

Como se mostró en las ecuaciones en derivadas parciales asociadas al modelo original (ecuaciones (2.6) y (2.7) del capítulo II), se utiliza como función de activación de la reacción $C \rightarrow D$ la función escalón unitario “ Θ ” (o función escalón de Heaviside). Durante la experimentación y análisis de los resultados de la simulación, se encontró que esta función presenta una variación de valores demasiado súbita (por su forma característica). Al ejecutarse dos casos de simulación casi idénticos, pero donde uno presentaba una leve variación en concentraciones de las especies, en el momento de la primera activación de la reacción (controlado por la función escalón), se generaban diferencias considerables en los resultados finales. Esto, comprometía la convergencia espacial, afectando sensiblemente la robustez y confiabilidad de la herramienta numérica.

Para amortiguar estas diferencias se formuló una función alternativa de tipo sigmoidea, en nuestro caso una lineal, definida por partes porque es capaz de cumplir el mismo propósito y además por la simplicidad para ser usada en los cálculos. La función se ilustra en la **Fig. 7-7**.



$$S(x) = \begin{cases} 0 & x < -0.005 \\ 160(x + 0.005) + 0.1 & -0.005 \leq x \leq 0 \\ 1 & x > 0 \end{cases}$$

Fig. 7-7. Función sigmoidea utilizada.

Realizando el reemplazo de la función escalón por esta función sigmoidea, y teniendo en cuenta el algoritmo para el cálculo de vecino y los chequeos de saturación a cierto radio, el modelo presentado en la sección 4 del capítulo II, se puede volver a escribir como se muestra a continuación:

$$\frac{\partial a}{\partial t} = D_a \nabla^2 a - kab \quad (6.1)$$

$$\frac{\partial b}{\partial t} = D_b \nabla^2 b - kab \quad (6.2)$$

$$\frac{\partial c}{\partial t} = D_c \nabla^2 c + kab - k_1 c \text{Sig}(c - c^*) - k_2 cd \quad (6.3)$$

$$\frac{\partial d}{\partial t} = k_1 c \text{Sig}(c - c^*) + k_2 cd \quad (6.4)$$

sujeto a:

$$\forall x_n \in \text{Celdas} \quad (6.5)$$

si $d(t, x_n) \geq \rho$ *entonces*

$$k_1(t, x_n) = k_2(t, x_n) = 0$$

$$\text{y } \forall x_i \in \text{Radio}(x_n, \text{Cradius}) \quad (6.6)$$

si $d(t, x_i) \geq \rho$ *entonces*

$$c^*(t, x_n) = 0$$

Donde x_n representa la n-ésima celda y t el tiempo. $k_1(t, x_n)$ representa el valor de k_1 en la celda x_n en el tiempo t ($k_2(t, x_n)$ y $c^*(t, x_n)$ representan los valores de k_2 y c^* respectivamente). $d(t, x_n)$ representa la concentración de la especie D en la celda x_n en el tiempo t y $\text{Radio}(x_n, \text{Cradius})$ indica las celdas en un radio Cradius desde la celda x_n .

Esta forma de expresar el modelo matemático es la que fue implementada en los algoritmos desarrollados. A continuación, se analizará el funcionamiento del algoritmo que controla el avance de tiempo de la simulación.

8. Algoritmo de actualización de paso de tiempo de la simulación

Las simulaciones de los patrones de Liesegang presentan situaciones críticas en las cuales es prioritario que el paso de tiempo sea lo suficientemente pequeño, para evitar errores numéricos que luego afecten el resultado. Un ejemplo de esto, se da frente a cambios rápidos en las concentraciones de las especies en momentos específicos, que pueden causar que no sean correctamente ponderados (por ejemplo, si ocurre un cambio de concentración en un tiempo no simulado entre dos tiempos que si son simulados). En cambio, en otras situaciones, los pasos de tiempo pueden ser mucho mayores sin introducir grandes errores en los resultados.

Si no se tuviese un algoritmo dinámico de ajuste de paso de tiempo, nos veríamos limitados a utilizar un paso de tiempo fijo. Este, al ser siempre igual, tendría que ser apto para todos los eventos en las simulaciones, y debería estar limitado superiormente por un máximo demasiado pequeño para que no afectase los resultados, alargando las simulaciones a tiempos indeseables o incluso difícilmente computables. Si se utilizaran valores algo mayores para el paso de tiempo, los resultados se verían modificados de forma impredecible (mostrando dependencia de los parámetros temporales y dando lugar a divergencias de tiempo).

Para obtener la convergencia temporal en las simulaciones (y a su vez la convergencia espacial⁹), hay que comprender la forma en la que se calculan y actualizan los pasos de tiempo. A continuación, se explican las cuestiones a tener en cuenta relacionadas a esto:

⁹ Como se mencionó antes, la convergencia de tiempo y la de malla están mutuamente relacionadas en este caso.

- Discontinuidades por condición de borde de la especie A: Es de extrema importancia que los primeros pasos de tiempo en la simulación sean suficientemente pequeños. Esto es así debido a la gran discontinuidad que existe entre las concentraciones del borde del dominio desde donde se introduce y difunde la especie A (esto se explica en más detalle en el capítulo VII) y la de las celdas siguientes, que tienen concentración nula. Esto, genera un gradiente de concentración muy alto que afecta principalmente a la ecuación (2.4).

Este gradiente de concentración alto de A, origina problemas debido al número de Fourier, que parte de la condición de estabilidad incondicional [45]. En términos sencillos, el número de Fourier refleja que proporción de una celda atravesará una especie en movimiento (debido al flujo difusivo), en un paso de tiempo. Si el número de Fourier es mayor a 1, las partículas de dicha especie se moverán a través de más de una celda en un paso de tiempo, y esto afecta negativamente la convergencia porque existe una pérdida de información sobre la simulación.

- Pasos de tiempo grandes generan errores numéricos: Los grandes pasos de tiempo generan errores numéricos debido a la naturaleza iterativa y aproximada de los solucionadores y métodos usados internamente para resolver ecuaciones en derivadas parciales.
- Grandes variaciones de la especie D (precipitado): En el modelo elegido, existe un evento catalizador que genera grandes cantidades de esta especie y es altamente discontinuo. Si no se simula con un paso de tiempo pequeño cuando ocurre, las bandas finales pueden aparecer en lugares erróneos. La naturaleza secuencial de la resolución del sistema de ecuaciones (ecuaciones (2.4) a (2.7)) genera restricciones adicionales sobre el paso de tiempo para garantizar la conservación de masa. Esto es causado porque los cambios de concentración son demasiado abruptos en un avance de tiempo, lo que no es deseable por lo exployado en los dos puntos anteriores.

Teniendo en cuenta los hechos presentados, se desarrolló un algoritmo de control/actualización de paso de tiempo. El diagrama de flujo correspondiente se puede observar en la **Fig. 7-9**.

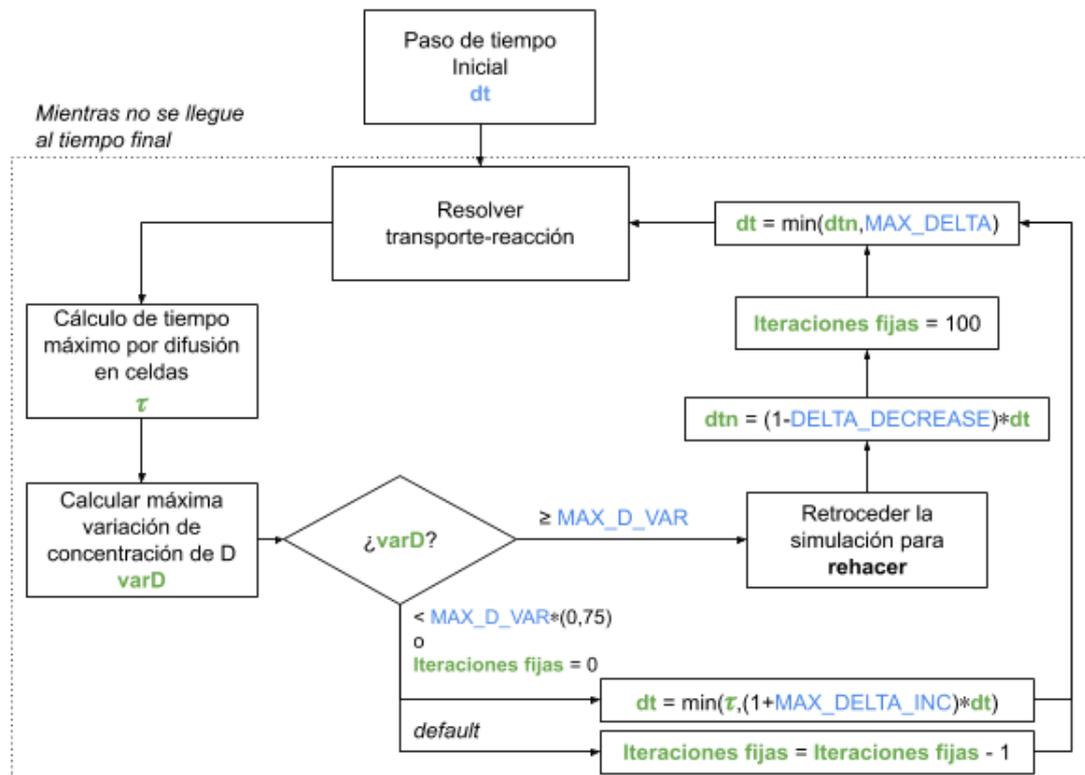


Fig. 7-9. Diagrama de flujo de la actualización del paso de tiempo. En celeste se ven las constantes cargadas desde archivos de configuración y en verde variables del algoritmo.

En la **Fig. 7-9** se pueden observar las múltiples variables utilizadas por el algoritmo, tanto internas (verde) como externas leídas del archivo de configuración *controlDict* (celeste). La función de cada una se describe a continuación:

- **Δt** : Esta variable corresponde al parámetro deltaT, explicado en la sección 2 de este capítulo. Es decir, el paso de tiempo inicial para la simulación.
- **τ (tau)**: Esta variable representa el tiempo máximo por difusión en celdas, y es el resultado de un cálculo que en sí depende de varios factores. De forma intuitiva, cuanto más rápido sea el flujo difusivo y más pequeñas sean las celdas sobre la cual este flujo existe, más bajo será el valor calculado (y viceversa).
- **varD** : Representa la máxima variación medida de la concentración de la especie D. Esta variación es la que existió entre un punto en el tiempo y el momento anterior (tiempo actual - último paso de tiempo). Se obtiene el máximo que hubo entre las variaciones de todas las celdas del dominio.
- **Iteraciones fijas**: Utilizada para evitar que el algoritmo rehaga pasos de tiempo continuamente. Esto sucedía si encontraba situaciones donde debía recalcular continuamente el paso de tiempo, porque cualquier intento de aumentarlo lo llevaba a sobrepasar el límite aceptado de varD al avanzar el tiempo de simulación. Esta lógica se explica en más detalle luego, con un ejemplo.
- **MAX_D_VAR**: Esta variable corresponde al parámetro de la sección 2, denominado *variationMax*. Esta es la máxima variación porcentual admitida para la concentración de D entre puntos en el tiempo. Este parámetro es uno de los que determinará la velocidad de avance de la simulación (modificando también la precisión de los resultados).
- **MAX_DELTA_INC**: Representa el porcentaje máximo de incremento de deltaT para el próximo paso de tiempo. El propósito de esta variable es el de mantener la simulación “controlada” para evitar que se utilicen grandes pasos de tiempo, aunque esté permitido por lo calculado con τ . Así, se evita un crecimiento rápido del paso de tiempo que reduciría la precisión en los resultados. Si bien su valor es modificable, las simulaciones realizadas y los resultados presentados en este informe fueron obtenidos con un valor de 0.2 (20%) que permite buenos resultados en tiempo de simulación y precisión de resultados. Esta variable puede ser modificada solamente con el propósito de agregar flexibilidad al algoritmo. En el archivo *controlDict* debe especificarse con el nombre “maxDeltaTIncrease”.
- **DELTA_DECREASE**: Esta variable representa el porcentaje en el que se reducirá el valor de deltaT cuando se determina necesario retroceder el tiempo y rehacer un paso de tiempo (esto es, cuando en el último avance de tiempo se superó la variación máxima admitida). Similarmente a la variable explicada antes, su valor es modificable sólo por cuestiones de flexibilidad del algoritmo. Los resultados de este informe fueron obtenidos con un valor también de 0.2 (20%) por ser conveniente en cuanto a tiempos de simulación y precisión de resultados. En el archivo *controlDict* debe especificarse con el nombre “deltaTDecrease”

Una vez desarrollado el algoritmo de tiempo, los hechos a tener en cuenta sobre los pasos de tiempo (y, por consiguiente, sobre el algoritmo) mencionados previamente, fueron tratados de la siguiente manera:

- **Discontinuidades por condición de borde de la especie A**: Se estableció un paso de tiempo inicial suficientemente bajo (por debajo del cual los resultados obtenidos son consistentes¹⁰) en los casos simulados, de forma que el algoritmo de tiempo pueda comenzar desde este valor base incrementándolo.

El incremento en el paso de tiempo se realiza usando el algoritmo presentado.

- **Pasos de tiempo grandes generan errores numéricos**: Con el algoritmo de tiempo presentado, el paso de tiempo usado en cada momento, está limitado por el valor máximo posible τ .

¹⁰ Antes de desarrollar el algoritmo, este valor bajo, a partir del cual los resultados no eran alterados, no existía.

- **Grandes variaciones de la especie D (precipitado):** El algoritmo presentado detecta dinámicamente cuando ocurren grandes variaciones de la especie D, y vuelve a resolver las ecuaciones involucradas en caso de que estas ocurran, con un paso de tiempo más pequeño.

Para ayudar al entendimiento del algoritmo de actualización de paso de tiempo, se elaboró un ejemplo gráfico para esclarecer su funcionamiento. En la **Fig. 7-10** se ilustra dicho ejemplo.

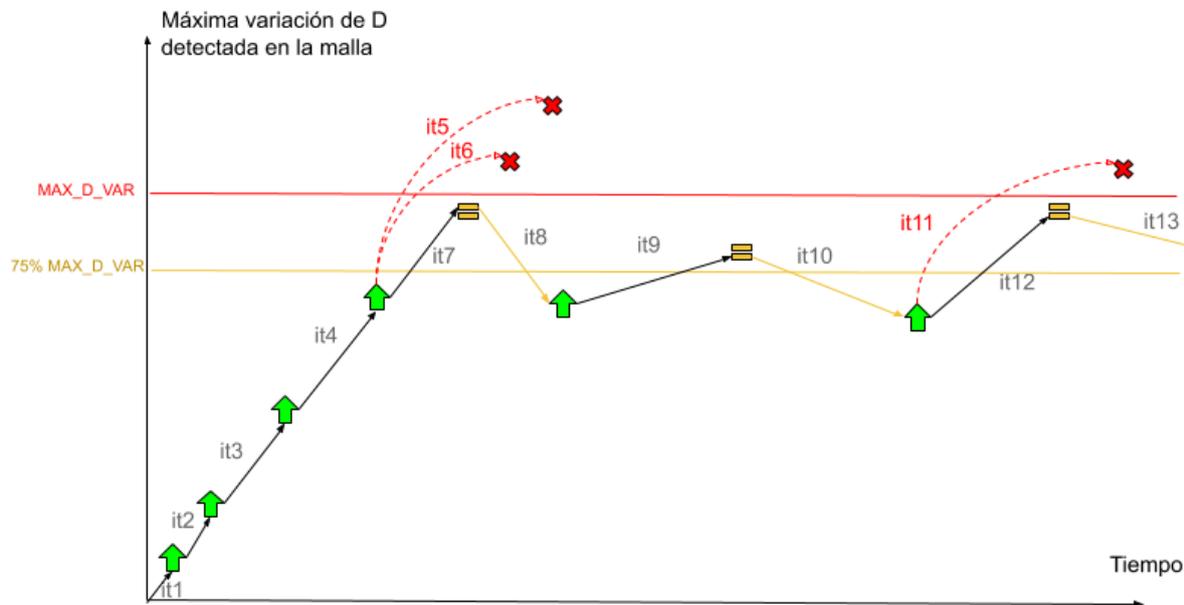


Fig. 7-10. Ejemplo de la ejecución del algoritmo de actualización de paso de tiempo.

A continuación, se explica paso a paso cómo sería la ejecución del algoritmo para el ejemplo de la figura:

- ❖ Entre las iteraciones 1 y 4 el paso de tiempo se incrementa en base a la constante τ o en un 20%, lo que sea menor.
- ❖ En la iteración 5, la máxima variación de D detectada en la malla, sobrepasa el límite de máxima variación de concentración de D admitida, por lo que la próxima iteración (6) se hace a partir desde el mismo tiempo, pero ahora con un paso de tiempo menor (-20%). Ocurre lo mismo con la iteración 6.
- ❖ Para la iteración 8, no se ha superado la variación máxima de D admitida, pero se encuentra dentro del 75% del máximo por lo que no se actualiza el paso de tiempo y se mantiene el paso de tiempo calculado en la iteración 7.
- ❖ En la iteración 9, la máxima variación de D en la malla cae por debajo del umbral del 75% (especificado en la **Fig. 7-9**), por lo que se actualiza el paso de tiempo con τ .

Como en la iteración 6 se superó la máxima variación admitida en concentración de D, eso implicaría esperar hasta la iteración 107 (por la cantidad asignada de iteraciones fijas) para calcular un nuevo paso de tiempo (según el diagrama de flujo de la **Fig. 7-9**). Sin embargo, el algoritmo también admite la posibilidad de que, antes, suceda el hecho de estar por debajo del umbral del 75%, luego de lo cual se puede continuar calculando nuevos pasos de tiempo.

- ❖ En la iteración 10, se vuelve a estar dentro del umbral por lo que se vuelve a usar el valor de paso de tiempo previo.
- ❖ En la iteración 11, la máxima variación de D en la malla se encontraba en la zona permisible para actualizar el paso de tiempo. A causa de esto, se actualiza y sobrepasa el límite, por lo que se hace la iteración 12 desde el mismo punto, pero reduciendo el paso de tiempo (-20%).

8.1. Cálculo de tiempo máximo τ por difusión en celdas

El cálculo del valor τ es posible solamente para las especies con difusividad no nula (A, B, C), debido a que su cálculo se basa en esta cantidad. Se realiza un cálculo de τ para cada celda del dominio, para cada una de las especies y, de todos estos valores calculados, se conserva el mínimo para definir una propuesta de nuevo paso de tiempo.

8.1.1. Derivación expresión 1D

Para obtener una primera expresión matemática para este valor, nos inspiramos en la definición del número adimensional de Fourier [46] y su utilidad. Este número, es muy utilizado para caracterizar la difusión de calor, pero puede también usarse con problemas análogos de difusión de materiales.

Basándonos en este número, se obtuvo la siguiente ecuación:

$$t_J = k \frac{|h|^2 J_{ref}}{D_J |\nabla J|}, \quad J \in \{A, B, C\} \quad (6.7)$$

En esta ecuación, D_J y J representan la difusividad y la concentración de una especie¹¹. J_{ref} representa la máxima concentración presente en la simulación de una especie (parametrizable por archivos de configuración como se explicó antes). h es la longitud máxima que puede recorrer el flujo (es decir el factor ∇J) en un paso de tiempo, que debe ser igual a la longitud sobre el eje X de una celda dada (**Fig. 7-11**), para evitar pérdida de información en la simulación. Finalmente, existe en la ecuación un parámetro configurable “ k ” con el cual se puede acelerar o desacelerar la simulación, ya que es solamente un valor de proporcionalidad.



Fig. 7-11. Longitud de interés de la celda en un flujo unidimensional.

Tomando el mínimo de este número entre todas las celdas y luego entre especies, nos quedan las siguientes dos ecuaciones:

$$\tau_J = \min \left\{ k \frac{|h_{celda}|^2 J_{ref}}{D_J |\nabla J|_{celda}} \right\}, \quad \forall celda \in MALLA, J \in \{A, B, C\} \quad (6.8)$$

$$\tau = \min\{\tau_J\} \quad (6.9)$$

De esta forma, se obtiene un valor para τ . Una propiedad de este valor es que, con una malla graduada (las celdas de inicio son más pequeñas que las del final de la malla) y con difusión constante como fue planteado en el proyecto, su valor tiende a incrementar con el tiempo. En 1D se puede razonar intuitivamente considerando que el flujo en las celdas iniciales cada vez será menor, incrementando τ para estas celdas. Además, estas celdas “abarcadas” por un flujo ínfimo cada vez serán más grandes en tamaño, reforzando el incremento.

¹¹ Esta especie puede ser solo A, B o C, ya que D posee difusividad nula en el fenómeno estudiado.

8.1.2. Generalización de la expresión a 2D

Teniendo una situación en dos dimensiones, el flujo de la especie (∇J) no necesariamente está alineado con las caras de la celda examinada, por lo que hay que hacer algunas generalizaciones. En la **Fig. 7-12** se muestra esta situación, para un dominio formado por una malla cuadriculada.

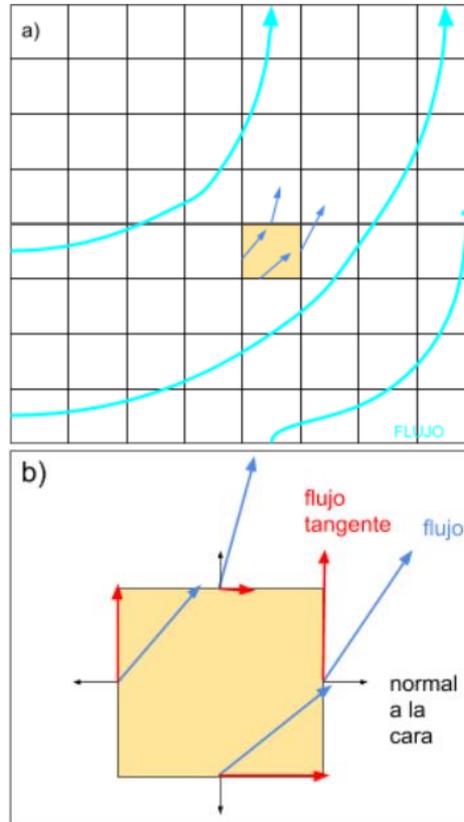


Fig. 7-12. a) Flujo de ejemplo en una malla bidimensional cuadriculada. b) Misma celda coloreada de a) ampliada.

Si bien el dominio mostrado en la **Fig. 7-12** está constituido por una malla cuadriculada, esto es una suposición para facilitar la obtención de expresiones generalizadas.

En este caso bidimensional, el cálculo es similar a lo visto previamente, pero ahora sobre cada cara de la celda. Con la idea de obtener una expresión análoga, y considerando el hecho de que en el caso 1D el flujo y la cara son colineales, es necesario calcular el flujo tangencial a la cara (indicado en rojo en **Fig. 7-12**). Este se puede definir de la siguiente forma:

$$|\nabla J| |\sin \theta| \quad (6.10)$$

Donde θ es el ángulo entre el flujo y la normal de la cara.

Multiplicando por el módulo de la normal a una cara de la celda (igual a 1, por lo que la expresión es la misma), resulta:

$$|\nabla J| |n_{cara}| |\sin \theta| \quad (6.11)$$

Retomando algunos conceptos de álgebra lineal, la expresión puede reescribirse como el módulo del producto cruz entre vectores ∇J y n_{cara} :

$$|\nabla J \times n_{cara}| \quad (6.12)$$

Si en la ecuación (6.7), a partir de la cual se derivó una expresión unidimensional, consideramos a h como la longitud de la cara de una celda y reemplazamos ∇J por la ecuación (6.12) (por considerarse ahora el flujo tangencial a una cara), resulta la siguiente expresión:

$$t_J = k \frac{|h_{cara}|^2}{D_J} \frac{J_{ref}}{|\nabla J \times n_{cara}|}, \quad J \in \{A, B, C\} \quad (6.13)$$

Finalmente, de forma similar a las expresiones 1D, tomamos el mínimo entre todas las caras y luego entre las especies, obteniendo:

$$\tau_J = \min \left\{ k \frac{|h_{cara}|^2}{D_J} \frac{J_{ref}}{|\nabla J \times n_{cara}|} \right\}, \quad \forall cara \in MALLA, J \in \{A, B, C\} \quad (6.14)$$

$$\tau = \min\{\tau_J\} \quad (6.15)$$

De esta manera se calcula el valor de τ en el algoritmo de actualización de paso de tiempo desarrollado. Estas expresiones se encuentran generalizadas para dominios bidimensionales, y son capaces de funcionar sobre una amplia variedad de mallas y celdas, como se mostrará en el siguiente capítulo.

VII. Funcionamiento de la aplicación

1. Dominios Unidimensionales - 1D

1.1. Configuración de los casos de simulación

Como se mencionó en el capítulo VI, para poder ejecutar la simulación con OpenFOAM, primero hay que configurar el caso a ser ejecutado, estableciendo valores en determinados archivos de parámetros.

A partir del artículo de Lagzi, se pudo extraer algunos de los valores iniciales para configurar la simulación; los demás se establecieron con la idea de lograr un balance inicial entre costo computacional y precisión. Los valores utilizados se muestran en la **Tabla 8-1**.

Parámetro/Variable	Valor	Observaciones
Concentración especie A	0.5 [molar]	Valor aplicado como condición de borde izquierdo del dominio. Esto es denominado una condición de borde de Dirichlet [47]. En el resto del dominio es 0 molar.
Concentración especie B	0.01 [molar]	Mismo valor en todo el dominio.
Concentración especie C	0 [molar]	
Concentración especie D	0 [molar]	
Concentración máx. especie D en bandas (ρ)	0.03 [molar]	Valor elegido entre los usados por Lagzi.
Difusividad especie A	9×10^{-10} [m^2/s]	
Difusividad especie B	9×10^{-10} [m^2/s]	
Difusividad especie C	9×10^{-10} [m^2/s]	
Difusividad especie D	0 [m^2/s]	
Valor de punto de floculación c^* (cs)	0.006 [molar]	
Constante de velocidad química para la reacción $A+B \rightarrow C$ (kf)	1 [molar/seg]	
Constantes de velocidad para la coagulación (k1)	1 [1/seg]	
Constantes de velocidad para la formación de precipitado autocatalítico (k2)	1 [1/seg]	
Extensión del dominio	0.08 [m]	
Cantidad de celdas	800	Inicialmente se estableció una malla de 800 celdas horizontales a partir de lo especificado en el artículo.
Tiempo de inicio de simulación (startTime)	0 [seg]	
Tiempo de fin de simulación (endTime)	700000 [seg] (~ 8 días)	Se eligió este valor, un poco mayor a lo que duró el experimento químico mostrado en el artículo, que se extendió durante 7 días (604800 segundos), para dar tiempo a consumir la especie B y por practicidad de manejo.

Paso de tiempo inicial para la simulación (deltaT)	1×10^{-10} [seg]	Este valor se determinó como el máximo posible a partir del cual menores valores no cambian el resultado (lo que demuestra la convergencia temporal). Los valores mayores mostraron divergencias en los resultados.
Máximo DeltaT (maxDeltaT)	20 [seg]	Valor inicialmente seleccionado de forma arbitraria. Al avanzar con el proyecto se descubrió que influye en el resultado si el algoritmo de control de paso de tiempo no es ideal. Con el nuevo algoritmo para convergencia de tiempo, este ya no era el caso. La limitación del paso se hace teniendo en cuenta la reacción.
Concentración de referencia de la especie A (refValA)	0.5 [molar]	
Concentración de referencia de la especie B (refValB)	0.01 [molar]	
Concentración de referencia de la especie C (refValC)	0.01 [molar]	Este valor se estableció debido al punto de floculación.

Tabla 8-1. Configuración inicial casos 1D.

Los parámetros indicados en la **Tabla 8-1**, son cargados dentro de los distintos archivos que conforman un “caso” de OpenFOAM, explicados en el capítulo VII. Los parámetros se deben especificar en los archivos correspondientes y en la forma estandarizada por OpenFOAM, o la forma esperada por la implementación realizada en este proyecto. En el anexo E se ejemplifica el contenido de los distintos archivos de los casos, para un caso 2D simple.

1.2. Estructura del dominio

Durante este proyecto, en primer lugar, se realizaron simulaciones del fenómeno en dominios unidimensionales. Las mismas, se hicieron tomando como referencia el artículo científico de Lagzi, usando la implementación realizada del modelo presentado allí. El objetivo era ajustar el modelo en lo que fuese necesario para obtener uno que contara con ciertas características deseables de convergencia antes de poder proceder a dominios de mayores dimensiones.

Los cálculos de las simulaciones eran realizados usando OpenFOAM y los resultados luego podían ser visualizados usando la herramienta ParaView cargando los datos generados por OpenFOAM¹². El dominio unidimensional, donde se simula el fenómeno, representa una probeta que contiene la especie B y, desde un extremo, se introduce la especie A, que reacciona produciendo la especie C. Luego, esta especie se precipitará como la especie D. En la **Fig. 8-1** se ilustra el dominio utilizado con una pequeña cantidad de celdas para simplificar la representación. En un esfuerzo por facilitar la inclusión de imágenes del dominio en este informe, se prefirió orientar el dominio unidimensional de forma horizontal (en lugar de vertical, como sería en un experimento de laboratorio no simulado).

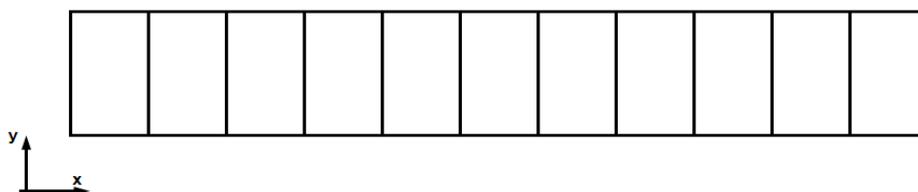


Fig. 8-1. Ejemplo de dominio 1D con pocas celdas.

El dominio, como se mencionó en la sección anterior, contaba con una longitud total de 0.08 metros.

¹² Los pasos para cargar los resultados de las simulaciones en ParaView están descritos en el Anexo F - “Instrucciones de visualización de casos en ParaView”.

1.3. Funcionamiento de la implementación del modelo inicial

En las **Fig. 8-2 – 8-5**, se muestran visualizaciones usando ParaView de los resultados de las simulaciones¹³. En dichas figuras, se representan las concentraciones de las distintas especies expresadas en concentraciones molares/molaridad (mM) al finalizar la simulación (a los 700000 segundos), en el dominio unidimensional y utilizando la renderización de ParaView. Se incluye una referencia del valor de concentraciones de acuerdo al color a la derecha de las figuras.



Fig. 8-2. Concentración de la especie A en el dominio al finalizar la simulación.



Fig. 8-3. Concentración de la especie B en el dominio al finalizar la simulación.

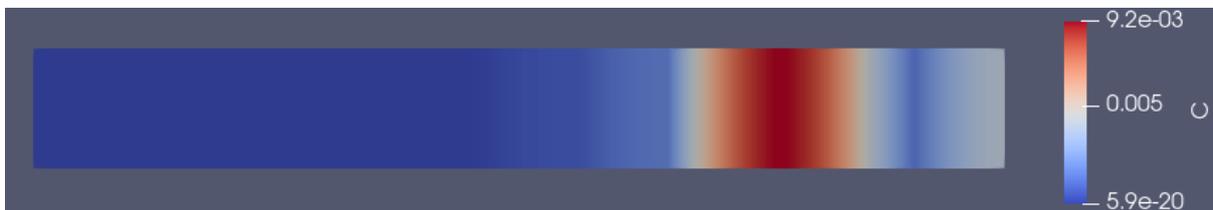


Fig. 8-4. Concentración de la especie C en el dominio al finalizar la simulación.

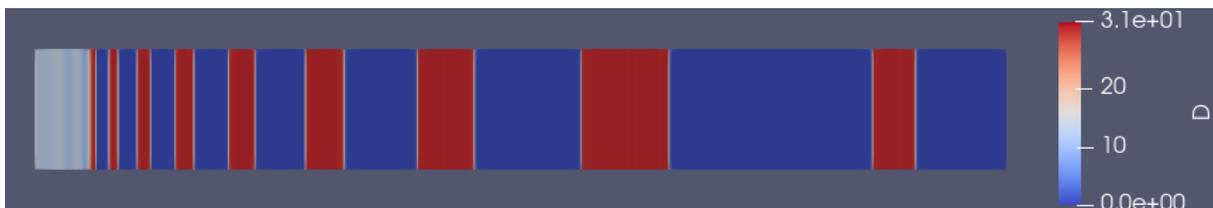


Fig. 8-5. Concentración de la especie D en el dominio al finalizar la simulación. Se observan las bandas características a los patrones de Liesegang formadas (a excepción de la última que no alcanzó a ensancharse completamente).

Como se observa en la **Fig. 8-2**, al finalizar la simulación, la especie A ha difundido desde el punto de ingreso (extremo izquierdo del dominio) reaccionando con la especie B en el proceso (**Fig. 8-3**).

La especie B ha sido consumida casi en su totalidad durante la reacción con A y ha dado lugar a la formación de la especie C en distintos puntos (determinados por el modelo matemático usado), generando saturaciones de ella en esos lugares (**Fig. 8-4**). Luego se precipita como especie D reduciéndose la concentración de C allí. Es por esto que en la **Fig. 8-4** solo se observa concentración en la ubicación aproximada de la última banda, que es la que todavía está en etapa de ensanchamiento.

Finalmente, como se muestra en la **Fig. 8-5**, la precipitación de la especie C da lugar a la formación de la especie D formando bandas siguiendo los patrones de Liesegang conocidos.

¹³ Independientemente del modelo y algoritmos usados para la simulación (sea la implementación realizada del modelo de Lagzi o el modelo desarrollado e implementado en este proyecto), el resultado poseerá la misma forma general (variando posiblemente la cantidad de bandas, anchos y sus posiciones).

Como se puede notar en la última figura, la cantidad de B y la cantidad de segundos simulados hicieron imposible la generación de más especie C y, en consecuencia, D. Por este motivo, la última banda no pudo formarse completamente y no terminó de “ensancharse”.

1.4. Convergencia temporal

Para comprobar la convergencia temporal (definida en la sección 6, capítulo III) en el modelo original implementado, obtenido del artículo de Lagzi, y con el algoritmo de control de tiempo tomado de porousScalarTransportFoam, se procedió a ejecutar la simulación con los parámetros mostrados en la **Tabla 8-1**, pero se probaron distintos valores de deltaT inicial (definido en el capítulo VI, determina el paso de tiempo inicial de la simulación).

El objetivo de variar el parámetro deltaT, era corroborar si existía un valor a partir del cual valores más pequeños no modificaban el resultado considerablemente o si seguían produciéndose cambios importantes en los resultados (mostrando la dependencia de los parámetros en la posición y ancho de las bandas).

Al realizarse las pruebas iniciales, se obtuvieron los resultados que se muestran en las **Fig. 8-6 y 8-7**. En las figuras, puede observarse gráficas que indican la concentración de la especie D en el dominio para distintos valores del parámetro deltaT.

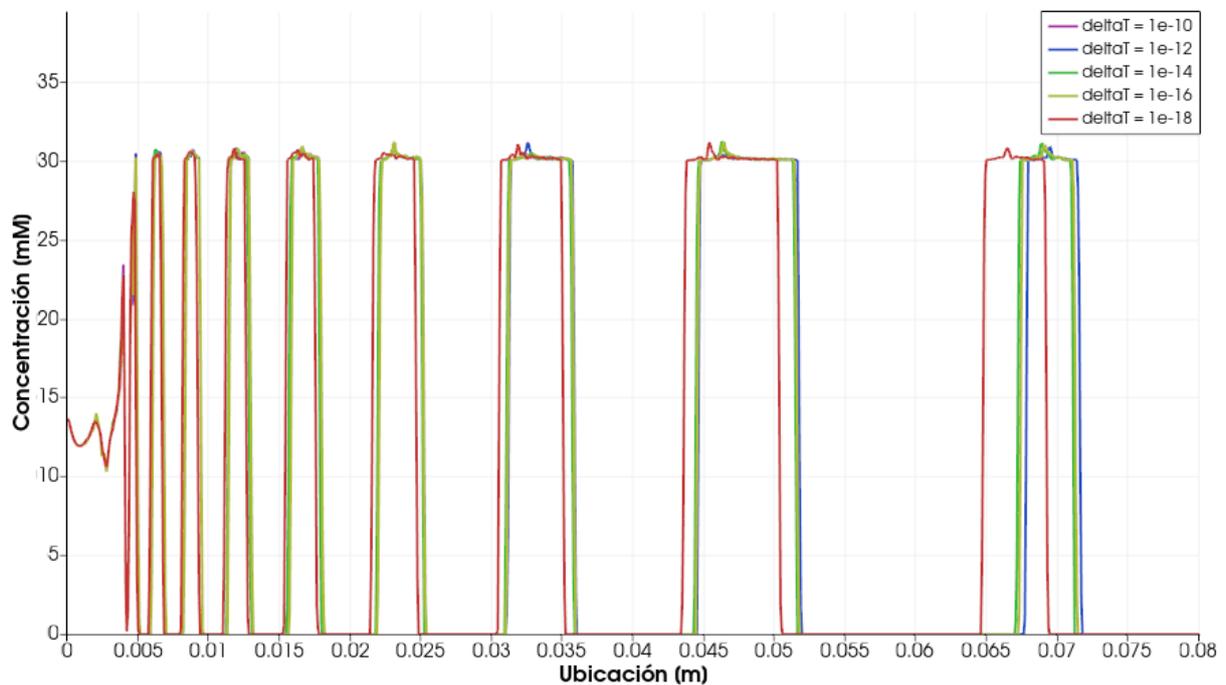


Fig. 8-6. Concentración de la especie D a lo largo del dominio para distintos valores del parámetro deltaT.

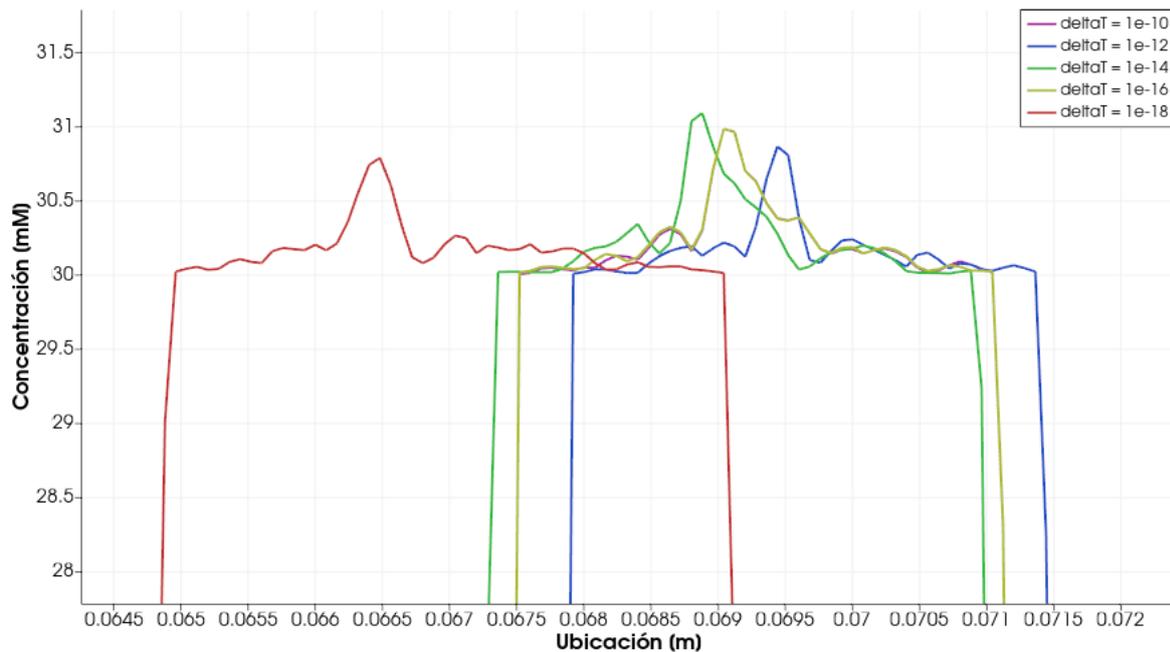


Fig. 8-7. Acercamiento sobre la última banda mostrada en la **Fig. 8-6**. Las bandas correspondientes a $\text{deltaT} = 1 \times 10^{-10}$ y $\text{deltaT} = 1 \times 10^{-16}$ están superpuestas.

Como se puede apreciar en dichas figuras, los resultados mostraron una gran variación al cambiar deltaT . La posición de las bandas era afectada por el valor del parámetro deltaT , mostrando una divergencia temporal. La convergencia no se mostraba tan lejana en los primeros casos (valores más grandes de deltaT), pero es en el último caso donde se hace evidente que no existe convergencia temporal.

Otro efecto indeseado, presente en esos resultados, son los sobrepicos en las bandas, que indican una acumulación mayor de la especie D de lo que estaba admitido en las bandas.

Mediante el cuidadoso estudio del problema y análisis de los resultados, se detectó que la falta de convergencia temporal y los demás efectos eran una consecuencia directa de un inadecuado manejo del avance temporal de la simulación. Esto estaba originado en el algoritmo “heredado” del solucionador *porousScalarTransportFoam* usado como base, que no era el más conveniente para el fenómeno estudiado.

En el momento inicial de la simulación, una gran cantidad de la especie A ingresa desde el extremo izquierdo del dominio. Esta situación da lugar a que el gradiente de concentración de la especie A sea muy grande en el en ese punto. Debido a esto, la velocidad de simulación allí (o, en otras palabras, el valor del paso de tiempo) es realmente importante, porque determina fuertemente los resultados de resolver las ecuaciones diferenciales. Resultados erróneos en los primeros momentos de la simulación (o en las primeras celdas del dominio) afectan los resultados en lo posterior (en tiempo o espacio), como se mencionó en la sección 8 del capítulo VI.

Como consecuencia de lo explicado anteriormente, se observó que los resultados de la simulación eran muy sensibles a los valores de los parámetros temporales, principalmente el valor de deltaT , a partir del cual los próximos pasos de tiempo son calculados. Un modelo y algoritmo de control de avance de tiempo que presente convergencia temporal (definida en el capítulo II, sección 6), debería ser capaz de atenuar la dependencia en los resultados de los parámetros temporales. Debería existir suficiente independencia como para que los resultados varíen despreciablemente ante cambios de dichos parámetros (a partir de cierto punto).

Una vez desarrollados los nuevos algoritmos para el control de avance de tiempo de la simulación, correspondientes al sprint 3 y descritos en el capítulo VI, las pruebas fueron repetidas con el mismo conjunto de parámetros que antes. Sin embargo, con los nuevos algoritmos, fueron incluidos nuevos parámetros al modelo, para que los algoritmos pudiesen efectuar las correcciones en tiempo de simulación necesarias. Los parámetros agregados fueron los mostrados en la **Tabla 8-2**.

Parámetro/Variable	Valor	Observaciones
Variación máxima admitida de la especie D entre pasos de tiempo (variationMax)	5 %	Este valor mostró ser un punto medio que permite alcanzar resultados suficientemente precisos y en tiempos aceptables.
Valor de constante k para fórmula de TauD (kValueTauD)	50000	Con este valor se pudo obtener resultados suficientemente precisos y en tiempos aceptables.

Tabla 8-2. Parámetros agregados para los nuevos algoritmos desarrollados para lograr convergencia temporal.

Los nuevos resultados obtenidos frente a variaciones de deltaT se pueden observar en las **Fig. 8-8 y 8-9**.

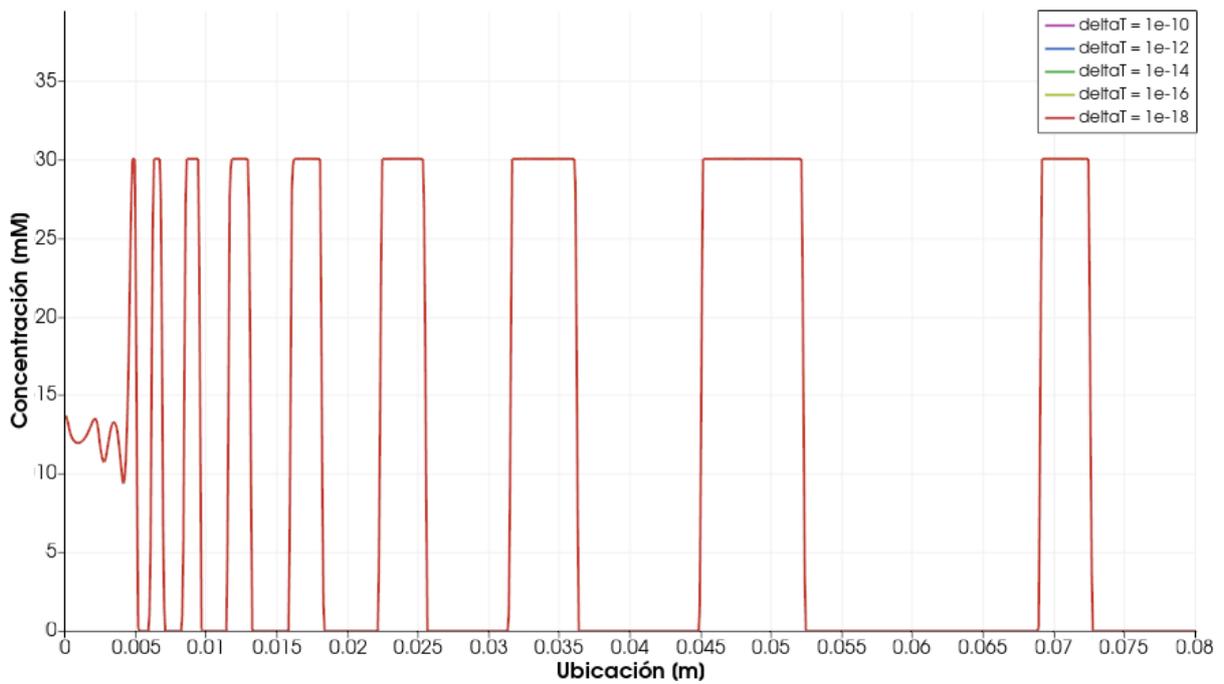


Fig. 8-8. Concentración de la especie D a lo largo del dominio para distintos valores del parámetro deltaT, usando nuevos algoritmos.

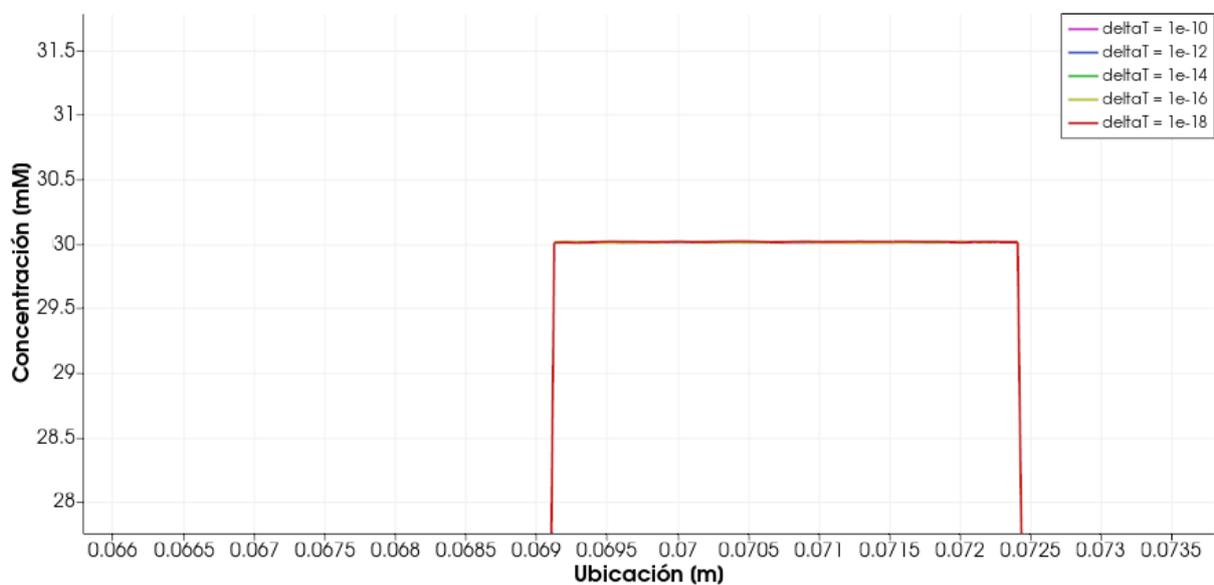


Fig. 8-9. Acercamiento sobre la última banda mostrada en la **Fig. 8-8**. Se utilizó el mismo nivel de acercamiento que en las pruebas anteriores.

Como se puede observar, los resultados mostraron un alto nivel de convergencia temporal. Para todos los casos ejecutados, y como se muestra en la **Fig. 8-8**, las bandas están superpuestas. Si se realizaba un acercamiento (en la misma magnitud que en la **Fig. 8-7**), como se observa en la **Fig. 8-9**, la diferencia entre las bandas es imperceptible. Se puede decir que, en sentidos prácticos, las bandas están en el mismo lugar.

Los resultados mostrados fueron alcanzados sin comprometer demasiado en términos de costo computacional, dado que la velocidad de la simulación es ajustada en tiempo de ejecución, teniendo en cuenta la velocidad de la reacción química que se da en el dominio. Sin embargo, si fuese necesario, se podría configurar para que los resultados tuviesen una precisión un poco menor, pero las simulaciones fuesen menos costosas computacionalmente (lo que se traduce a menor tiempo de ejecución). Para esto, solo habría que admitir una velocidad máxima de reacción mayor, modificando `variationMax` (que es el parámetro que controla la velocidad a la que avanza la simulación, aumentando o reduciendo la precisión de los resultados de resolver las ecuaciones).

Sin duda, la convergencia temporal se alcanzó con los algoritmos desarrollados, obteniendo una estabilidad en los resultados que previamente no existía.

1.5. Convergencia de malla

Una vez alcanzada la convergencia temporal, el siguiente paso fue probar la convergencia de malla en el modelo inicial, que ahora ya fue modificado.

Para realizar las pruebas de convergencia de malla, se utilizó la mayoría de los valores de los parámetros de la **Tabla 8-1**, y se mantuvieron los cambios/agregados especificados en la **Tabla 8-2**. Los cambios realizados sobre los valores previos de los parámetros son mostrados en la **Tabla 8-3**.

Parámetro/Variable	Valor	Observaciones
Cantidad de celdas	≥ 5000	A medida que se avanzó sobre las pruebas, se mostró necesario aumentar la cantidad de celdas para poder hacer un mejor análisis de convergencia y mejorar la precisión de los resultados. Menor cantidad de celdas no permitía obtener conclusiones respecto a la convergencia.
Tiempo de fin de simulación (endTime)	450000 [seg] (~ 5 días)	Se redujo el valor anterior como estrategia para reducir el costo computacional total de ejecutar las simulaciones, que se incrementó enormemente al agregar celdas al dominio simulado (y se configuró para que estas se expandan a lo largo del dominio). La convergencia de malla podía ser estudiada de igual manera con este valor.
Paso de tiempo inicial para la simulación (deltaT)	1×10^{-10} [seg]	Se tomó este valor como el mínimo necesario para contar con convergencia de tiempo.

Tabla 8-3. Cambios realizados sobre parámetros para probar convergencia de malla.

Una vez establecidos los parámetros indicados, y para comprobar la convergencia de malla en el modelo original propuesto por Lagzi, se ejecutó la simulación configurando distintas cantidades de celdas en el dominio.

La variación de la cantidad de celdas pretendía encontrar un valor a partir del cual una mayor cantidad de celdas no modificarían el resultado o lo harían despreciablemente. Si esto no sucedía, existía divergencia en los resultados y en el modelo usado, por lo que debieron implementarse modificaciones al modelo existente en forma de nuevos algoritmos que permitieran alcanzar la convergencia.

Al realizarse las pruebas mencionadas, con el modelo original, se obtuvieron los resultados que se muestran en las **Fig. 8-10, 8-11, 8-12 y 8-13**. A fin de brindar mayor claridad, además del gráfico con las concentraciones de las especies para los distintos casos ejecutados, se agregan los gráficos correspondientes a cada caso por separado para que se pueda identificar fácilmente la posición de las bandas en cada uno. Por otro lado, se representan las bandas con gráficas de la concentración de la especie D a lo largo del dominio, para distintas cantidades de celdas.

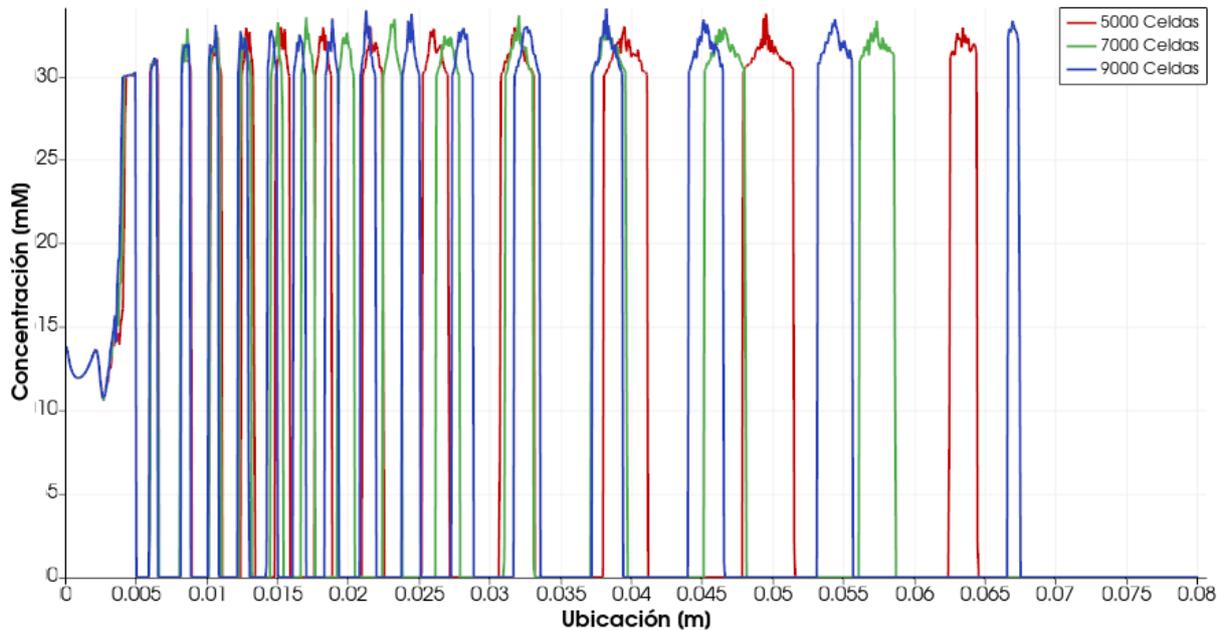


Fig. 8-10. Concentración de la especie D a lo largo del dominio para distinta cantidad de celdas en el dominio simulado.

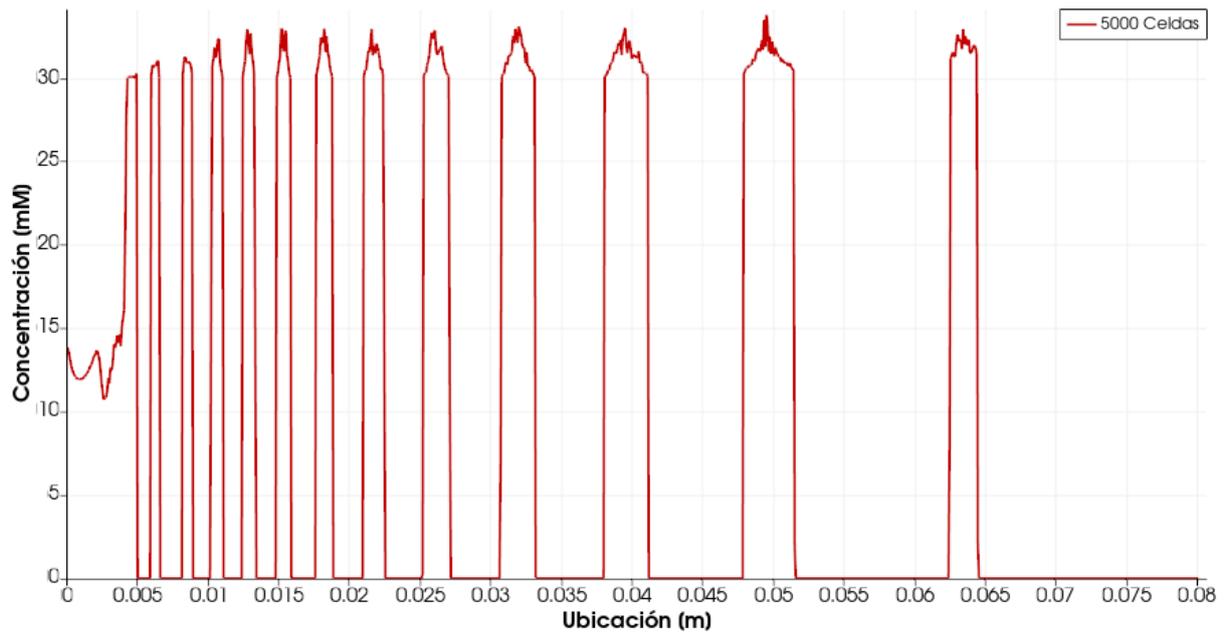


Fig. 8-11. Concentración de la especie D a lo largo del dominio para un dominio con 5000 celdas.

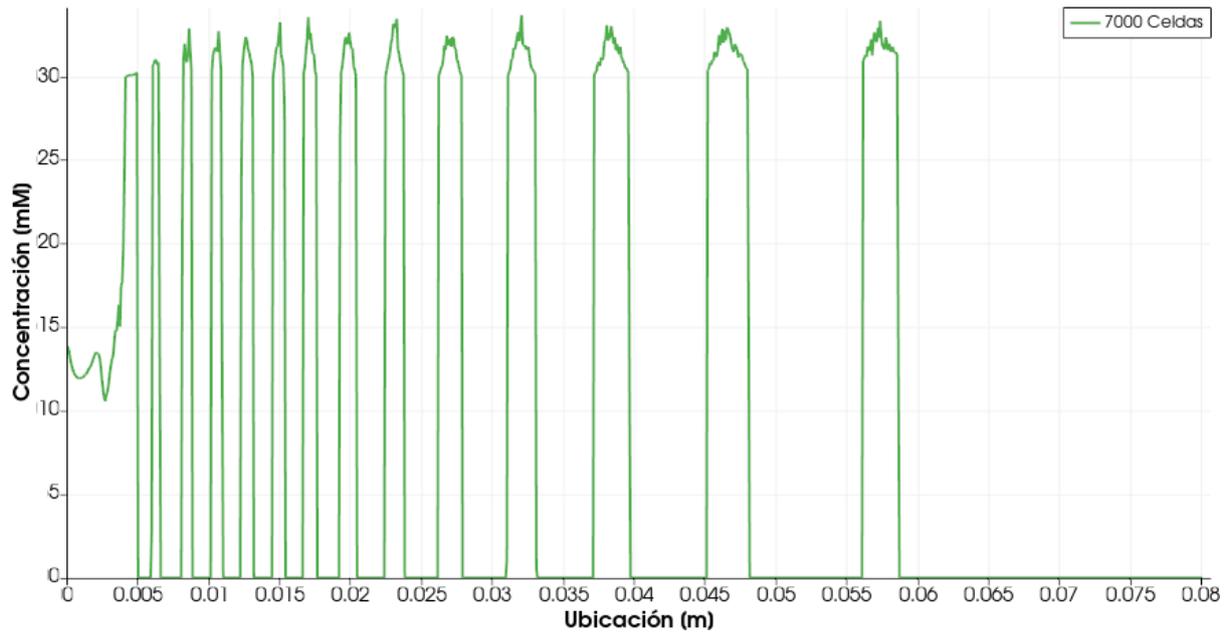


Fig. 8-12. Concentración de la especie D a lo largo del dominio para un dominio con 7000 celdas.

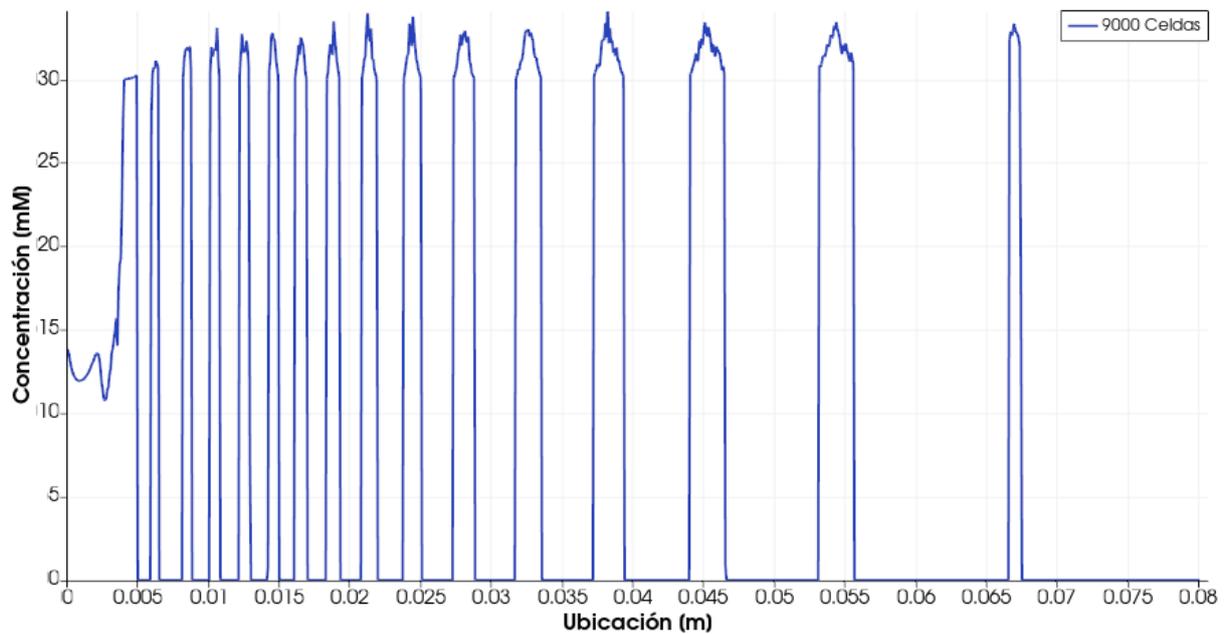


Fig. 8-13. Concentración de la especie D a lo largo del dominio para un dominio con 9000 celdas.

Como se puede observar en los resultados mostrados, no había indicios de convergencia a un único resultado a medida que variaba el tamaño de las celdas. Como sucedió con la convergencia de tiempo, el modelo original no presentaba convergencia de malla. El cambio de densidad de celdas en el dominio simulado, afectaba directamente la posición y ancho de las bandas haciendo que los resultados finales variarían ampliamente entre sí. Adicionalmente, se puede ver que la forma también está influenciada por el aumento de celdas en el dominio.

Al igual que sucedía con la convergencia de tiempo, el gradiente tan pronunciado de la especie A en el extremo izquierdo del dominio, genera en el resultado final una gran dependencia de lo que se simula en las primeras celdas (y la precisión con la cual se resuelven las ecuaciones involucradas en ese punto). Es por esto, que el parámetro más crítico en cuanto a obtención de resultados consistentes, termina siendo el tamaño (y la distribución de dichos tamaños) de celdas porque determina cuántos valores existen para calcular las concentraciones de las especies en una porción de espacio determinado.

Como consecuencia de lo observado en los resultados, se desarrollaron los algoritmos para lograr convergencia de malla (durante el *sprint* 4) descritos en el capítulo VI. Los algoritmos se orientaron a la detección de saturación de la especie C (punto en el cual se forma D) en base a distancias geométricas en lugar de celdas. Además, se implementó la graduación de celdas en el dominio (expansión de celdas, explicado también en el capítulo VI) para contar con celdas más pequeñas y, por consiguiente, mayor precisión de resultados en la región donde se producía la entrada de la especie A (extremo izquierdo del dominio).

Con los agregados al modelo y algoritmos mencionados, se logró alcanzar la convergencia de malla cuando la cantidad de celdas es suficientemente grande. A medida que se aumenta el número de celdas, los resultados obtenidos son cada vez más similares entre sí, hasta llegar al punto donde la diferencia es casi insignificante. Ahora, si la cantidad de celdas siguiese aumentando, se alcanzaría un punto donde los resultados no cambiarían más (pero en el proceso, las simulaciones demorarían cada vez más en ser terminadas).

Para realizar las nuevas pruebas de convergencia de malla, se conservaron los parámetros y modificaciones usadas para las pruebas previas de convergencia de malla. Se agregaron parámetros adicionales, propios del nuevo modelo y algoritmos desarrollados. Estos se pueden observar en la **Tabla 8-4**.

Parámetro/Variable	Valor	Observaciones
Radio de cristalización (r)	0.00015 [m]	
Relación de expansión de celdas (expR)	20	Se determinó que el valor de 20 proporcionaba la precisión adecuada en las celdas al principio del dominio.

Tabla 8-4. Parámetros agregados para los nuevos algoritmos desarrollados para lograr convergencia espacial o de malla.

El valor al que se estableció el radio de cristalización, fue obtenido a partir del hecho de que el dominio usado por Lagzi medía 0,08 metros de largo y las celdas utilizadas median 1×10^{-4} metros de largo. Para que el modelo original fuese compatible en la detección de saturación de vecinos, debía cubrirse al menos una celda hacia los lados a partir del centro de una celda dada.

Los resultados obtenidos para distinta cantidad de celdas en la malla y con el nuevo modelo y algoritmos se pueden observar comparados en la **Fig. 8-14**. Por otro lado, en la **Fig. 8-15** se ilustra un acercamiento sobre la ubicación de la última banda para observar la tendencia a convergencia.

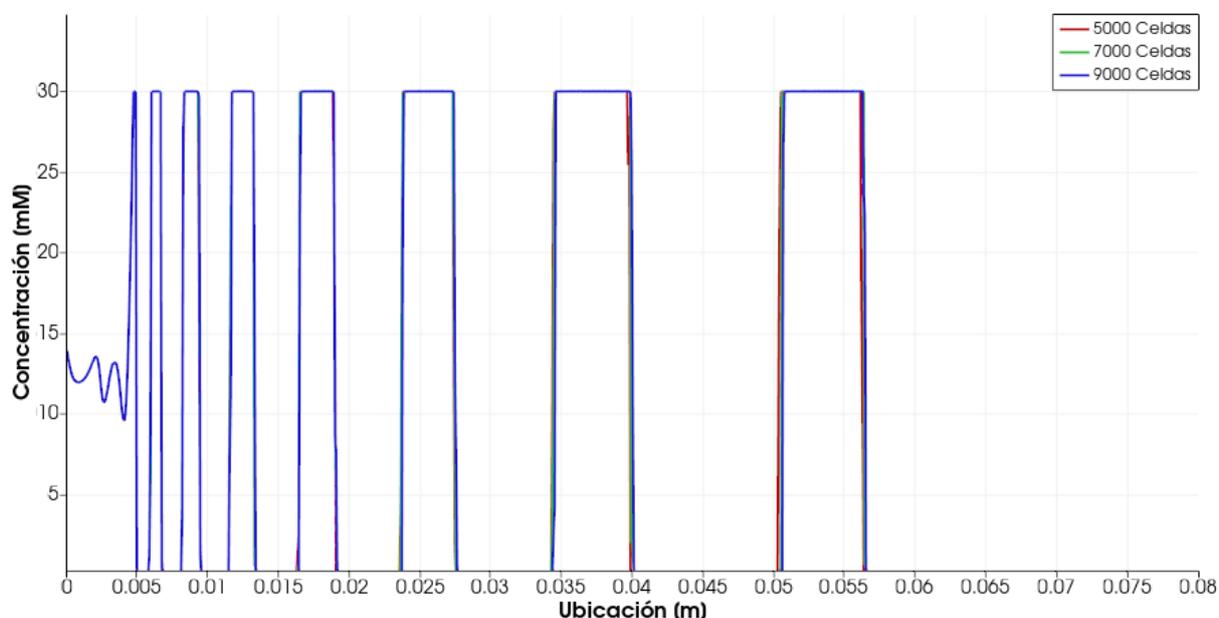


Fig. 8-14. Concentración de la especie D a lo largo del dominio para distinta cantidad de celdas en el dominio simulado, usando nuevos algoritmos.

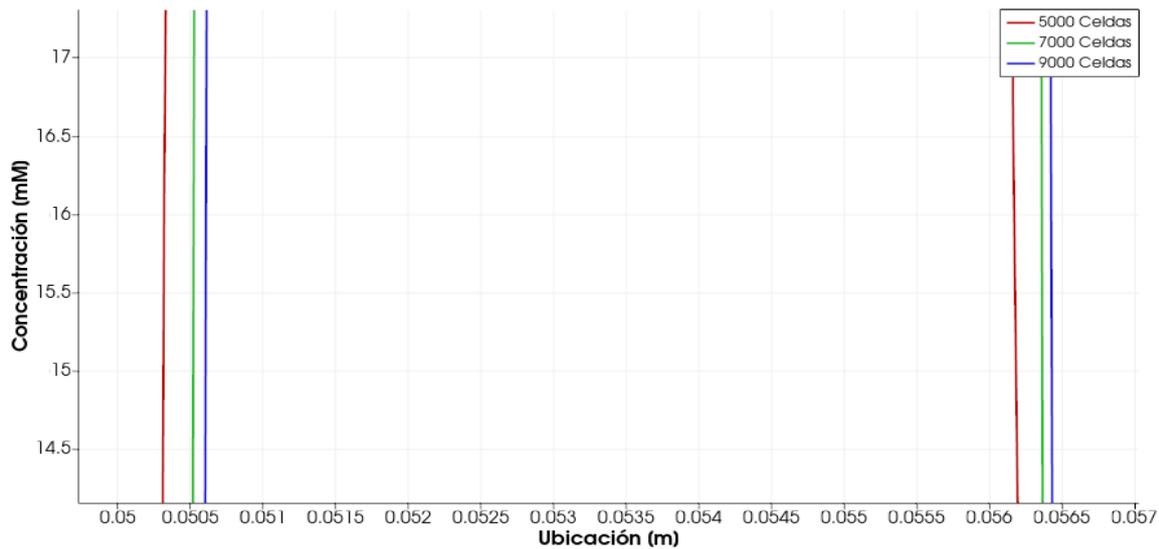


Fig. 8-15. Acercamiento sobre la última banda mostrada en la Fig. 8-14. Se puede apreciar la tendencia a converger a medida que se aumenta la cantidad de celdas.

Como se puede observar en la Fig. 8-14, los nuevos resultados evidencian claramente un alto grado de similitud en los resultados. Para distintos casos de ejecución, con distintas cantidades de celdas, se obtienen resultados prácticamente similares. Más aún, en la Fig. 8-15 se puede observar que a medida que la cantidad de celdas es incrementada en el dominio, si bien la ubicación de las bandas varía, lo hace en distancias cada vez menores. Esto, muestra que llegará un punto a partir del cual los resultados no cambiarán más ante cambios en la cantidad de celdas, lo que, en otras palabras, indica convergencia de malla.

2. Dominios Bidimensionales - 2D

Una vez logrados los objetivos planteados para dominios unidimensionales, se debía conseguir ambos tipos de convergencia (de tiempo y de malla) en dominios bidimensionales. Para lograr esto era requerido generalizar el modelo y los algoritmos desarrollados para este tipo de dominios.

2.1. Pruebas Híbridas 1D/2D

Antes de poder comenzar a trabajar con dominios 2D, y simular el fenómeno estudiado, era necesario contar con alguna forma de validación intermedia de los algoritmos entre los dominios unidimensionales y bidimensionales. Para esto, se diseñó un conjunto de pruebas en las cuales se extendió el dominio en una dimensión a uno en dos dimensiones agregando “filas” de celdas (anteriormente se disponía de una sola fila de celdas dispuestas horizontalmente). El flujo difusivo de las especies en estas pruebas se daría en la misma dirección que antes. Se puede observar en la Fig. 8-16 como era el dominio anterior y el nuevo, además de la dirección de flujo.

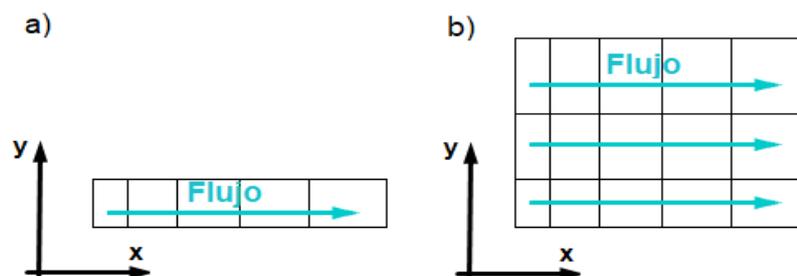


Fig. 8-16. a) Dominio utilizado en pruebas unidimensionales. b) Dominio a utilizar en pruebas híbridas 1D/2D para validar el funcionamiento de algoritmos.

Con una malla configurada de esta manera, se pretendía validar el funcionamiento de los algoritmos (en particular el asociado al precálculo de vecinos de celdas, descrito en la sección 5, capítulo VI) en un dominio donde existieran dos dimensiones espaciales. En este caso, se deberían obtener los mismos resultados que se obtenían en el caso unidimensional, debido a que el flujo es equivalente.

A continuación, se realizaron las simulaciones correspondientes para verificar qué resultados se obtenían con múltiples filas. Para realizar estas simulaciones, se conservaron los parámetros y modificaciones usadas para las pruebas anteriores de convergencia de malla. La única modificación realizada fue la de la cantidad de celdas configuradas a lo largo del eje X (a fin de evitar tiempos de simulación excesivos debido a las múltiples “filas”). En este caso se establecieron 3000 celdas horizontales.

En la **Fig. 8-17** se muestra la representación de ParaView del resultado obtenido para las 3 simulaciones (las 3 situaciones son indistinguibles entre sí en esta representación). Es importante notar que si bien se agregaron celdas en la dirección del eje Y (aumentando las filas), la dimensión total del dominio en el eje Y no fue modificada, razón por la cual no se puede observar un incremento en altura en la figura.

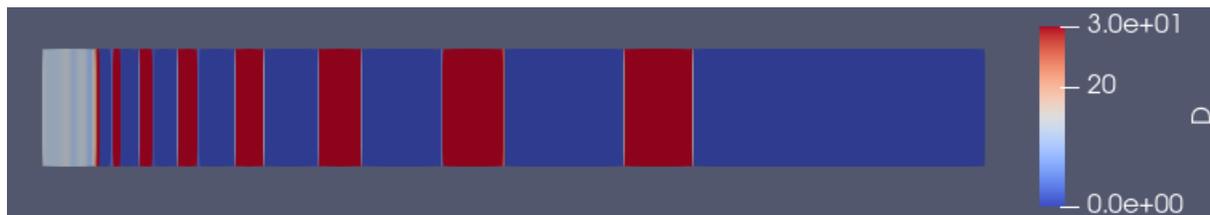


Fig. 8-17. Concentración de la especie D en el dominio al finalizar la simulación.

En la **Fig. 8-18** se puede observar una comparación entre el resultado para una sola fila y el resultado para dos casos donde existían 8 “filas” de celdas. La diferencia entre estos últimos fue que uno se configuró con una expansión de celdas a lo largo del eje vertical Y (además de la expansión horizontal ya existente de las pruebas de convergencia de malla). De esta forma se ponía a prueba también el funcionamiento del algoritmo en situaciones donde la expansión de celdas era en más de una dirección.

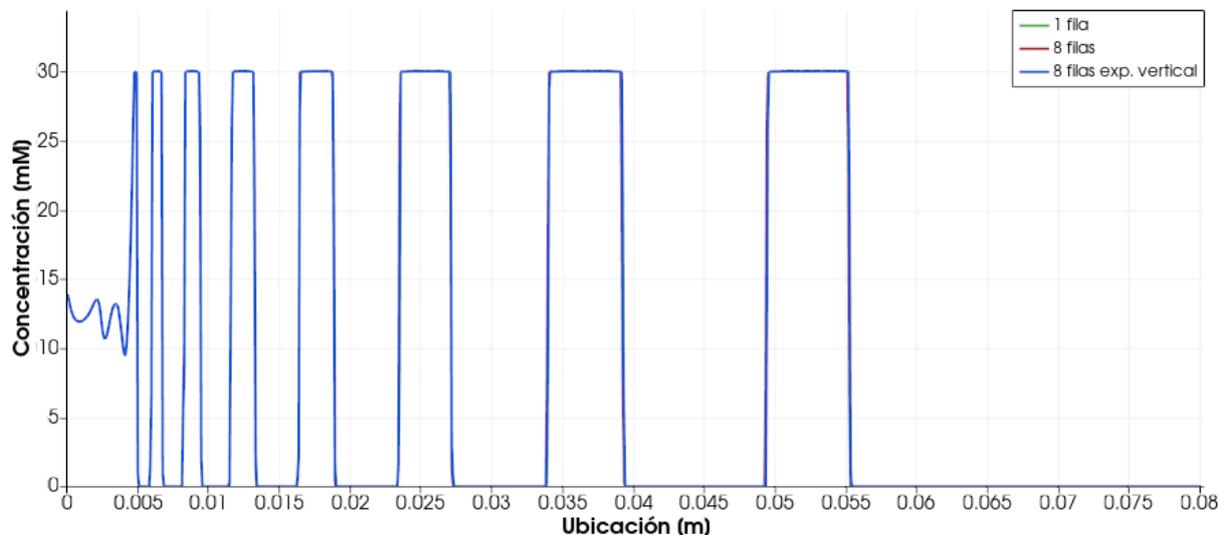


Fig. 8-18. Comparativa de resultado obtenido para un dominio unidimensional y dos casos similares con más de una fila en una segunda dimensión (ambos con 3000 celdas horizontales). Uno de los casos con 8 filas cuenta con expansión de celdas vertical (además de horizontal).

Como se puede ver en la **Fig. 8-18**, los tres resultados obtenidos son idénticos en sentidos prácticos. Esto demostró que los algoritmos no presentaban limitaciones al momento de trabajar con dominios donde existían

celdas dispuestas de forma bidimensional, y son suficientemente robustos como para entregar soluciones válidas en dichos escenarios.

2.2. Configuración de los casos de simulación

En este caso, la situación modelada ya no representa una probeta, sino que representa alguna superficie donde se distribuye uniformemente un gel saturado con la especie B. Luego, sobre esta capa de gel, se deja caer una gota de A para que reaccione y genere los patrones estudiados.

Para este tipo de dominios, la configuración de los casos debía ser cambiada para obtener un dominio bidimensional. Para hacerlo, se modifica el archivo `blockMeshDict` (presentado en el capítulo VI), definiendo una nueva cantidad de celdas, nuevos vértices, nuevos bloques de celdas, bordes y límites de dominios.

En las pruebas para convergencia de tiempo, el dominio se configuró para que estuviese compuesto por 20 mil celdas. Este número tan alto se justifica por el hecho de que hay una dimensión adicional, en la cual se debe contar con suficiente precisión. Este valor luego fue modificado para pruebas posteriores.

En el archivo `blockMeshDict`, se modificaron también las longitudes del dominio 2D. Los detalles son explicados en la siguiente sección.

Como se mencionó antes, en el Anexo E de este informe se puede encontrar ejemplificado el contenido de los archivos (incluido `blockMeshDict`) para un caso sencillo con un dominio bidimensional.

Por otro lado, la concentración de la especie A, ya no es aplicada en el mismo lugar que antes (porque el dominio cambió de forma). Ahora, la concentración está establecida sobre un borde del dominio que representaría el borde de una gota de A. Esta gota, en un experimento real sería colocada sobre un gel saturado de B. En este caso, el gel representa el resto del dominio y la concentración de B es la misma que antes. En la siguiente sección también se explica la situación en mayor detalle.

Los demás parámetros y configuraciones de los casos, se mantuvieron iguales a lo utilizado en las pruebas satisfactorias de convergencia de malla en casos 1D.

2.3. Estructura del dominio - Pruebas de convergencia temporal

En este escenario, el dominio está constituido por una malla bidimensional generada con herramientas propias de OpenFOAM (`blockMesh`). Luego de realizarse las simulaciones propiamente dichas, los resultados pueden ser cargados en ParaView para generar una representación gráfica de los mismos.

La estructura del dominio, usada para realizar las primeras pruebas sobre convergencia temporal en 2D, fue la que se ilustra en la **Fig. 8-16**. Para pruebas posteriores, nuevas estructuras fueron necesarias que se explicarán más adelante.

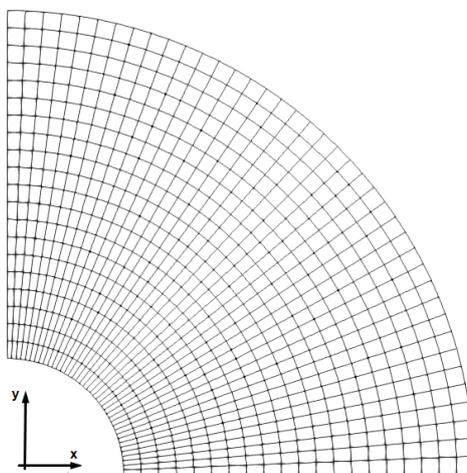


Fig. 8-16. Dominio utilizado para las pruebas de convergencia temporal.

Como se puede observar, con este dominio se pretende simular solo un cuarto de círculo, lo que representa un cuarto de una placa de Petri donde se distribuye el gel. En estos casos, se aprovecha la simetría del problema y del dominio para efectuar simplificaciones, a fin de reducir el dominio a simular, la cantidad de celdas necesarias y, fundamentalmente, los tiempos de ejecución. La solución al problema correspondiente a las porciones del dominio no contempladas se puede obtener simplemente reflejando la solución obtenida en el cuarto de círculo simulado.

Es oportuno mencionar que el espacio sin celdas en el origen de coordenadas corresponde al espacio ocupado por la gota de A mencionada previamente, que se sitúa sobre el gel embebido con la especie B en un experimento de laboratorio. Simular la gota en sí, o donde sitúa, no proporciona valor para el análisis de los patrones.

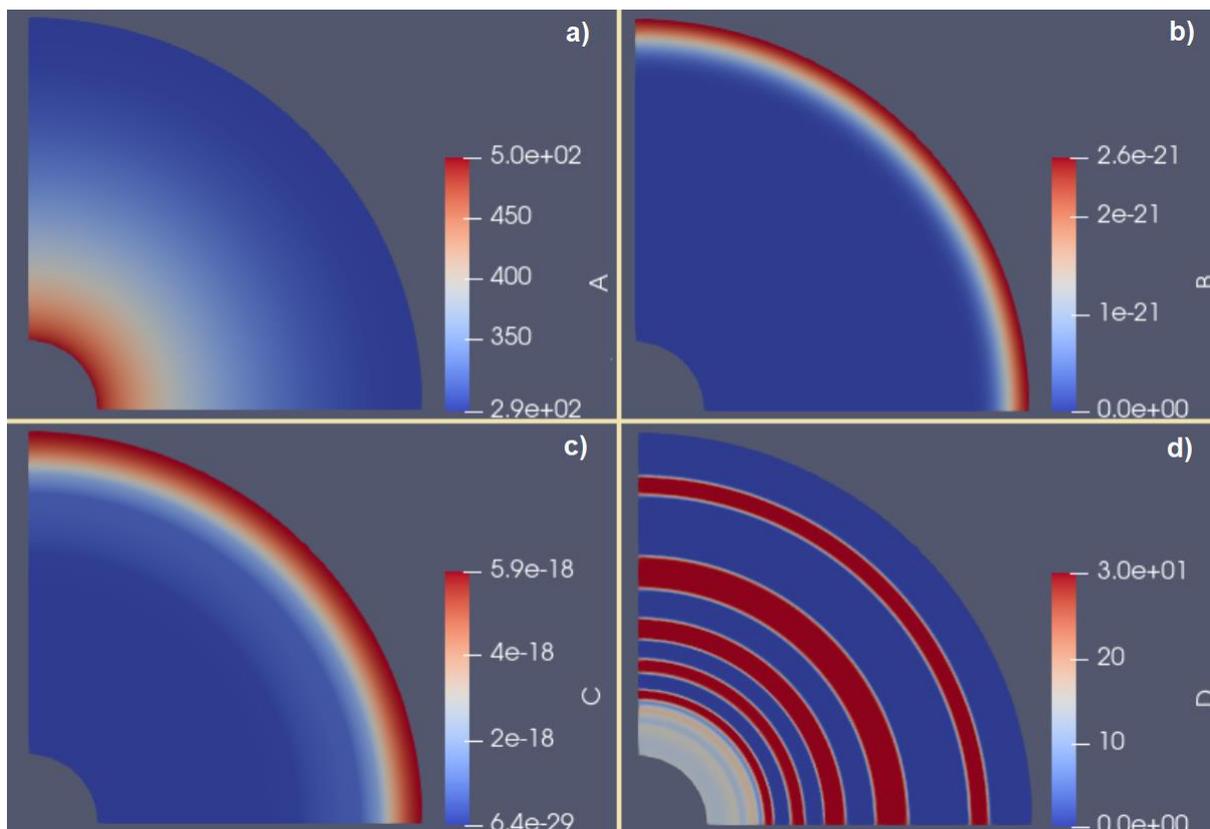
Por otro lado, la longitud radial del dominio fue configurada de 3 centímetros o 0.03 metros (a diferencia de los 8 centímetros de longitud del caso unidimensional) para reducir el dominio a simular (reduciendo también el costo computacional involucrado).

Como se mencionó previamente, en las pruebas para convergencia de tiempo, el dominio fue configurado con 20 mil celdas. Esto luego fue cambiado para analizar la convergencia de malla.

2.4. Funcionamiento del nuevo modelo generalizado a dos dimensiones

En la **Fig. 8-17**, se puede observar la representación gráfica generada por ParaView para los resultados de la ejecución de una simulación en un caso de ejemplo¹⁴.

Además, en la figura, se puede ver las concentraciones de las distintas especies expresadas como mol/m^3 (milimolar). Se incluye una referencia del valor de concentración de acuerdo al color a la derecha de cada cuadro.



¹⁴ Dependiendo del caso ejecutado, las bandas cambian de posición, ancho y cantidad, pero la forma general de los resultados se mantiene.

Fig. 8-17. Resultados generales del nuevo modelo, generalizado a dos dimensiones. Se visualizan las concentraciones de cada especie al final de la simulación.

Como se puede ver en la **Fig. 8-17**, al finalizar la simulación, la especie A ha difundido desde la gota que fue liberada sobre el gel (extremo izquierdo inferior en la imagen), reaccionando con la especie B a lo largo del gel (**Fig. 8-17 a**).

La especie B fue consumida casi totalmente en la reacción con A, y ha dado lugar a la formación de la especie C en distintos puntos (determinados por el modelo matemático), generando saturaciones de la misma en esos lugares (**Fig. 8-17 b**).

Finalmente, como se muestra en la **Fig. 8-17 c**, la precipitación de la especie C da lugar a la formación de la especie D, formando bandas siguiendo el patrón de los patrones de Liesegang conocidos, pero ahora en un dominio bidimensional.

Al igual que sucedía en el caso unidimensional (por una configuración similar de parámetros), la cantidad de B (y en parte, de tiempo simulado) resultó escasa para formar completamente la última banda (porque no se generó suficiente especie C y, en consecuencia, D). Esta banda no terminó de “ensancharse”, como se ve en la **Fig. 8-17 d**.

2.5. Convergencia temporal

En dominios bidimensionales, para poder corroborar la convergencia temporal, se propuso realizar pruebas similares a las realizadas en el caso unidimensional, variando el parámetro ΔT , para ver si a partir de cierto valor, los resultados ya no cambiaban de forma significativa.

Para poder usar el modelo previamente desarrollado, era necesario adaptar el mismo (y los relativos algoritmos) al nuevo tipo de dominios, generalizando su funcionamiento. El algoritmo encargado del correcto manejo del avance de tiempo de la simulación fue modificado y generalizado (durante el *sprint* 6, explicado en la sección 3 del capítulo V), para que pudiese cumplir su función en dominios bidimensionales (para configuraciones de dominio heterogéneas, donde las celdas pudieran ser de formas y tamaños distintos, incluso, en el mismo dominio).

Para este nuevo modelo y algoritmos generalizados, no fue necesario implementar parámetros adicionales a los que ya se agregaron hasta este punto.

En la **Fig. 8-18**, se muestran los resultados de ejecutar la simulación en el dominio mostrado en la **Fig. 8-16**. En la figura, el gráfico muestra la concentración de la especie D a lo largo del dominio, en casos con distintos valores asignados al parámetro ΔT . Los valores usados son los mismos que se utilizaron para conseguir convergencia de malla en dominios unidimensionales (sección 1.5 de este capítulo).

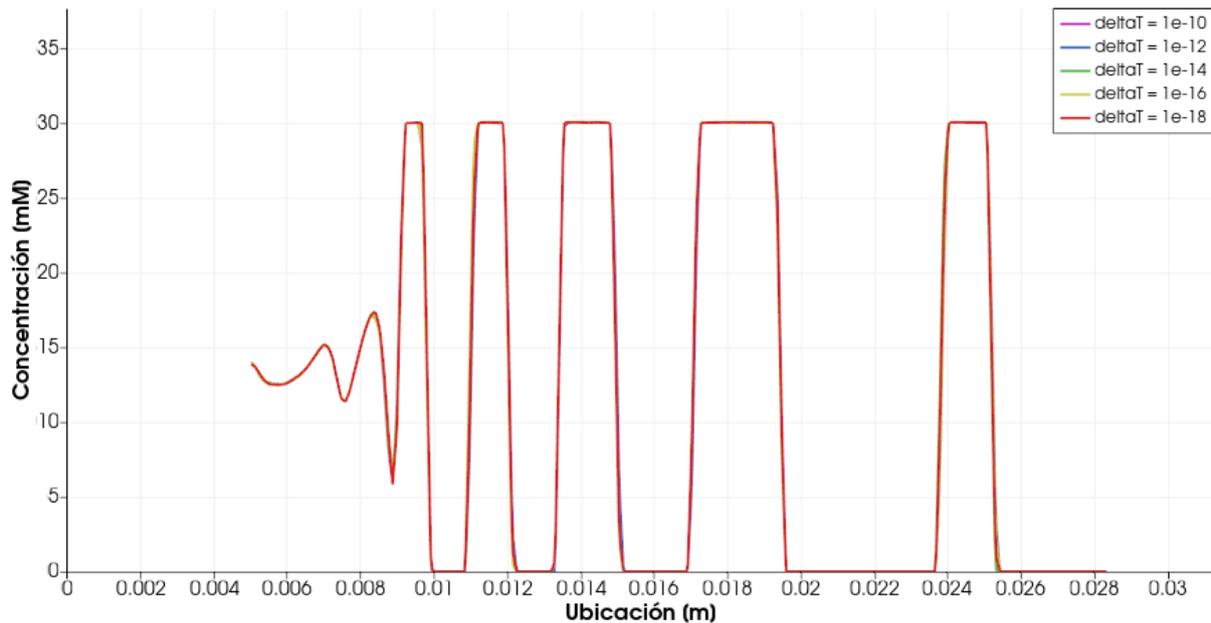


Fig. 8-18. Concentración de la especie D a lo largo del dominio para valores de Δt distintos en un dominio bidimensional. Se utilizaron algoritmos generalizados.

Es importante notar que, en la **Fig. 8-18**, la porción del gráfico al principio que no registra concentración de D, corresponde a la porción del dominio donde se ubica la gota de la especie A (porción que no es simulada).

En la figura se puede observar que el grado de convergencia temporal es muy alto, existiendo diferencias despreciables en la ubicación de las bandas (solo notable con un acercamiento considerable). Los resultados varían de forma despreciable a medida que se varía el parámetro Δt , por lo que se concluye que, con el modelo generalizado, también existe convergencia temporal.

2.6. Convergencia de malla

Los dominios bidimensionales presentaban una dificultad importante al momento de realizar un estudio completo sobre la convergencia de malla. Esta dificultad se debía al hecho de que los dominios, por el simple hecho de ser bidimensionales, deben contar con una cantidad de celdas mayor (a veces mucho mayor) para obtener resultados que se asemejen en calidad a los obtenidos en dominios 1D. Lo que esto provoca, es que las simulaciones sean mucho más costosas a nivel computacional para poder ser realizadas, extendiendo considerablemente los tiempos requeridos para obtener resultados.

Como consecuencia de esta dificultad, la convergencia de malla, no pudo ser evaluada únicamente como se hizo previamente (dominios 1D), es decir, incrementando la cantidad de celdas progresivamente hasta determinar que existía convergencia. En cambio, se procedió a analizar la convergencia teniendo en cuenta los siguientes hechos:

- Dada la configuración de los escenarios de los experimentos reales simulados, el flujo por difusión tiene la propiedad de ser completamente radial (no hay componente angular). Esto es así debido a que la difusión se produce desde la gota de A hacia el límite exterior del dominio. Por esta razón, las celdas recorridas de forma radial se comportan de forma tal que siguen el mismo comportamiento que la “fila” del caso unidimensional, donde existe convergencia de malla.

El comportamiento se puede observar representado en la **Fig. 8-19**.

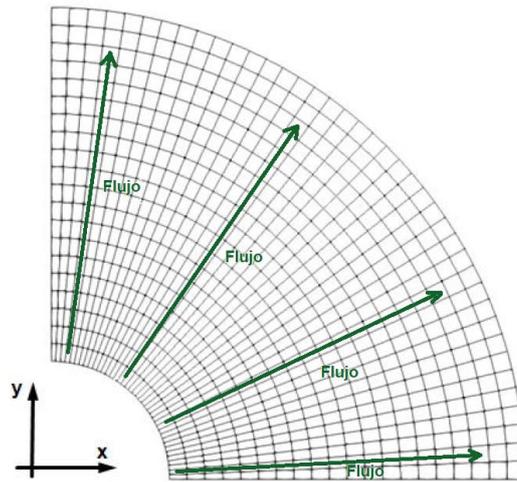


Fig. 8-19. Flujo por difusión completamente radial.

- Pruebas sobre mallas bidimensionales variadas, donde la forma, orientación y distribución de las celdas cambien (incluso dentro de una misma prueba), deberían ser capaces de generar patrones de Liesegang, verificando que el modelo y el algoritmo son independientes de la malla. También debería cumplirse esto al cambiar la cantidad de celdas configuradas en una malla.

Dado que el segundo punto necesitaba comprobación, se procedió a diseñar un conjunto de pruebas para evaluar la independencia de malla. En la siguiente sección, se muestran los resultados obtenidos con distintos tipos de mallas.

Las mallas presentadas en las secciones 2.7.2 a 2.7.5 de este capítulo, fueron pensadas/diseñadas con el propósito de disponer de un dominio rectangular para reflejar la geometría de los experimentos de laboratorio de la bibliografía.

2.7. Evaluación de independencia de malla

Las pruebas mostradas a continuación, fueron ejecutadas con los mismos parámetros y valores usados en las pruebas de convergencia de tiempo bidimensionales (sección 2.4 de este capítulo).

2.7.1. Malla Circular

Esta malla es la que se utilizó para la prueba de convergencia de tiempo en dominios bidimensionales. Se la puede ver representada nuevamente en la **Fig. 8-20**.

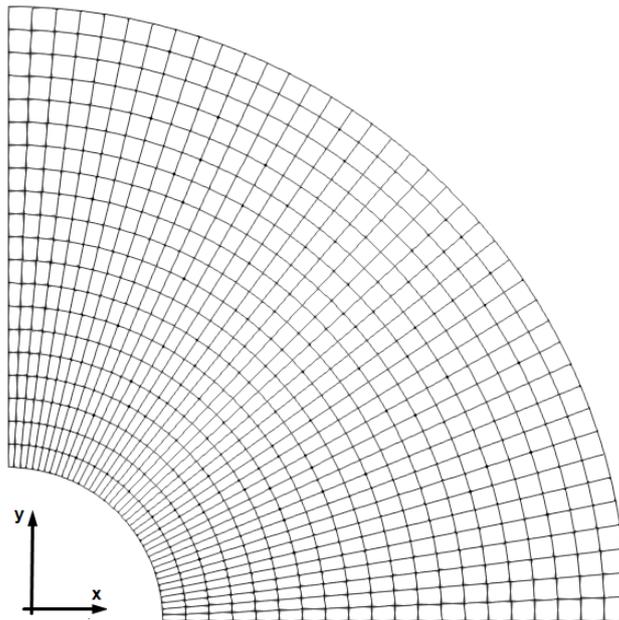


Fig. 8-20. Malla circular.

Los resultados obtenidos al ejecutar la simulación en esta malla son los que se pueden observar en la **Fig. 8-20**. Esta malla se configuró y ejecutó con aproximadamente 44 mil celdas (a diferencia de lo usado en las pruebas de convergencia temporal bidimensional).

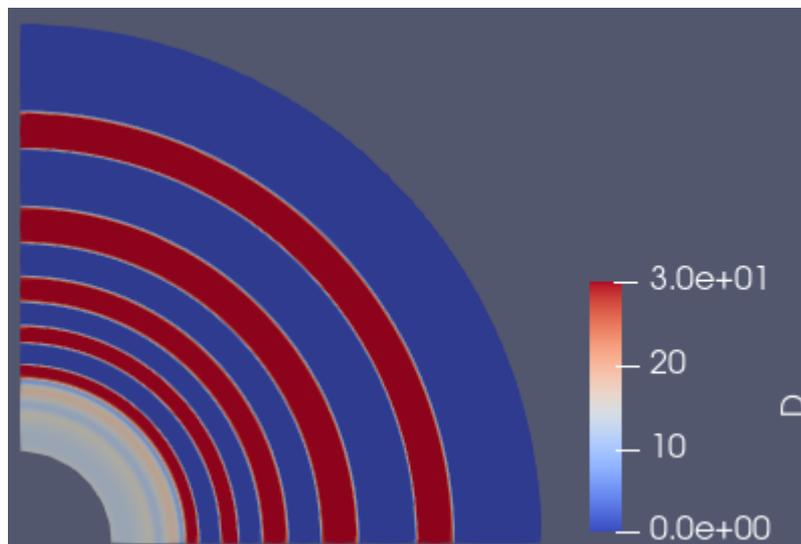


Fig. 8-21. Resultados de la simulación en una malla circular con aproximadamente 44 mil celdas.

Como se puede observar en la **Fig. 8-21**, los resultados presentan bandas perfectamente formadas. Sin embargo, para obtener este resultado no era necesario 44 mil celdas debido a que la forma, orientación y distribución de las mismas favorecen ampliamente la formación de este tipo de patrón. Es por esto que, si se hubiese usado una menor cantidad de celdas, los resultados probablemente habrían tenido la misma calidad. Se seleccionó este valor para intentar mantener un valor general para las pruebas entre distintas mallas; aunque como se verá y explicará más adelante, esto no es posible en ciertas mallas sin sufrir deformaciones indeseadas en las bandas.

2.7.2. Malla Combinada

Esta malla está diseñada y estructurada como se muestra en la **Fig. 8-22**.

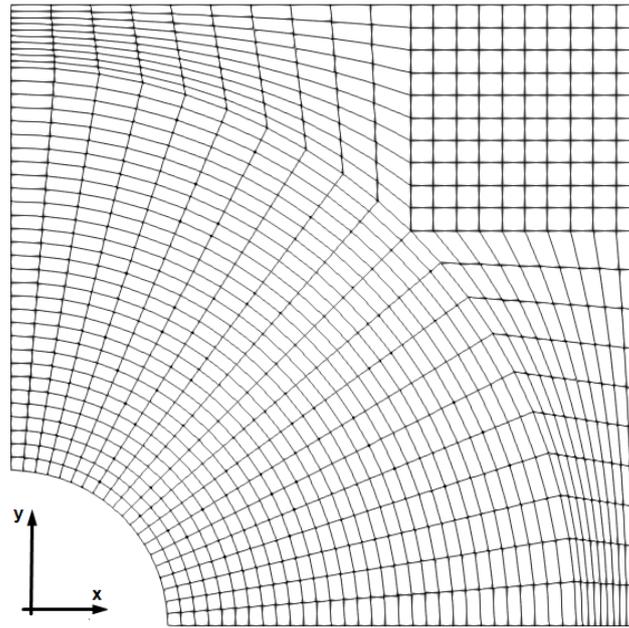


Fig. 8-22. Malla combinada.

Como puede observarse en la **Fig. 8-22**, se combinaron 3 tipos de celdas distintas en esta malla. En la prueba, el dominio se configuró con aproximadamente la misma cantidad de celdas que la anterior. El resultado obtenido se muestra en la **Fig. 8-23**.

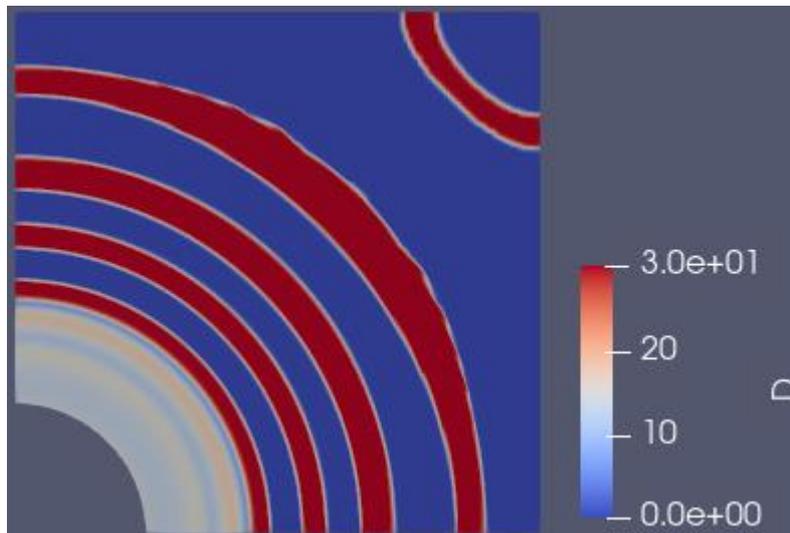


Fig. 8-23. Resultados de la simulación en una malla combinada con aproximadamente 44 mil celdas.

En la **Fig. 8-23**, puede observarse tres cuestiones importantes:

- Una inversión de la curvatura de la última banda: este efecto es causado únicamente por la configuración de límites del dominio y su forma. Se produce en bandas cuya distancia radial desde la posición de la gota de A es mayor a los lados del dominio. El efecto observado se origina por la velocidad de difusión de las especies y no existiría en un dominio no acotado.
- Un mayor ensanchamiento en la zona media de la anteúltima banda, causado por motivos relacionados a la inversión de la última banda. Los límites del dominio afectan el ensanchamiento de las porciones de la banda que se encuentran más cerca del borde derecho y superior del dominio. Se puede observar que esto no sucede en otras bandas.

- Una última banda más fina que lo esperado: esto se debe a que la especie B se consumió antes de poder terminarse completamente la formación de la última banda.

Más allá de los detalles mencionados, la simulación generó patrones de Liesegang como era esperado.

2.7.3. Malla V-circular

Esta malla se diseñó y estructuró como se muestra en la **Fig. 8-24**.

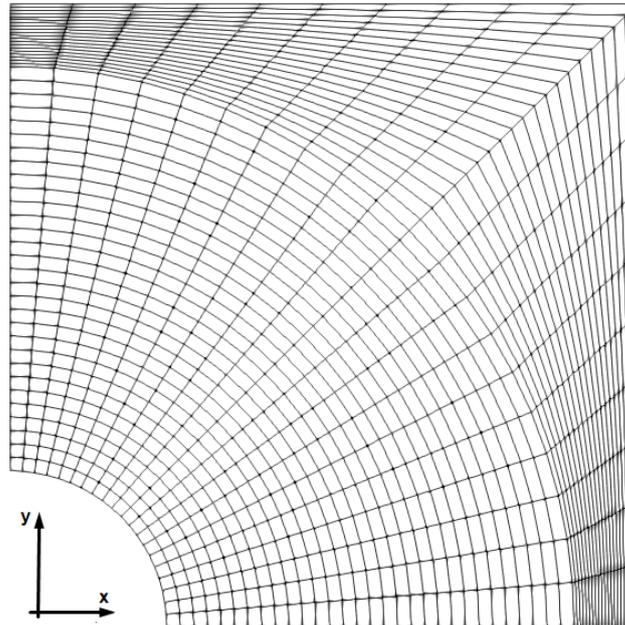


Fig. 8-24. Malla V-circular.

En este caso se combinaron 2 tipos de celdas distintas en esta malla. En la prueba, el dominio nuevamente se configuró con 44 mil celdas. El resultado obtenido se puede observar en la **Fig. 8-25**.

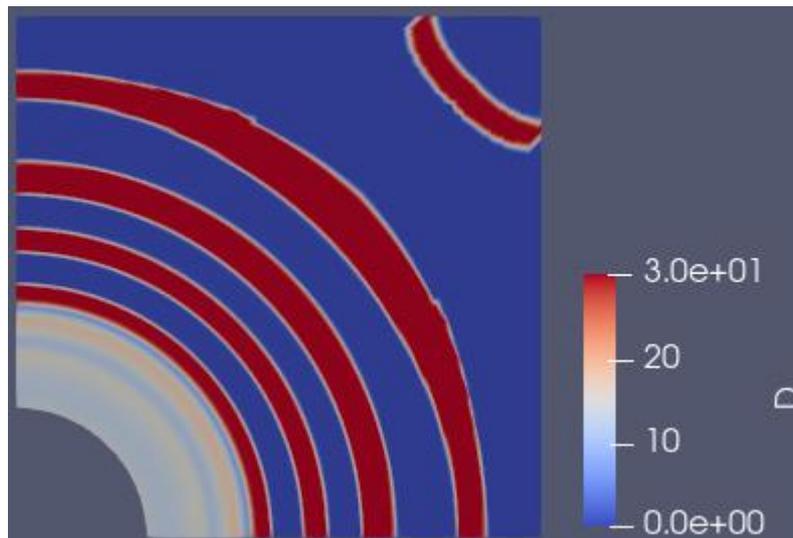


Fig. 8-25. Resultados de la simulación en una malla V-circular con aproximadamente 44 mil celdas.

En la figura, se puede observar que existe una “imperfección” en la banda final (además del cambio de orientación, explicado en la malla vista previamente). Esto es debido a una pérdida en la precisión al simular y realizar los cálculos relacionados con el fenómeno en esa sección. El origen de la falta de precisión, es que la forma de las celdas en esa porción del dominio no es recomendable para simular la forma de las bandas. Para

un resultado más fidedigno, la densidad de celdas en esa porción del dominio (capacidad provista por blockMesh) debería ser incrementada (achicando el tamaño de las celdas).

Otros detalles de la figura, como el ensanchamiento en la zona media de la anteúltima banda y el grosor de la última tienen la misma causa explicada en el tipo de malla anterior.

2.7.4. Malla V

Esta malla tiene la estructura y diseño representada en la Fig. **Fig. 8-26**.

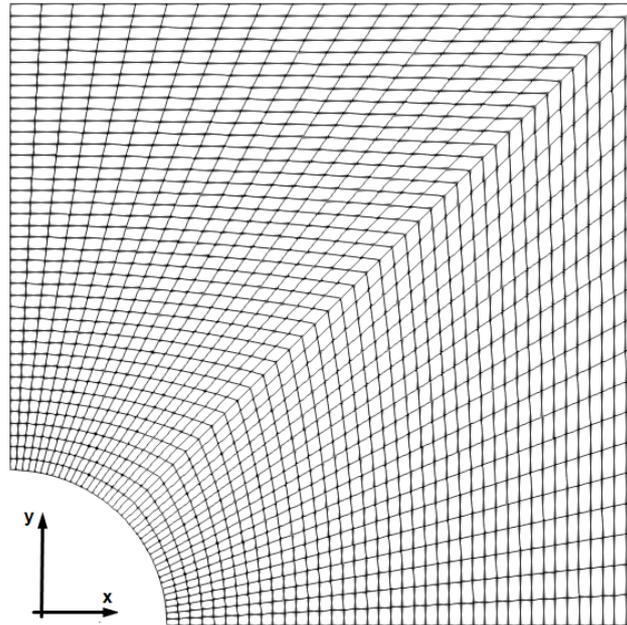


Fig. 8-26. Malla V.

En este caso la malla está compuesta solo de un tipo de celda. Dicho tipo de celdas es sumamente inadecuado para la simulación de este fenómeno y de las bandas debido a las características del fenómeno y la oblicuidad (skewness) que se observa en las celdas en ciertas partes.

Es sabido que las celdas demasiado oblicuas pueden disminuir la precisión y desestabilizar la solución de simulaciones [48]. Es por este motivo que este dominio requiere de mayor cantidad de celdas, de lo contrario, se da la existencia de serias distorsiones en las bandas. Teniendo esto en cuenta, esta malla fue generada con 80 mil celdas, con el objetivo de compensar los efectos de la oblicuidad de celdas, y obtener así bandas lo más definidas posible. El resultado obtenido se puede observar en la **Fig. 8-27**.

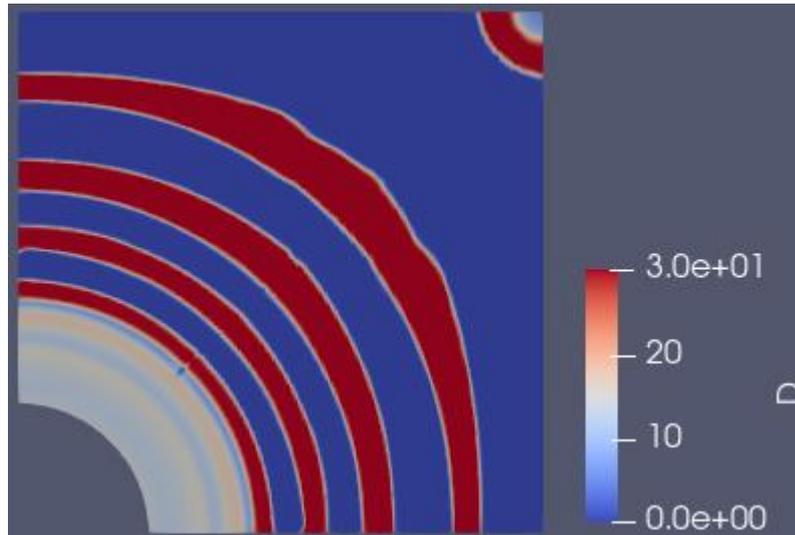


Fig. 8-27. Resultados de la simulación en una malla V-circular con aproximadamente 80 mil celdas.

Como se puede observar en la **Fig. 8-27**, una vez que la cantidad de celdas es suficientemente alta, los resultados muestran bandas sin distorsiones considerables. En general, si se utiliza una cantidad insuficiente de celdas, existirán distorsiones en las bandas formadas (esto puede verse ejemplificado para esta malla en el anexo D). Los resultados obtenidos se asemejan a los obtenidos para otras mallas. La diferencia radica en una posición de bandas algo distinta, lo que es natural debido a que se produjo un gran aumento en el número de celdas. Se esperaba que esos cambios sean cada vez menos importantes (lo cual se verificó con el siguiente tipo de mallas).

2.7.5. Mallas Gmsh

Las mallas anteriores ya eran suficientes para demostrar que, en dominios bidimensionales, el modelo y los algoritmos dan lugar a la generación correcta de patrones de Liesegang independientemente de la configuración de la malla. Sin embargo, se pretendía ir más allá, con casos más desafiantes y además verificar la convergencia de malla a medida que se aumentaba la cantidad de celdas.

Las nuevas mallas usadas en las pruebas fueron generadas con la herramienta Gmsh, que permite la generación de mallas totalmente desestructuradas para ser usadas con OpenFOAM. Se usaron mallas con distintas cantidades de celdas, para analizar los resultados y estudiar cómo eran afectados por las características de las celdas.

Las mallas más densas generadas con Gmsh contaban con 80 y 160 mil celdas. Con esta gran cantidad de celdas se pretendía tener suficiente información para poder analizar la convergencia de malla. En la **Fig. 8-28** se puede observar la malla con mayor cantidad de celdas utilizada durante el proyecto (160 mil celdas).

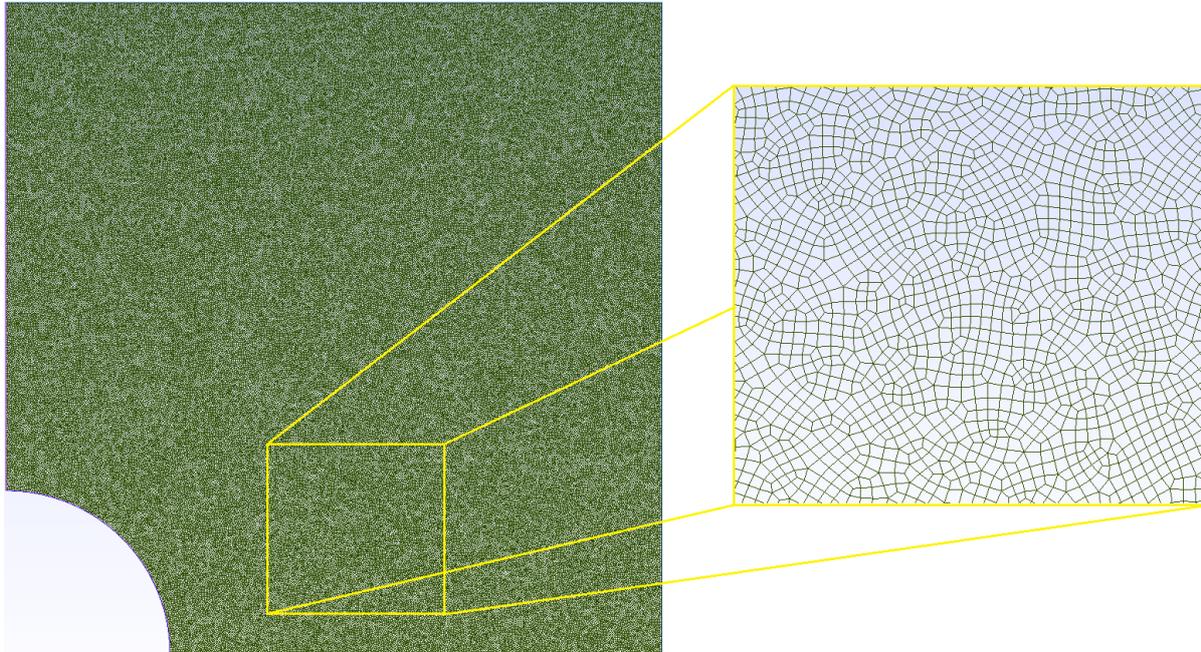


Fig. 8-28. Malla generada con Gmsh con 160 mil celdas.

Como se puede ver, la malla mostrada en la **Fig. 8-28** es no estructurada (concepto introducido en la sección 6 del capítulo II), con celdas totalmente variables en cuanto a tamaño, orientación y forma. Esto muestra porque se considera a estos como los casos más desafiantes probados. Un algoritmo que funcione sobre estas mallas, es esperable que lo haga sobre cualquier otra.

Los resultados obtenidos en los casos mencionados (80 mil y 160 mil celdas) se muestran en la **Fig. 8-29**. Ambos casos se corrieron hasta los 30 mil segundos (aproximadamente 8 horas), debido a que simular hasta ese punto proporcionaba suficiente información como para extraer conclusiones en relación a la convergencia de malla.

En la **Fig. 8-29** (ambas imágenes), el fenómeno que se evidencia en las esquinas superior izquierda e inferior derecha del dominio, se debe a los límites impuestos sobre el dominio (lado derecho y lado superior). A medida que la especie A difunde y reacciona con B, se origina una acumulación excesiva de la especie intermedia C en esos puntos. Al acumularse suficiente C, se produce la formación de la especie D (cuando C se precipita). Como se mencionó antes, el fenómeno está relacionado con los límites, al igual que la inversión de bandas en casos previos y su efecto aumenta o disminuye cuanto más cerca o lejos del límite del dominio se encuentran las bandas formadas en la simulación.

Por otro lado, como puede verse al comparar las imágenes de la figura **Fig. 8-29** (80 mil y 160 mil respectivamente), ambos resultados son casi idénticos. La diferencia entre ambos radica en 2 cosas:

- El fenómeno de las esquinas mencionado previamente, es un poco mayor en tamaño en el caso de 160 mil celdas. Esto es una pequeña variación en los resultados por la gran variación en cantidad de celdas. Este fenómeno cambiaría cada vez menos cuando se incrementan las celdas (de hecho, fue verificado en pruebas con menos de 80 mil celdas). Esto es simplemente una formación del mismo tipo que las bandas, y está sujeto al mismo comportamiento.
- En la región más cercana a la gota, donde aún no se formaron bandas, se puede observar puntos o “huecos” donde la concentración es menor (con un color azul). Este efecto es producido por la irregularidad de las celdas y se reduce simplemente aumentando la cantidad de celdas para mejorar la precisión, como puede verse al comparar los dos resultados.

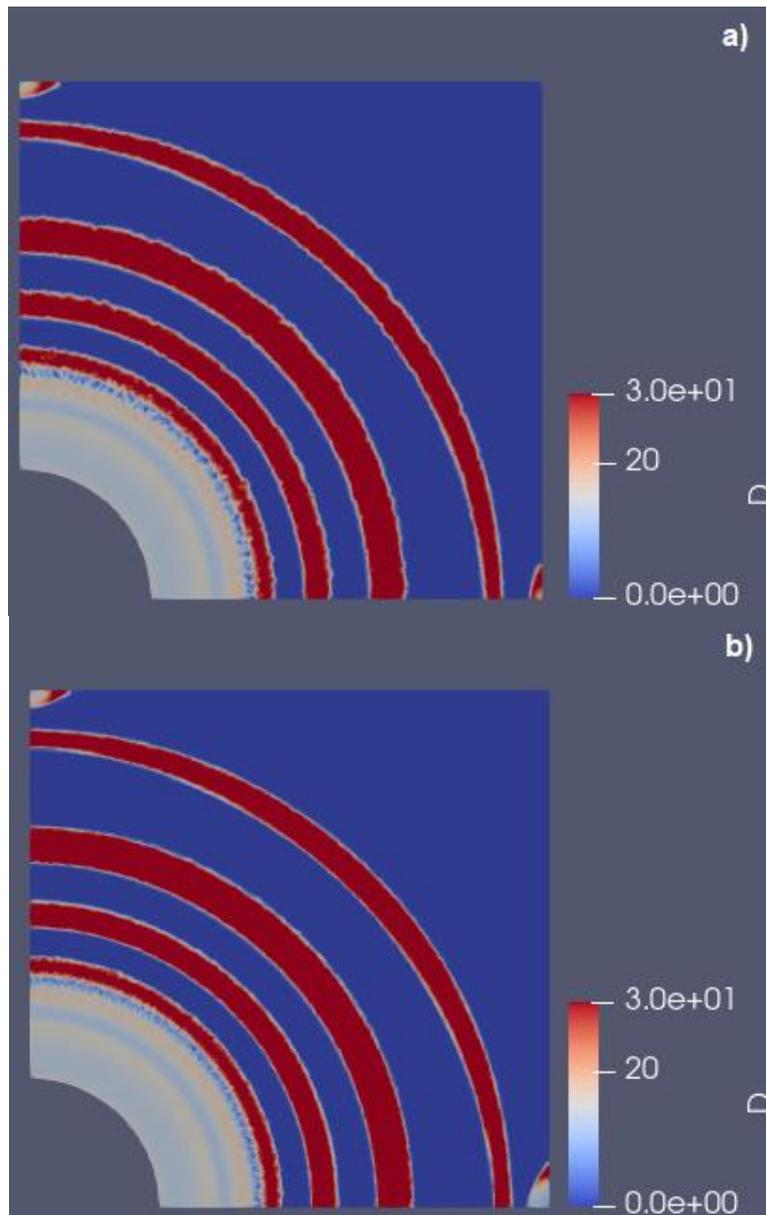


Fig. 8-29. Resultados de la simulación en una malla construida con Gmsh, con a) 80 mil celdas y b) 160 mil celdas.

Para un análisis más detallado de la posición de las bandas, se muestra en la **Fig. 8-30** un gráfico de concentraciones de la especie D en los dos casos mostrados.

A partir de la posición de las bandas, se puede decir que los resultados parecen converger a uno solo, donde las bandas (además de picos y valles de concentración) están bien definidas. El “ruido” observado en el caso de 80 mil celdas antes de la primera banda se debe a los “agujeros” antes explicados.

Un último detalle, que se considera importante mencionar, es que las pruebas realizadas con las mallas de 80 mil y de 160 mil celdas fueron muy costosas en términos computacionales (prolongando los tiempos de simulación). En el caso más grande, habiendo simulado solo 30 mil segundos (aproximadamente 8 horas), la simulación demoró 3 días en completarse.

En este caso, los tiempos no solo están afectados por la cantidad de celdas, sino que, cuando se utilizan mallas de Gmsh, muchas ventajas relativas a la simetría del dominio no pueden ser aplicadas por cómo se estructura la malla y la forma de sus celdas.

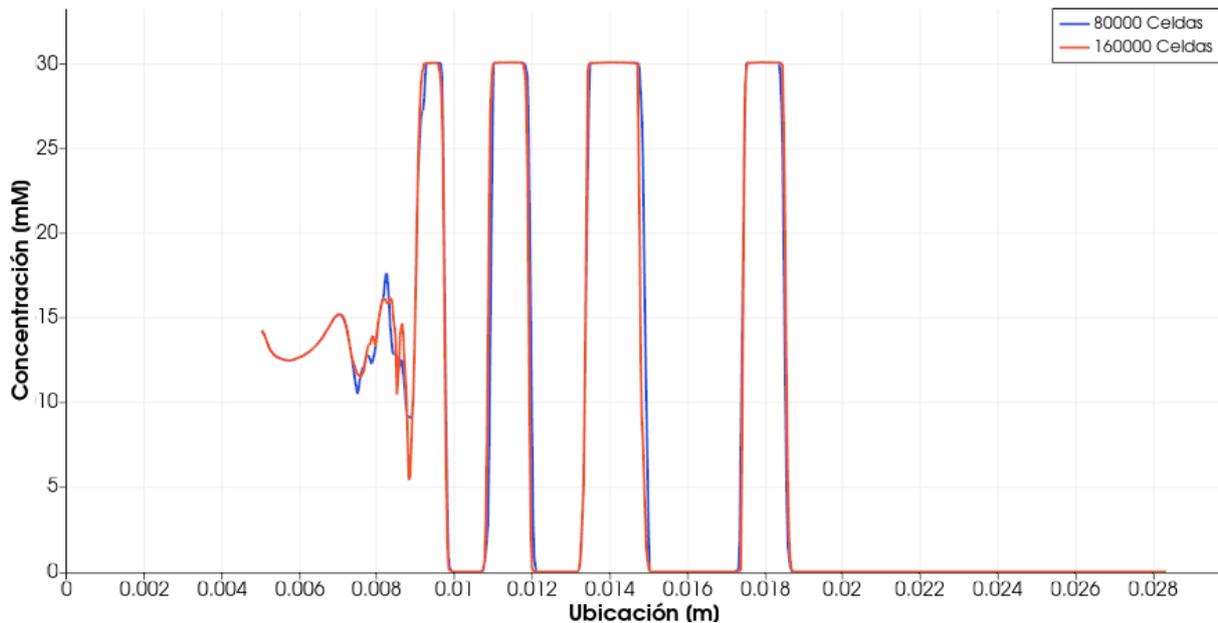


Fig. 8-30. Concentración de la especie D a lo largo del dominio para las mallas de Gmsh con 80 mil y 160 mil celdas.

Por otro lado, como existen celdas más pequeñas que el resto (como se ve en la Fig. 8-30), el cálculo de los pasos de tiempo (explicado en el capítulo VI) va a estar influenciado por estas y no las otras, lo cual es el funcionamiento deseado y correcto para el fenómeno simulado. Esto es otra evidencia de que el algoritmo es capaz de funcionar en situaciones donde las propiedades de las celdas que forman el dominio son sumamente heterogéneas.

2.7.6. Conclusión de convergencia de malla

Como se demostró en las pruebas con distintas mallas, el modelo y algoritmos generalizados para dominios bidimensionales presentan un gran nivel de convergencia de malla. Para mallas totalmente distintas los resultados siempre muestran los característicos patrones de Liesegang.

Por otro lado, los resultados entre mallas diferentes comparten una similitud, que si bien no es perfecta (debido a las características de las celdas), es bastante buena y, cuando la precisión utilizada (determinada por el número de celdas) es suficientemente alta, los resultados son prácticamente los mismos independientemente de la malla.

3. Reproducción de resultados de bibliografía específica

Una vez alcanzadas las convergencias en casos bidimensionales, el siguiente paso para verificar la calidad de la solución desarrollada, era intentar reproducir casos de experimentos no simulados obtenidos de bibliografía. En este caso, se trabajó con los resultados presentados en el artículo de Jiang y Sakurai [33], donde fue estudiada la formación de patrones ultrafinos de Liesegang.

3.1. Determinación de parámetros - Configuración casos

En el trabajo de Jiang y Sakurai, algunos de los parámetros requeridos, para realizar una simulación con el método y algoritmos desarrollados, no estaban disponibles directamente y fue necesario obtenerlos y/o estimarlos de otros datos (con ayuda del director y codirector).

En el caso de algunos parámetros, no fue posible obtenerlos ni estimarlos, dado que no había información suficiente en los datos proporcionados en el trabajo de referencia. El objetivo de los experimentos y el análisis realizado allí, no justificaba la inclusión de determinados datos sobre las condiciones de los experimentos realizados, dificultando la reproducción de estos.

Como consecuencia de este problema, se procedió a conseguir los parámetros faltantes de forma empírica utilizando las herramientas desarrolladas. Se realizaron diversas simulaciones, valiéndose de conceptos teóricos sobre el comportamiento y causas del fenómeno estudiado, así como también de los estudios realizados en pruebas anteriores. La idea era lograr recrear los resultados de un experimento elegido, realizado por Jiang y Sakurai, con el mayor grado de similitud posible.

Finalmente, los parámetros que permitieron alcanzar los mejores resultados (en similitud) fueron los mostrados en la **Tabla 8-5**.

Parámetro/Variable	Valor	Observaciones
Concentración especie A	0.5 [molar]	Valor aplicado como condición de borde del dominio que representa el borde de la gota. El resto es 0 molar.
Concentración especie B	0.14 [molar]	Mismo valor en todo el dominio. Estimado del experimento elegido para replicar (donde se especifica que el gel con B tenía una concentración al 3% de masa)
Dimensiones del dominio	10 × 10 [mm]	Dominio rectangular.
Tiempo de fin de simulación (endTime)	7200 [seg]	El experimento elegido se llevó a cabo durante 2 horas.
Cantidad de celdas	60000	Se considera un valor que permite alcanzar una precisión y tiempos de cómputo aceptables. Esta es la cantidad total de celdas en el dominio bidimensional. ¹⁵
Relación de expansión de celdas (expR)	20	Se considera que este valor proporciona la precisión adecuada al principio del dominio.
Concentración máx. especie D en bandas (rho)	0.450 [molar]	Este valor fue obtenido empíricamente.
Valor de punto de floculación c* (cs).	0.090 [molar]	Este valor fue obtenido empíricamente.
Constantes de velocidad para la coagulación (k)	1.3 [1/seg]	Este valor fue obtenido empíricamente.
Constantes de velocidad para la formación de precipitado auto catalítico (k)	1.3 [molar/seg]	Este valor fue obtenido empíricamente.
Radio de cristalización (r)	0.0000975 [m]	Si bien este valor se obtuvo empíricamente, de otros datos se estimaba que el radio debía ser menor que antes.
Concentración de referencia de la especie A (refValA)	0.5 [molar]	
Concentración de referencia de la especie B (refValB)	0.14 [molar]	
Concentración de referencia de la especie C (refValC)	0.09 [molar]	Este valor se estableció debido al punto de floculación.

Tabla 8-5. Parámetros que permitieron replicar de mejor manera el experimento elegido.

Los parámetros finalmente seleccionados tienen como fundamentación de que son el conjunto que nos permitió una mayor semejanza fisicoquímica respecto a los problemas estudiados.

¹⁵ Este valor es notablemente más grande que el usado en la mayoría de los casos anteriores dado que existe una dimensión adicional, en la cual también se debe contar con suficiente precisión.

Si bien este conjunto de valores permite alcanzar resultados muy similares a los del experimento de contraste, se sabe que existen otros (encontrados durante las pruebas). Es posible también, que otros conjuntos de valores puedan acercarse más a los resultados del experimento. Sin embargo, existen muchas combinaciones posibles, entre las numerosas variables involucradas, por lo que no es posible verificar este hecho.

3.2. Estructura del dominio

Para estas pruebas se seleccionó la malla bidimensional combinada, usada en las pruebas de convergencia de malla bidimensionales. Esta malla, además de ser rectangular, permite alcanzar resultados suficientemente precisos con una cantidad de celdas relativamente baja y con costos computacionales no tan excesivos, para este problema.

Como se mencionó antes, el dominio se configuró para que tuviera dimensiones de 10x10 mm y 60000 celdas.

3.3. Resultados y comparativa

Entre los experimentos realizados en laboratorio del artículo, existían dos resultados válidos para ser comparados con lo obtenido a través de las simulaciones, debido a la similitud de las condiciones bajo las cuales se llevaron a cabo esos experimentos. En las **Fig. 8-31** y **8-32** se realiza la comparación entre los resultados de laboratorio y lo obtenido por medio de la simulación del fenómeno estudiado. En la parte derecha de ambas figuras, se puede observar los resultados de laboratorio y en la izquierda lo obtenido a partir de la simulación. La visualización de lo simulado (**Fig. 8-31 a** y **Fig. 8-32 a**) fue orientada intencionalmente para facilitar la comparación con la fotografía del experimento de laboratorio (se incluyó una vista del dominio simulado por el mismo motivo).

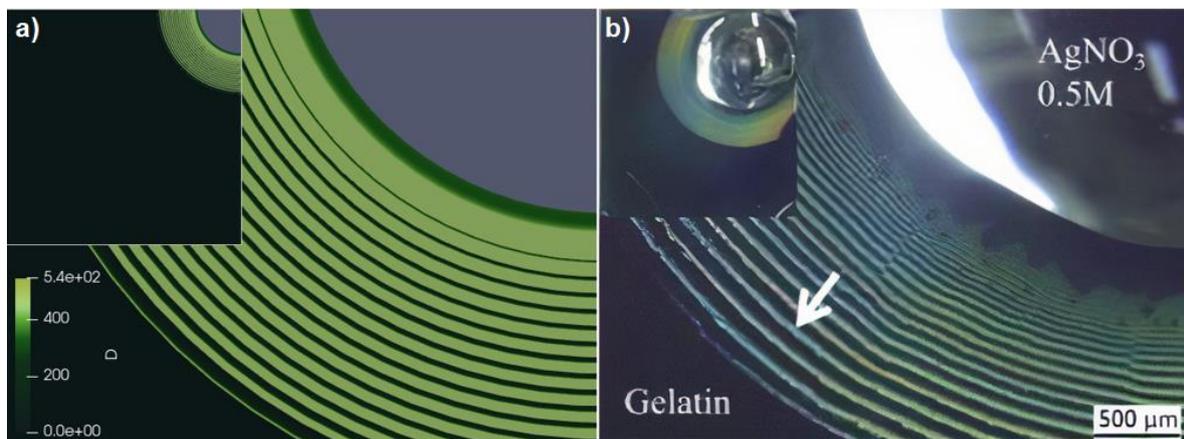


Fig. 8-31. a) Resultado obtenido por medio de la simulación. b) Resultado reportado en el artículo para el experimento de laboratorio. La imagen fue adaptada del artículo.

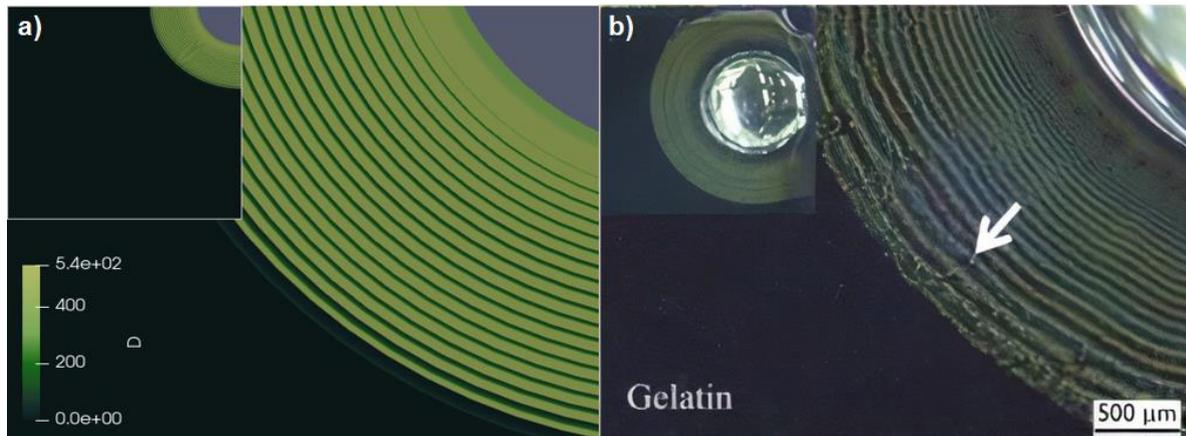


Fig. 8-32. a) Resultado obtenido por medio de la simulación. b) Resultado reportado en el artículo para el experimento de laboratorio. La imagen fue adaptada del artículo.

Como se puede apreciar al observar las figuras, los resultados obtenidos cuentan con un alto grado de similitud en la disposición y forma de las bandas. Se puede observar que se ha logrado la formación de una cantidad de bandas similar al experimento de contraste (contando bandas bien definidas en la fotografía proporcionada). Además, se observa como la separación entre bandas se asemeja a la encontrada en el experimento de laboratorio.

Finalmente, se puede apreciar cómo los niveles de concentración de D (indicados de acuerdo a la escala de colores), en las vecindades del borde circular del dominio (que representa el límite de la gota), son similares a lo observado en las fotografías originales del experimento elegido, en las regiones inmediatas a la gota.

Si bien existen algunas diferencias entre lo simulado y lo obtenido en el experimento, esto es directamente justificable a partir del hecho de que algunos parámetros no eran conocidos con exactitud (o directamente desconocidos) y debieron ser estimados. Es probable que, con una mayor disponibilidad de información, sea posible obtener resultados mucho más fieles al experimento de Jiang y Sakurai.

4. Validación de leyes en resultados

Como fue expresado en el capítulo de herramientas y tecnologías (capítulo III), para evaluar y validar la calidad de los resultados obtenidos en las simulaciones, se desarrolló un aplicativo en Python. Este aplicativo, interactuando con la interfaz programable de ParaView, realizaba la lectura e interpretación de las salidas de la simulación para proporcionar mediciones más avanzadas de los casos.

Los valores generados por el aplicativo permitieron contrastar estos con algunas leyes de Liesegang (introducidas en el capítulo II). De esta forma se realizó un análisis más detallado de los resultados.

El aplicativo genera los siguientes datos:

- La constante de proporcionalidad de la Ley de Espaciado (capítulo II, sección 5.1).
- La constante de proporcionalidad de la Ley de Ancho (capítulo II, sección 5.2).
- Mínimo, máximo y promedio de las constantes mencionadas anteriormente.
- La cantidad de bandas generadas.
- Las posiciones de las bandas, indicando el inicio y fin.

Este aplicativo, fue utilizado para validar los resultados de las pruebas realizadas durante el transcurso del proyecto. Dichas pruebas cumplían lo que establecen las leyes de los patrones de Liesegang.

Por otro lado, el aplicativo fue especialmente usado como apoyo en las tareas de verificación empírica, cuando se quisieron replicar los resultados de experimentos no simulados (sección 3 de este capítulo). El aplicativo ayudó a encontrar los parámetros faltantes (y ajustar los estimados), para adecuar los resultados a lo que se expone en el trabajo de Jiang y Sakurai.

El resultado de ejecutar el aplicativo con el caso simulado de la sección 3.3 de este capítulo (**Fig. 8-31 a**), es el que se muestra en la **Fig. 8-33**.

```

VisRTX 0.1.6, using devices:
 0: GeForce GTX 1070 (Total: 8.6 GB, Available: 7.1 GB)
Calculated rho 450.07
  Begin,      End,      Width,      Width law,      Space law
 0.000146,   0.000299,   0.000154,   nan,            nan
 0.000321,   0.000396,   0.000075,   0.490554,      1.323087
 0.000425,   0.000492,   0.000067,   0.883134,      1.241919
 0.000524,   0.000588,   0.000063,   0.951880,      1.194795
 0.000622,   0.000683,   0.000061,   0.968404,      1.162526
 0.000719,   0.000778,   0.000059,   0.969005,      1.139072
 0.000815,   0.000871,   0.000056,   0.944444,      1.119907
 0.000911,   0.000968,   0.000056,   1.000000,      1.110397
 0.001008,   0.001065,   0.000056,   1.007130,      1.100248
 0.001105,   0.001161,   0.000056,   0.989381,      1.090644
 0.001203,   0.001258,   0.000056,   0.982111,      1.083628
 0.001303,   0.001355,   0.000055,   0.947177,      1.077094
 0.001409,   0.001452,   0.000052,   0.817308,      1.071060
 0.001540,   0.001549,   0.000043,   0.491551,      1.066896
 0.001620,   0.001624,   0.000009,   0.214118,      1.063177
  Min,      Min,      Min,      0.214118,      1.063177
  Max,      Max,      Max,      1.007130,      1.323087
  Average,  Average,  Average,  0.777030,      1.056300
Fin primer banda 0.0002993
Fin ultima banda 0.0016241
Numero de bandas: 15

```

Fig. 8-32. Resultado de ejecutar el aplicativo desarrollado con Python.

En la **Fig. 8-32** se puede observar los cálculos realizados sobre las bandas. Es importante notar que las últimas 3 bandas no están formadas completamente. Además, las primeras 2 bandas son agrupación de bandas muy delgadas, y no sería correcto tomarlas en consideración para las conclusiones.

En el artículo de Jiang y Sakurai, se especifica que la constante de proporcionalidad de la Ley de Espaciado medida era de 1.07. En la simulación, la constante medida se aproxima a dicho valor, como se puede observar en la columna Space Law.

Si bien lo obtenido a través de la simulación es equivalente a lo que se presentaba en los experimentos no simulados, como se dijo antes, mejores resultados podrían alcanzarse con más información sobre las condiciones (finales y de realización) de los experimentos.

Mayores detalles sobre el funcionamiento y ejecución del aplicativo se pueden encontrar adjunto en el Anexo C de este informe.

VIII. Conclusiones

Como se demostró en el último capítulo, en este proyecto fue posible desarrollar un modelo y un conjunto de algoritmos robustos, para la plataforma OpenFOAM, que permiten simular de forma correcta la formación de patrones de Liesegang. La validez de los resultados obtenidos se mantiene tanto en dominios unidimensionales (1D) como en dominios bidimensionales (2D).

En las numerosas pruebas realizadas sobre el modelo y los algoritmos se pudo mostrar que estos desarrollos son capaces de funcionar adecuadamente en variados dominios y con condiciones de simulación muy amplias.

La aplicación desarrollada se encuentra bajo licencia de código abierto y está disponible en un repositorio público¹⁶, para que sea accesible a futuros usuarios. Esta herramienta es la primera que reúne las características de:

- Ser capaz de simular correctamente el fenómeno de Liesegang en dominios uni y bidimensionales.
- Estar implementada en OpenFOAM.
- Estar claramente documentada y ser de acceso público.

Es por esto, que se espera que el trabajo realizado constituya una contribución importante para la comunidad de OpenFOAM e investigadores, cuyo campo de trabajo se relacione a los patrones de Liesegang.

Si bien el objetivo específico de publicar los resultados obtenidos en una revista o congreso científico de alto impacto no fue concretado al momento de elaborar este informe (debido a la situación sanitaria mundial actual), está planeado realizarlo próximamente.

Sin duda, el proyecto brindó experiencia a los alumnos en las actividades de investigación, desarrollo, generación de documentación y gestión de proyectos. Se espera que esta experiencia sea de utilidad para los mismos al momento de ejercer en el ámbito profesional.

Por otro lado, la documentación continúa durante el avance del proyecto y la materialización de un informe detallado a partir de lo documentado, brindará a los alumnos una base para la futura generación de documentos científicos relacionados a este proyecto u otros.

Realizar este proyecto y alcanzar los objetivos propuestos, demostró ser una tarea muy desafiante debido a la incertidumbre que lo caracterizó en ciertas etapas y a la temática en sí. Sin embargo, los resultados muestran como la preparación adquirida por los alumnos durante el transcurso de la carrera, los preparó para enfrentar un problema, de un dominio que conocían poco inicialmente, y generar una solución. Dicha solución, tiene el potencial de ser una valiosa herramienta para profesionales del área en el futuro.

1. Trabajos futuros

En base al trabajo realizado durante este proyecto, se pueden proponer múltiples trabajos a futuro posibles.

En primer lugar, como se mostró a lo largo del informe, los dominios soportados por el modelo y algoritmos desarrollados deben ser uni o bidimensionales, dejando abierta la posibilidad de futuros desarrollos sobre dominios tridimensionales (3D), para los cuales será necesario generalizar los algoritmos.

En cuanto a lo desarrollado, existe interés en realizar una mayor cantidad de pruebas apuntadas a replicar los resultados de experimentos de laboratorio. En estas, los parámetros desconocidos en el caso estudiado previamente, serían provistos por los investigadores que realicen dichos experimentos.

En cuanto a las pruebas realizadas, se espera poder realizar una validación de la ley de Matalon-Packter en sus resultados, debido a que solo las 3 primeras leyes, de la sección 5 del capítulo II, pudieron ser validadas. La ley de Matalon-Packter no se validó debido a que implicaría desarrollos de algoritmos adicionales, que no

¹⁶ Indicado en el Anexo A de este informe.

podieron llevarse a cabo en el marco del cronograma original del proyecto. Es mencionable, que (a diferencia de las leyes de ancho y espaciamiento) dicha ley, pasó a conocimiento del equipo durante la implementación de los algoritmos.

Otro trabajo futuro de interés, es la posibilidad de añadir compatibilidad con la ejecución en paralelo, debido a que el método y algoritmos desarrollados actualmente no son capaces de aprovechar las ventajas del paralelismo (donde distintos procesadores trabajan con partes del dominio diferentes¹⁷) para entregar resultados válidos. Disponer de dicha capacidad permitiría la ejecución de las simulaciones en un clúster de cómputo, con una considerable reducción en los tiempos de espera para obtener los resultados.

Por otro lado, como se definió cuando se planeó este proyecto, se buscará realizar una publicación de los resultados obtenidos en un congreso científico o revista de alto impacto como la revista *Langmuir*.

Finalmente, se planea realizar una integración de los algoritmos y modelos aquí desarrollados en la herramienta *Fronts*, que constituye una librería numérica de Python para problemas de difusión no lineal, basada en la transformación de Boltzmann. Dicha herramienta fue desarrollada por el codirector del proyecto y existe un gran interés en poder realizar dicho trabajo.

¹⁷ Esta técnica se denomina descomposición de dominio.

BIBLIOGRAFÍA

- [1] A. A. B. Pritsker, *Introduction to simulation and Slam II*, 3ra ed., New York, EEUU: Wiley, 1986.
- [2] H. Hensch, *Crystals in Gels and Liesegang Rings*. Cambridge, Reino Unido: Cambridge Univ. Press, 1988.
- [3] A. J. Growle, D. C. Lueker, H. S. Gaskill Jr., “Periodic (Liesegang) Precipitation of Chemicals”, *Nature*, vol. 199, no. 4893, pp. 623-624, 1963.
- [4] OpenFOAM Foundation. *OpenFOAM*. [En línea]. Disponible en: <https://www.openfoam.org/>, accedido 2020.
- [5] A. M. Turing, “The Chemical Basis of Morphogenesis”, *Philos. Trans. Roy. Soc. London B Biol. Sci.*, vol. 237, no. 641, pp. 37-72, 1952.
- [6] R. B. Bird, W. E. Stewart, E. N. Lightfoot, *Transport Phenomena*, 2da ed., New York, EEUU: John Wiley & Sons, 2006.
- [7] F.M. White, I. Corfield, *Viscous Fluid Flow*, 3ra ed., New York, EEUU: McGraw-Hill, 2006.
- [8] D. Young, B.R. Munson, T. Okiishi, W.W. Huebsch, *A brief introduction to fluid mechanics*, 5ta ed., Missouri, USA: John Wiley & Sons, 2010.
- [9] R. E. Liesegang, *Chemical Reactions in Gels*, Dusseldorf, Alemania, Liesegang, 1898.
- [10] F. F. Runge, "Der Bildungstrieb der Stoffe: veranschaulicht in selbstständig gewachsenen Bildern (Fortsetzung der Musterbilder)", F. F. Runge, 1855.
- [11] M. Flicker, J. Ross, “Mechanism of Chemical Instability for Periodic Precipitation Phenomena”. *The Journal of Chemical Physics*, vol. 60, no. 9, pp. 3458–3465, 1974.
- [12] I. Lagzi, “Controlling and Engineering Precipitation Patterns”, *Langmuir*, vol. 28, no. 7, pp. 3350-3354, 2012.
- [13] T. Antal, M. Droz, J. Magnin, Z. Racz, M. Zrinyi, “Derivation of the Matalon-Packter law for Liesegang patterns”, *J. Chem. Phys.*, vol. 109, no. 21, pp. 9479-9486, 1998.
- [14] H. G. Weller, G. Tabor, H. Jasak, C. Fureby, “A tensorial approach to computational continuum mechanics using object-oriented techniques”, *Comput. Phys.*, vol. 12, no. 6, pp. 620-631, 1998.
- [15] P. Horgue, C. Soulaire, J. Franc, R. Guibert, G. Debenest, “An open-source toolbox for multiphase flow in porous media”, *Comp. Phys. Commun.*, vol. 187, pp. 217-226, 2015.
- [16] P. Horgue, J. Franc, G. Gerlero et al. *porousMultiphaseFoam*. [En línea]. Disponible en: <https://github.com/phorgue/porousMultiphaseFoam>, accedido 2020.
- [17] National Technology & Engineering Solutions of Sandia, LLC (NTESS), Kitware Inc. *ParaView*. [En línea]. Disponible en: <https://www.paraview.org/>, accedido 2020.
- [18] B. Stroustrup, *The C++ Programming Language*, 4ta ed., Massachusetts, EEUU: Addison-Wesley, 2013.
- [19] International Organization for Standardization. *ISO: Home page*. [En línea]. Disponible en: <https://www.iso.org/>, accedido 2020.
- [20] C. Geuzaine, J.-F. Remacle. *Gmsh*. [En línea]. Disponible en: <https://gmsh.info/>, accedido 2020.
- [21] *Git*. L. Torvalds, J. Hamano et al. [En línea]. Disponible en: <https://git-scm.com/>, accedido 2020.
- [22] GitHub Inc. *GitHub*. [En línea]. Disponible en: <https://github.com/>, accedido 2020.
- [23] GitLab Inc. *GitLab*. [En línea]. Disponible en: <https://gitlab.com/>, accedido 2020.

- [24] Google. *Google Workspace*. [En línea]. Disponible en: <https://workspace.google.com/>, accedido 2020.
- [25] Python Software Foundation. *Python*. [En línea]. Disponible en: <https://www.python.org/>, accedido 2020.
- [26] Canonical Ltd. *Ubuntu*. [En línea]. Disponible en: <https://ubuntu.com/>, accedido 2020.
- [27] Microsoft. *Windows Subsystem for Linux*. [En línea]. Disponible en: <https://docs.microsoft.com/en-us/windows/wsl/>, accedido 2020.
- [28] N. Bonnefon et al. *glogg*. [En línea]. Disponible en: <https://glogg.bonnefon.org/>, accedido 2020.
- [29] D. Ho. *Notepad++*. [En línea]. Disponible en: <https://notepad-plus-plus.org/>, accedido 2020.
- [30] Microsoft. *Visual Studio Code*. [En línea]. Disponible en: <https://code.visualstudio.com/>, accedido 2020.
- [31] B. Fox, C. Ramey et al. *Bash*. [En línea]. Disponible en: <https://www.gnu.org/software/bash/>, accedido 2020.
- [32] F. Moukalled, L. Mangani, L. Darwish, *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM® and Matlab®*, New York, EEUU: Springer, 2015.
- [33] J. Jiang, K. Sakurai, “Formation of Ultrathin Liesegang Patterns”, *Langmuir*, vol. 32, no. 36, pp. 9126–9134, 2016.
- [34] OpenFOAM Foundation. *C++ Source Code Guide*. [En línea]. Disponible en: <https://cpp.openfoam.org/v8/>, accedido 2020.
- [35] OpenFOAM Foundation. *OpenFOAM: API Guide*. [En línea]. Disponible en: <https://www.openfoam.com/documentation/guides/latest/api/index.html>, accedido 2020.
- [36] CFD Online Inc.. *CFD Online Discussion Forums*. [En línea]. Disponible en: <https://www.cfd-online.com/Forums/>, accedido 2020.
- [37] M. Lacey, *The Scrum Field Guide: Agile Advice for Your First Year and Beyond*, 2da ed., Massachusetts, EEUU: Addison-Wesley, cap 25, pp. 187, 2015.
- [38] D. Morris, *Scrum in easy steps: An ideal framework for agile projects*. London, Reino Unido: Easy Steps Ltd., 2017.
- [39] CFD Direct Limited. *OpenFOAM v8 User Guide: Mesh generation with blockMesh*. [En línea]. Disponible en: <https://cfd.direct/openfoam/user-guide/v8-blockMesh/>, accedido 2020.
- [40] CFD Direct Limited. *OpenFOAM v8 User Guide: Time and data input/output control*. [En línea]. Disponible en: <https://cfd.direct/openfoam/user-guide/v8-controldict/>, accedido 2020.
- [41] CFD Direct Limited. *OpenFOAM v8 User Guide: Numerical schemes*. [En línea]. Disponible en: <https://cfd.direct/openfoam/user-guide/v8-fvschemes/>, accedido 2020.
- [42] CFD Direct Limited. *OpenFOAM v8 User Guide: Solution and algorithm control*. [En línea]. Disponible en: <https://cfd.direct/openfoam/user-guide/v8-fvsolution/>, accedido 2020.
- [43] CFD Direct Limited. *OpenFOAM v8 User Guide: Transport/rheology models*. [En línea]. Disponible en: <https://cfd.direct/openfoam/user-guide/v8-transport-rheology/>, accedido 2020.
- [44] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, “Elementary Graph Algorithms” in *Introduction to Algorithms*, 2da ed., Massachusetts, EEUU: MIT Press, 2001.
- [45] O. C. Zienkiewicz, K. Morgan, *Finite elements and approximation*, Massachusetts, United States: Courier Corporation, 2006.
- [46] F. P. Incropera, D. P. DeWitt, “Introduction to Convection” in *Fundamentals of Heat and Mass Transfer*, 7th ed. New York, EEUU: Wiley, 2011.

[47] J. Donea, A. Huerta, *Finite element methods for flow problems*, Chichester, Inglaterra: John Wiley & Sons, 2003.

[48] D. You, R. Mittal, M. Wang, P. Moin, "Analysis of stability and accuracy of finite-difference schemes on a skewed mesh", *J. Comput. Phys.*, vol. 213, no. 1, pp. 184-204, 2006.

Anexos

Anexo A - Repositorio con código fuente

El código fuente de la aplicación desarrollada en este proyecto, junto con casos de ejemplo pueden ser encontrados en el siguiente repositorio de GitHub:

<https://github.com/octaviogarcia/porousMultiphaseFoam-reactionsLiesegang>

Para compilar solamente la librería desarrollada (modelo de reacciones de patrones de Liesegang) junto con el solucionador modificado se debe correr el script bash Partialwmake (se aconseja ejecutar Partialwclean primero si hubo alguna corrida previa del Partialwmake) como sigue.

El código fuente del modelo de reacciones se encuentra en el directorio `/libraries/reactionModels/liesegangReaction`. Por otro lado, el código fuente del solucionador modificado se encuentra en `/solvers/porousScalarTransportFoam`.

Anexo B - Plan de gestión de riesgos

En la **Tabla A-1**, se pueden observar todos los riesgos identificados para este proyecto (con un código de referencia), así como también la categoría y subcategoría a la que pertenecen¹⁸, la probabilidad asignada¹⁹, el impacto estimado²⁰, el factor de exposición de riesgo calculado²¹ y el nivel de exposición de riesgo²² al que se corresponde.

Ref.	Riesgo	Categoría	Prob.	Impacto	ER	Nivel de Riesgo
R1	Interrupción de algún servicio utilizado	Externo > Subcontratistas y Proveedores	Baja	Medio	12,5	Aceptable
R2	Cambio en la forma de evaluación y seguimiento por parte de la cátedra	Externo > Regulatorios	Muy baja	Alto	7,5	Aceptable
R3	Imposibilidad de concretar reuniones por el clima	Externo > Condiciones Climáticas	Baja	Bajo	6,25	Aceptable
R4	Superar plazos planificados por incumplimiento de las cuotas horarias semanales	Organizacional y Planeación > Autocontrol	Muy Alta	Medio	45	Muy riesgoso
R5	Pérdida de tiempo y aumento de complejidad por sobre-ingeniería	Organizacional y Planeación > Autocontrol	Baja	Alto	18,75	Poco riesgoso
R6	Pérdida de tiempo por falta de coordinación en actividades comunes	Organizacional y Planeación > Estructura Funcional y Organizacional	Media	Medio	25	Riesgoso
R7	Seguimiento inefectivo por parte del director del proyecto	Organizacional y Planeación > Estructura Funcional y Organizacional	Muy baja	Muy alto	9	Aceptable
R8	Falta de conocimiento teórico sobre la problemática para desarrollar la aplicación	Recursos Humanos > Conocimiento, Experiencia y Destrezas	Muy Alta	Alto	67,5	Muy riesgoso

¹⁸ Según taxonomía de Otero Polo.

¹⁹ Donde Muy baja = 0,1, Baja = 0,25, Media = 0,5, Alta = 0,75, Muy Alta = 0,90

²⁰ Donde Muy bajo = 10, Bajo = 25, Medio = 0,5, Alto = 75, Muy Alto = 90

²¹ Calculado como $ER = Probabilidad \text{ Riesgo} \times Impacto \text{ Riesgo}$

²² Siendo "Aceptable" cuando $ER \leq 12,5$, "Poco riesgoso" si $12,5 < ER \leq 18,75$, "Riesgoso" si $18,75 < ER \leq 25$ y "Muy riesgoso" si $ER > 25$.

R9	Disminución de la calidad del producto generado por desconocimiento de la herramienta OpenFOAM	Recursos Humanos > Conocimiento, Experiencia y Destrezas	Alta	Alto	56,25	Muy riesgoso
R10	Impacto de obligaciones laborales y académicas del alumno investigador	Recursos Humanos > Disponibilidad del Personal	Media	Medio	25	Riesgoso
R11	Falla en equipamiento utilizado para el proyecto	Recursos Físicos y Financieros > Instrumentos y Equipos	Muy baja	Medio	5	Aceptable
R12	Problemas de ejecución del software utilizado para realizar el proyecto	Procesos y Sistemas > Tecnología	Baja	Alto	18,75	Poco riesgoso
R13	Retrasos por necesidad de reconfigurar entornos de trabajo en las PCs de los participantes del proyecto	Procesos y Sistemas > Tecnología	Alta	Bajo	18,75	Poco riesgoso
R14	Las validaciones al final de cada <i>sprint</i> demoran más de lo esperado	Dirección de Proyecto > Estimación	Media	Medio	25	Riesgoso
R15	Las actividades de investigación se extienden por dificultad de comprensión del material	Dirección de Proyecto > Estimación	Media	Alto	37,5	Muy riesgoso
R16	El desarrollo fue subestimado	Dirección de Proyecto > Estimación	Media	Alto	37,5	Muy riesgoso
R17	El plan de actividades realizado no refleja las actividades que se requiere realizar.	Dirección de Proyecto > Planificación	Baja	Alto	18,75	Poco riesgoso
R18	Cambios en las tareas en una iteración	Técnicos > Requisitos	Alto	Bajo	18,75	Poco riesgoso
R19	Deterioro de la calidad del código debido a la falta de comprensión del problema impactando negativamente en la mantenibilidad	Técnicos > Complejidad	Media	Medio	25	Riesgoso
R20	Algoritmos desarrollados no se aplican a todos los casos posibles	Técnicos > Fiabilidad	Alta	Alto	56,75	Muy riesgoso
R21	Arquitectura definida para el software resulta inadecuada	Técnicos > Diseño	Baja	Bajo	6,25	Aceptable

R22	Imposibilidad de obtener artículos científicos y libros en la etapa de investigación	Legales > Permisos	Medio	Alto	37,5	Muy riesgoso
-----	--	--------------------	-------	------	------	--------------

Tabla A-1. Riesgos identificados y sus características.

Los riesgos que fueron catalogados con un nivel de “Riesgoso” o “Muy riesgoso” debían contar con una estrategia a llevar a cabo si se materializaban. Las acciones posibles son las de evitar, reducir, o asumir el riesgo. Esta información puede encontrarse en la **Tabla A-2**. La tabla fue organizada de forma que los riesgos se encuentran ordenados de mayor a menor exposición de riesgo (según valores calculados para la **Tabla A-1**). Adicionalmente, se indica para cada caso la respuesta a tomar con mayor detalle.

Para los riesgos cuya acción asignada es la de “Evitar” o “Reducir”, se especifica un disparador para conocer cuando dicho riesgo está próximo a materializarse o ya ocurrió.

Ref.	Riesgo	Acción	Respuesta
R8	Falta de conocimiento teórico sobre la problemática para desarrollar la aplicación	Asumir	Se crearon etapas de investigación de manera iterativa considerando este riesgo.
R20	Algoritmos desarrollados no se aplican a todos los casos posibles	Reducir	Se pasará por múltiples etapas de validaciones considerando este riesgo. Se realizará una batería de pruebas para reducir la probabilidad.
R9	Disminución de la calidad del producto generado por desconocimiento de la herramienta OpenFOAM	Reducir	La documentación de OpenFOAM es extensa y detallada por lo que sirve de material de consulta. El director propuesto cuenta con experiencia en la temática, además se dispone de contactos con conocimientos avanzados en OpenFOAM en CIMEC.
R4	Superar plazos planificados por incumplimiento de las cuotas horarias semanales	Evitar	Aumentar la dedicación al proyecto en los restantes días de la semana.
R15	Las actividades de investigación se extienden por dificultad de comprensión del material	Reducir	Ya se obtuvo y estudió material inicial relacionado al proyecto de investigación. Se considera que el acceso a material nuevo se puede realizar de manera progresiva para facilitar la comprensión. Además, el director del proyecto tiene experiencia en la temática a tratar.
R16	El desarrollo fue subestimado	Asumir	Es posible que las estimaciones iniciales de tiempo de desarrollo se vean afectadas por desconocimiento de la temática o falta de información.
R22	Imposibilidad de obtener artículos científicos y libros en la etapa de investigación	Reducir	Se cuenta con la experiencia del director propuesto y con contactos involucrados en el tema.
R6	Pérdida de tiempo por falta de coordinación en actividades comunes	Asumir	Se considera que es parcialmente inherente al desarrollo en equipo.

R10	Impacto de obligaciones laborales y académicas del alumno investigado	Asumir	La carga horaria semanal ya fue diseñada para que se tenga en cuenta estos imprevistos personales.
R14	Las validaciones al final de cada <i>sprint</i> demoran más de lo esperado	Asumir	Se determinaron tiempos de validación que consideran este riesgo, por lo que, si se materializa, será de un impacto menor.
R19	Deterioro de la calidad del código debido a la falta de comprensión del problema impactando negativamente en la mantenibilidad	Asumir	Las características del problema plantean un escenario donde un enfoque inicial no permite un avance claro en la estructuración del código.

Tabla A-2. Tratamiento de riesgos clasificados como “Riesgoso” o “Muy riesgoso”.

Anexo C - Uso de aplicativo Python

Para la utilización del aplicativo IgLaws, que post-procesaba los casos de simulación ejecutados, es necesario copiar el archivo correspondiente (desde el directorio /utilities/IgLaws/ del repositorio mencionado en el anexo A) al directorio /bin de la instalación de ParaView. Existen 2 archivos distintos dependiendo de si el caso de simulación es 1D o 2D.

Luego, se debe agregar la ubicación del archivo IgLaws (carpeta bin) al PATH del sistema operativo.

Finalmente, se deberá situar en el directorio raíz de un caso de simulación ejecutado (ver capítulo VI para más información) y ejecutar el código python como sigue:

IgLaws-1D.py

o

IgLaws-2D.py

según corresponda.

Para el procesado de los resultados de la simulación y la generación de la información, el aplicativo realiza la siguiente secuencia de pasos generales:

- 1) Inicialización de la interfaz ParaView y gráfica del caso.

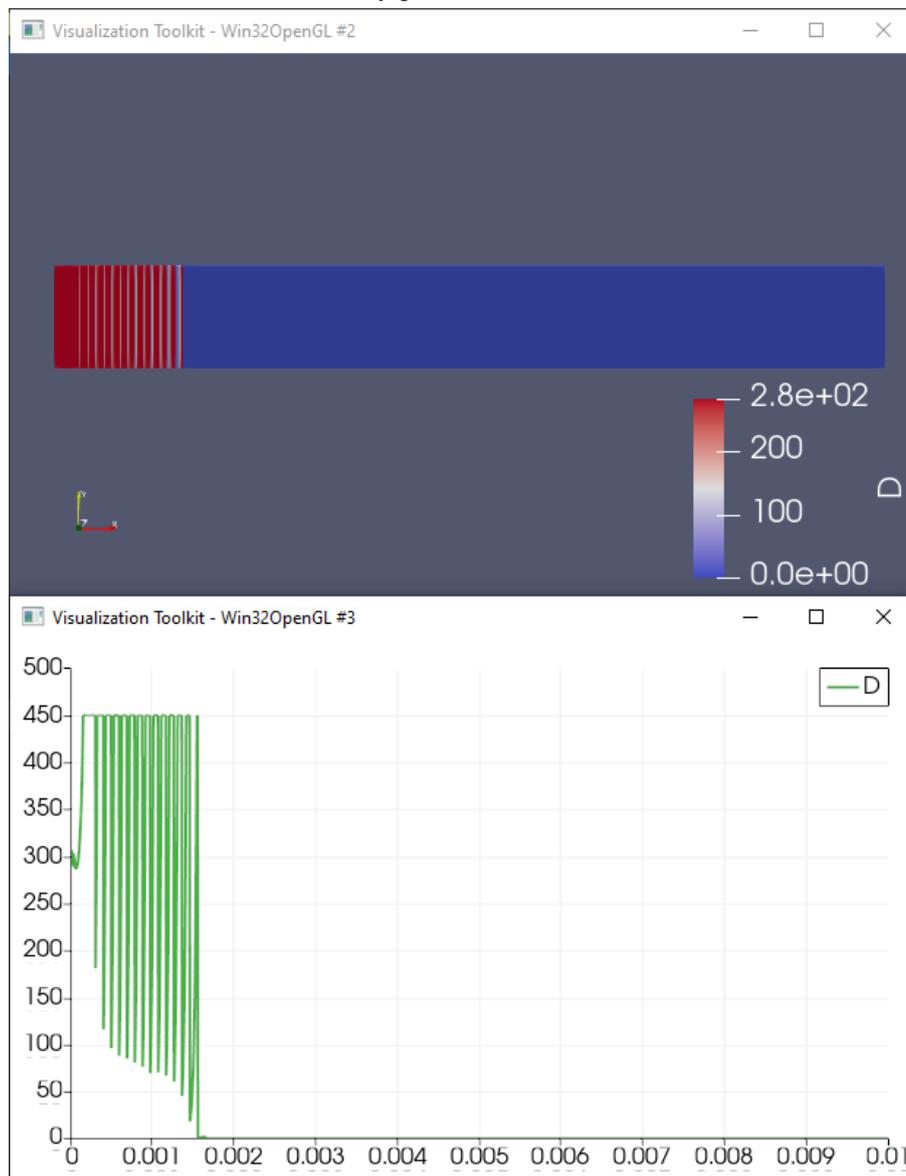


Fig C-1. Visualización 3D y gráfico de concentración sobre una línea


```

0.000139>lgLaws10mm.py
VisRTX 0.1.6, using devices:
 0: GeForce RTX 2060 (Total: 6.4 GB, Available: 5.3 GB)
Calculated rho 450.07
  Begin,      End,      Width,      Width law,      Space law
0.000166,    0.000273,    0.000107,    nan,            nan
0.000359,    0.000410,    0.000051,    0.476636,      1.501832
0.000501,    0.000547,    0.000046,    0.901961,      1.334146
0.000641,    0.000683,    0.000042,    0.913043,      1.248629
0.000780,    0.000818,    0.000038,    0.904762,      1.197657
0.000918,    0.000957,    0.000039,    1.026316,      1.169927
0.001057,    0.001094,    0.000037,    0.948718,      1.143156
0.001194,    0.001231,    0.000037,    1.000000,      1.125229
0.001333,    0.001370,    0.000037,    1.000000,      1.112916
0.001472,    0.001507,    0.000035,    0.945946,      1.100000
0.001610,    0.001646,    0.000036,    1.028571,      1.092236
0.001749,    0.001784,    0.000035,    0.972222,      1.083840
0.001887,    0.001922,    0.000035,    1.000000,      1.077354
0.002025,    0.002060,    0.000035,    1.000000,      1.071800
0.002164,    0.002198,    0.000034,    0.971429,      1.066990
0.002302,    0.002335,    0.000033,    0.970588,      1.062329
0.002438,    0.002472,    0.000034,    1.030303,      1.058672
0.002577,    0.002610,    0.000033,    0.970588,      1.055825
0.002716,    0.002748,    0.000032,    0.969697,      1.052874
0.002855,    0.002885,    0.000030,    0.937500,      1.049854
0.003007,    0.003020,    0.000013,    0.433333,      1.046794
  Min,      Min,      Min,      0.433333,      1.046794
  Max,      Max,      Max,      1.030303,      1.501832
  Average,  Average,  Average,  0.920081,      1.132603
Fin primer banda 0.000273
Fin ultima banda 0.00302
Numero de bandas: 21

```

Fig C-5. Salida de una ejecución sobre un caso.

Anexo D - Efectos de baja precisión en la formación de bandas

Como se mencionó en el capítulo VII, cuando una simulación es configurada con un dominio que carece de suficientes celdas (lo que varía según la forma, el tamaño y orientación de estas), los resultados obtenidos se ven seriamente afectados (para el fenómeno simulado, es visible principalmente como distorsiones en las bandas). Este comportamiento conocido como error de discretización es normal y esperable en las simulaciones de este tipo.

En las **Fig. D-1, D-2 y D-3** puede verse cómo las distorsiones en las bandas responden a medida que se va incrementando la cantidad de celdas.

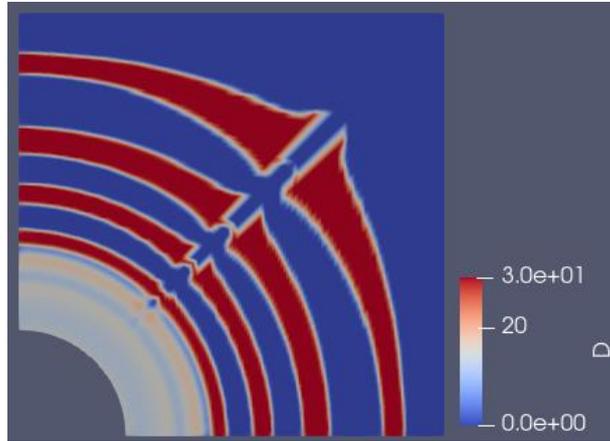


Fig. D-1. Resultados de la simulación para un dominio con 10 mil celdas.

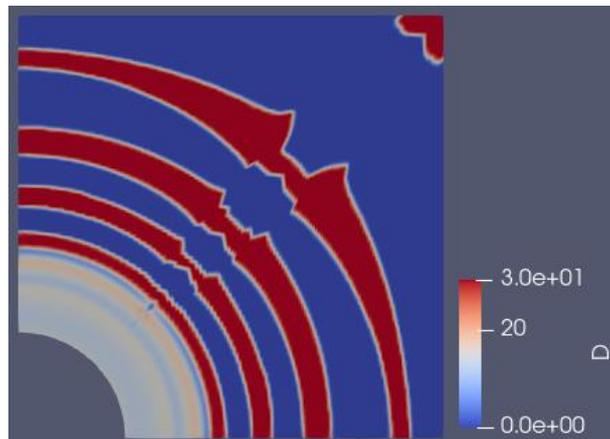


Fig. D-2. Resultados de la simulación para un dominio con 40 mil celdas.

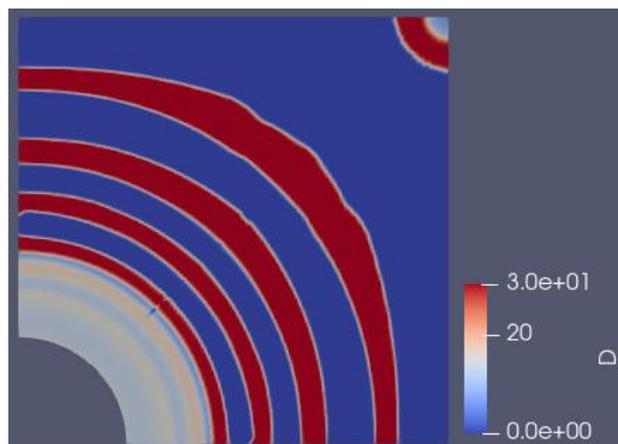


Fig. D-3. Resultados de la simulación para un dominio con 80 mil celdas.

Anexo E - Archivos de un caso

A continuación, se mostrará un ejemplo para cada uno de los archivos existentes en la carpeta de un caso de simulación (antes de su ejecución). El caso corresponde a un problema bidimensional discretizado empleando una malla circular.

Para cada archivo, se hicieron comentarios sobre cada elemento dentro. Los elementos que aparecen más de una vez dentro del archivo (una por reacción, por ejemplo) fueron comentados solo la primera vez.

Los valores numéricos de los parámetros, en algunos casos, fueron especificados en notación científica usando notación E (por ejemplo, $1e-2 = 1 \times 10^{-2} = 0.01$). En los casos donde los datos deben ser especificados con unidades (por ejemplo, en concentraciones), la convención utilizada por OpenFOAM tiene el siguiente formato:

nombre [A B C D E F G] valor

En lugar de las letras A a G, debe ir la potencia de la unidad correspondiente (se utilizan las unidades del sistema internacional de unidades). La asignación de unidades a cada lugar es como sigue:

- A: masa (kg)
- B: longitud (metros)
- C: temperatura (kelvin)
- E: moles (mol)
- F: corriente (ampère)
- G: intensidad luminosa(candela)

Por ejemplo:

cs [0 -3 0 0 1 0 0] 6

Indicaría 6 moles/m³.

Archivo controlDict:

```

FoamFile
{
  version      2;
  format       ascii;
  class        dictionary;
  location     "system";
  object       controlDict;
}
application   porousScalarTransportFoam;    //nombre aplicación
startFrom     startTime;                    //parámetro a usar para iniciar
startTime     0;                            //tiempo inicio
stopAt        endTime;                     //parámetro a usar para detener
endTime       450000;                       //tiempo fin
deltaT        1e-10;                        //deltaT inicial
writeControl   adjustableRunTime;          //no se modificó
writeInterval  10000;                       //intervalo de escritura estado/salidas
purgeWrite    0;                            //no se modificó
writeFormat    ascii;                       //formato de salida
writePrecision 6;                            //no se modificó
writeCompression off;                       //no se modificó
timeFormat     general;                     //no se modificó
timePrecision  6;                            //precisión al manejar cantidades de tiempo
adjustTimeStep yes;                         //activa o desactiva ajuste de
                                                //tiempo (algoritmo desarrollado)
runTimeModifiable true;                    //no se modificó

```

```

maxDeltaT 20; //máximo deltaT posible
CSVoutput false; //activa o desactiva escribir salidas
// también a archivo CSV
eventTimeTracking true; //no se modificó
variationMax 0.05; //variación máxima admitida de la especie D
kValueTauD 50000; //constante k para fórmula de ajuste de tiempo

```

Archivo blockMeshDict:

Importante:

A continuación, se mostrará un ejemplo del contenido del archivo blockMeshDict. Sin embargo, es importante tener en cuenta que el contenido de dicho archivo genera una estructura que sigue el diagrama que se observa en la **Fig. D-1**.

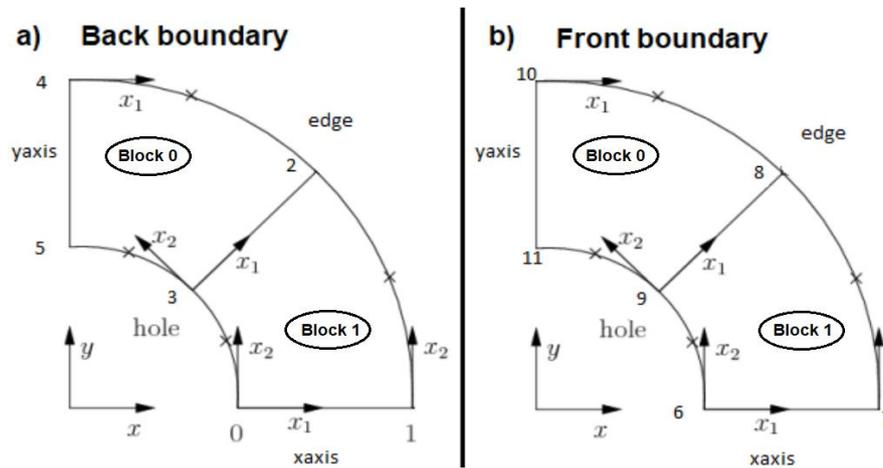


Fig. D-1. Vértices, bloques de celdas, bordes de dominio y fronteras del dominio generados con blockMesh.

Como se puede observar, el dominio generado (cuando se juntan back boundary y front boundary), en realidad es tridimensional. En la simulación, esta tercera dimensión vertical no tiene importancia ya que existe solo una celda en la dirección perpendicular a back boundary (o front boundary).

```

FoamFile
{
  version 2.0;
  format ascii;
  class dictionary;
  location "system";
  object blockMeshDict;
}
convertToMeters 1e-2; //conversión de unidades para coordenadas
vertices //definición de vértices con coordenadas cartesianas
(
  (0.5 0 0) //0
  (2 0 0) //1
  (1.414213 1.414213 0) //2
  (0.353553 0.353553 0) //3
  (0 2 0) //4
  (0 0.5 0) //5
  (0.5 0 0.01) //6
  (2 0 0.01) //7
  (1.414213 1.414213 0.01) //8

```

```

(0.353553 0.353553 0.01) //9
(0 2 0.01) //10
(0 0.5 0.01) //11
);
blocks //definición de bloques de celdas
(
  hex (3 2 4 5 9 8 10 11) (200 20 1) simpleGrading (1 1 1)
//formato: hex (<vertices_forman_bloque>) (<num_celdas_en_ejes_xyz_DEL_BLOQUE>)
// <tipo_graduacion_celdas> (graduacion_en_XYZ_del_bloque)
  hex (0 1 2 3 6 7 8 9) (200 20 1) simpleGrading (1 1 1)
//Esta malla cuenta con un total de 200x20x2 = 8000 celdas, como se puede ver
);
edges //definición de bordes de dominio como arcos
(
  arc 0 3 (0.469846 0.17101 0) //formato: arc vertice_inicial vertice_final (<coord_punto_medio_arco>)
  arc 3 5 (0.17101 0.469846 0)
  arc 1 2 (1.847759 0.76537 0)
  arc 2 4 (0.765367 1.847759 0)
  arc 6 9 (0.469846 0.17101 0.01)
  arc 9 11 (0.17101 0.469846 0.01)
  arc 7 8 (1.847759 0.76537 0.01)
  arc 8 10 (0.765367 1.847759 0.01)
);
boundary //definición de límites de dominio
(
  yaxis //nombre borde dominio
  { type empty; //tipo borde
    faces //caras que componen el borde
    (
      (4 5 11 10) //vértices que componen esta cara
    );
  }
  xaxis
  { type empty;
    faces
    ((0 1 7 6));
  }
  hole
  { type patch;
    faces
    ((5 3 9 11)
      (3 0 6 9)
    );
  }
  edge
  {
    type patch;
    faces
    ((1 2 7 8)
      (2 4 8 10)
    );
  }
  front
  {
    type empty;
    faces

```

```

( (11 9 8 10)
  (9 6 7 8)
  );
}
back
{
  type empty;
  faces
  ( (5 4 2 3)
    (3 2 1 0)
    );
}
);
mergePatchPairs          //no se modificó
();

```

Archivo fvSchemes:

```

FoamFile
{
  version 2.0;
  format  ascii;
  class   dictionary;
  location "system";
  object  fvSchemes;
}
ddtSchemes          //esquema usado para cálculo de derivadas de tiempo
{ default CrankNicolson 1;
}
gradSchemes         //esquema usado para cálculo de gradiente
{ default Gauss linear;
}
divSchemes          //esquema usado para cálculo de divergencia
{ default none;
  div(phi,C) Gauss upwind;
}
laplacianSchemes   //esquema usado para cálculo de laplaciano
{ default none;
  laplacian(Deff,C) Gauss linear corrected;
}
interpolationSchemes //esquema usado para cálculo de interpolaciones
{ default linear;
}
snGradSchemes     //esquemas usado para cálculo de gradiente de superficie normal
{ default corrected; }

```

Archivo fvSolution:

```

FoamFile
{
  version 2.0;
  format  ascii;
  class   dictionary;
  location "system";
  object  fvSolution;
}

```

```

solvers {
  C {
    solver    PBiCG;    //método para resolver sistema lineal de ecuaciones
    preconditioner DILU; //método para factorizar matrices para resolver
    tolerance  1e-9;    //tolerancia de residuo en resultados de ecuaciones
    relTol     0;      //tolerancia relativa
  }
}
reactions {
  tolerance  1e-9;    //tolerancia de residuo para ecuaciones de reacción
  massConservative false; //activa o desactiva algoritmo conservación de masa
}

```

Archivo A:

```

FoamFile
{
  version  2.0;
  format   ascii;
  class    volScalarField; //tipo de campo escalar
  location "0";           //carpeta en la que se ubica el archivo
  object   A;
}
dimensions [0 -3 0 0 1 0 0]; //dimensiones de la concentración de la especie
internalField uniform 0;     //valor en centros de celdas, definido como campo
boundaryField
{
  hole //especifica valor en borde "hole" en blockmeshdict
  { type fixedValue; //tipo de valor especificado
    value uniform 500; //valor de la concentración
  }
  edge
  { type zeroGradient; }
  front
  { type empty; }
  back
  { type empty; }
  yaxis
  { type empty; }
  xaxis
  { type empty; }
}

```

Archivo B:

```

FoamFile
{
  version  2.0;
  format   ascii;
  class    volScalarField;
  location "0";
  object   B;
}
dimensions [0 -3 0 0 1 0 0];
internalField uniform 10;
boundaryField

```

```

{
  hole
  { type      zeroGradient;}
  edge
  { type      zeroGradient;}
  front
  { type      empty; }
  back
  { type      empty; }
  yaxis
  { type      empty; }
  xaxis
  { type      empty; }
}

```

Archivo C:

```

FoamFile
{
  version  2.0;
  format   ascii;
  class    volScalarField;
  location "0";
  object   C;
}
dimensions [0 -3 0 0 1 0 0];
internalField uniform 0;
boundaryField
{
  hole
  { type      zeroGradient; }
  edge
  { type      zeroGradient; }
  front
  { type      empty; }
  back
  { type      empty; }
  yaxis
  { type      empty; }
  xaxis
  { type      empty; }
}

```

Archivo D:

```

FoamFile
{
  version  2.0;
  format   ascii;
  class    volScalarField;
  location "0";
  object   D;
}
dimensions [0 -3 0 0 1 0 0];
internalField uniform 0;
boundaryField

```

```

{
  hole
  { type      zeroGradient; }
  edge
  { type      zeroGradient; }
  front
  { type      empty; }
  back
  { type      empty; }
  yaxis
  { type      empty; }
  xaxis
  { type      empty; }
}

```

Archivo Utheta:

```

FoamFile
{
  version  2.0;
  format   ascii;
  class    volVectorField;
  location "0";
  object   Utheta;
}
dimensions [0 1 -1 0 0 0 0];
internalField uniform (0.0 0.0 0.0);
boundaryField
{
  hole
  { type      zeroGradient; }
  edge
  { type      zeroGradient; }
  front
  { type      empty; }
  back
  { type      empty; }
  yaxis
  { type      empty; }
  xaxis
  { type      empty; }
}

```

Archivo reactions:

```

FoamFile
{
  version  2.0;
  format   ascii;
  class    dictionary;
  location "constant";
  object   reactions;
}
reactionModel liesegangReaction; //selección del modelo de reacciones desarrollado
reactions

```

```

{
  ABReaction
  {
    reaction "A^1 + B^1 = C^0"; //ecuación de reacción A+B->C
    kf kf [0 3 -1 0 -1 0 0] 0.001; //cte de velocidad química para la reacción
  }
  CReaction
  {
    reaction "C^1 = D^0"; //ecuación de reacción C->D
    kf kf [0 0 -1 0 0 0 0] 0.001; //cte de velocidad para la coagulación (k1)
    cs cs [0 -3 0 0 1 0 0] 6; //punto de floculación
    liesegang liesegang [0 0 0 0 0 0 0] 1; //indica reacción patrones de Liesegang
    steepness steepness [0 0 0 0 0 0 0] 4096; //pendiente de sigmoidea
  }
  CDReaction
  {
    reaction "C^1 = D^1"; //2da ecuacion de reaccion C->D (efecto autocatalítico)
    kf kf [0 3 -1 0 -1 0 0] 0.001; //cte de velocidad para formación precipitado autocatalítico (k2)
    liesegang liesegang [0 0 0 0 0 0 0] 2; //indica reacción patrones de Liesegang
    rho rho [0 -3 0 0 1 0 0] 30; //concentración máx. especie D en bandas
    cradius cradius [0 1 0 0 0 0 0] 0.00015; //radio de cristalización
  }
}

```

Archivo transportProperties:

```

FoamFile
{
  version 2.0;
  format ascii;
  class dictionary;
  location "constant";
  object transportProperties;
}
eps eps [0 0 0 0 0 0] 1; //propiedad del medio poroso
porousTransport
{ phaseName theta; } //nombre de fase
species //especies involucradas en la simulación
(
  A
  B
  C
  D
);
A
{ Dm Dm [0 2 -1 0 0 0 0] 1e-9; //difusividad especie A
  porousTransport //otras propiedades transporte especie A
  {
    Kd Kd [-1 3 0 0 0 0 0] 0;
    rs rs [1 -3 0 0 0 0 0] 0;
    epsTotal epsTotal [0 0 0 0 0 0 0] 1;
    lambda lambda [0 0 -1 0 0 0 0] 0;
  }
  dispersionModel alphaDispersion; //modelo de dispersión
  alphaDispersionCoeffs //info de coeffs de dispersión alfa
  {

```

```

tau tau [0 0 0 0 0 0] 1;
alphaL alphaL [0 1 0 0 0 0] 0;
alphaT alphaT [0 1 0 0 0 0] 0;
}
}
B
{ Dm Dm [0 2 -1 0 0 0] 1e-9;
porousTransport
{
Kd Kd [-1 3 0 0 0 0] 0;
rs rs [1 -3 0 0 0 0] 0;
epsTotal epsTotal [0 0 0 0 0 0] 1;
lambda lambda [0 0 -1 0 0 0] 0;
}
dispersionModel alphaDispersion;
alphaDispersionCoeffs
{
tau tau [0 0 0 0 0 0] 1;
alphaL alphaL [0 1 0 0 0 0] 0;
alphaT alphaT [0 1 0 0 0 0] 0;
}
}
C
{ Dm Dm [0 2 -1 0 0 0] 1e-9;
porousTransport
{
Kd Kd [-1 3 0 0 0 0] 0;
rs rs [1 -3 0 0 0 0] 0;
epsTotal epsTotal [0 0 0 0 0 0] 1;
lambda lambda [0 0 -1 0 0 0] 0;
}
dispersionModel alphaDispersion;
alphaDispersionCoeffs
{
tau tau [0 0 0 0 0 0] 1;
alphaL alphaL [0 1 0 0 0 0] 0;
alphaT alphaT [0 1 0 0 0 0] 0;
}
}
D
{ Dm Dm [0 2 -1 0 0 0] 0;
porousTransport
{
Kd Kd [-1 3 0 0 0 0] 0;
rs rs [1 -3 0 0 0 0] 0;
epsTotal epsTotal [0 0 0 0 0 0] 1;
lambda lambda [0 0 -1 0 0 0] 0;
}
dispersionModel alphaDispersion;
alphaDispersionCoeffs
{
tau tau [0 0 0 0 0 0] 1;
alphaL alphaL [0 1 0 0 0 0] 0;
alphaT alphaT [0 1 0 0 0 0] 0;
}
}
}

```

Archivo run:

```
#!/bin/sh
. $WM_PROJECT_DIR/bin/tools/RunFunctions
runApplication blockMesh
runApplication porousScalarTransportFoam
```

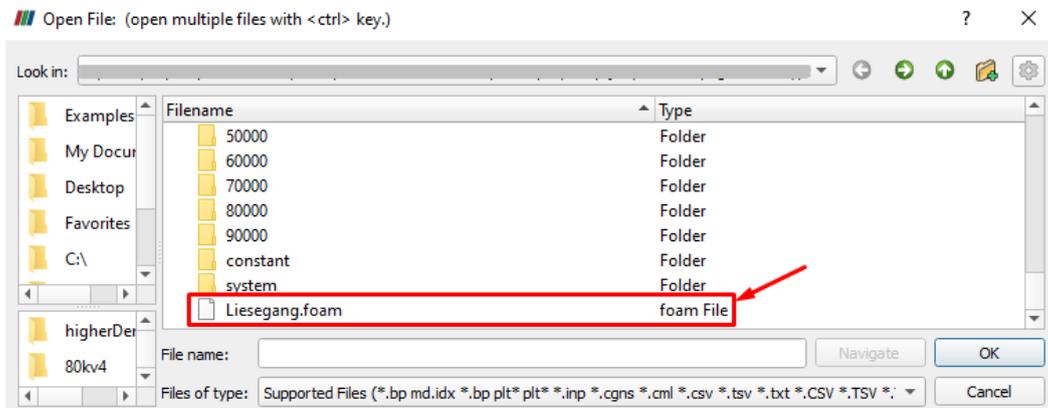
Archivo clean:

```
#!/bin/sh
. $WM_PROJECT_DIR/bin/tools/RunFunctions
rm -f log.* *.csv
rm -rf 1* 2* 3* 4* 5* 6* 7* 8* 9* constant/polyMesh
```

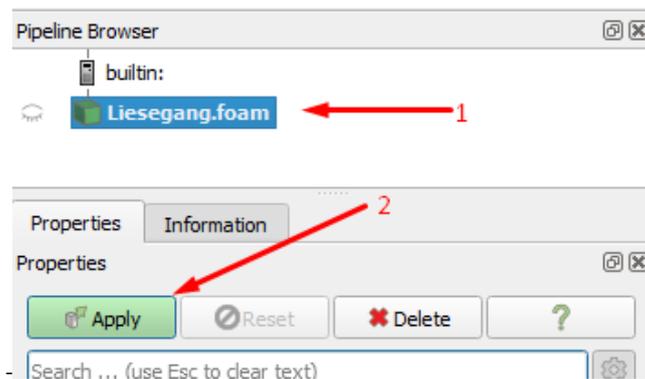
Anexo F - Instrucciones visualización de casos en ParaView

Para poder cargar un caso de simulación en ParaView, y poder visualizar los resultados obtenidos, se deben seguir los siguientes pasos:

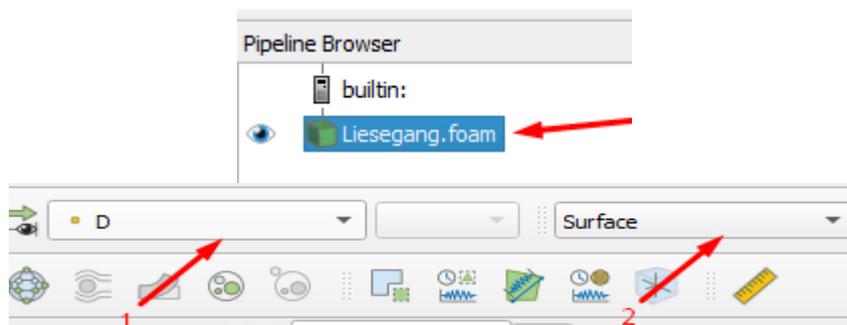
- Abrir ParaView.
- Localizar el caso y abrir el archivo .foam dentro de la carpeta del mismo:



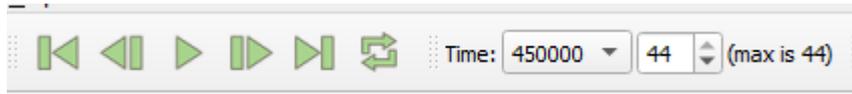
- Seleccionar el caso desde “Pipeline Browser” y hacer click en “Apply” para generar la visualización:



- Una vez generada la visualización, es posible observar la distribución de cada especie en el dominio. Para lograr eso, se selecciona el caso, luego se debe seleccionar la especie de interés y seleccionar la opción de visualizar superficie:

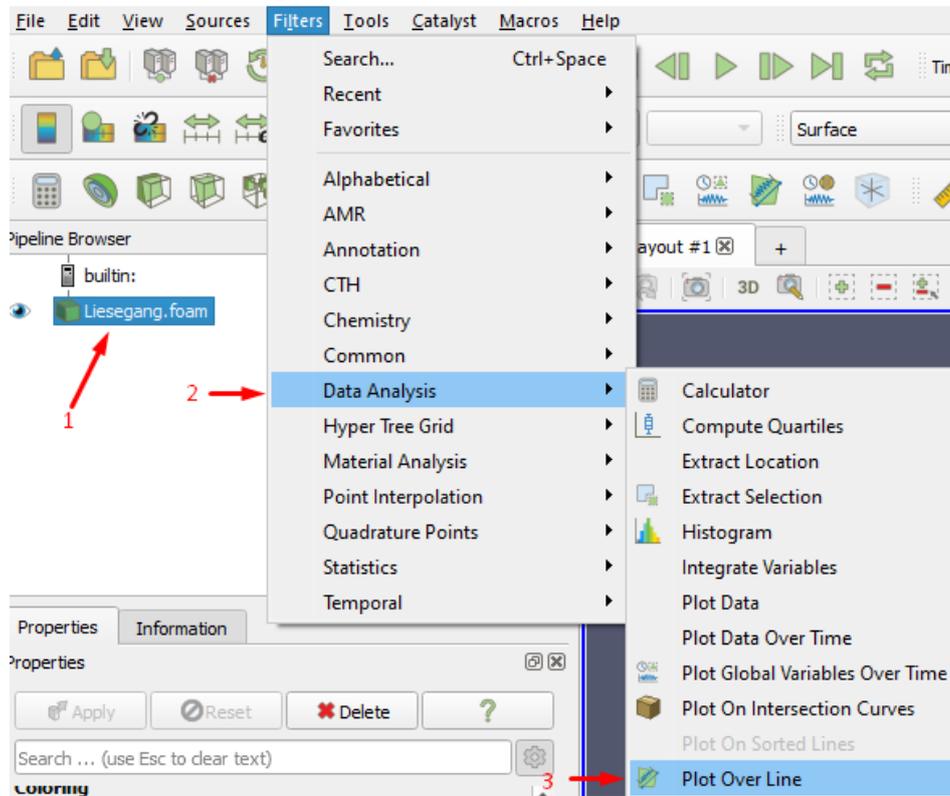


El control del tiempo en ParaView se puede manejar usando los botones de la parte superior de la ventana:



Para visualizar los gráficos de concentraciones a lo largo del dominio, mostrados en el capítulo VII, realizar los pasos siguientes:

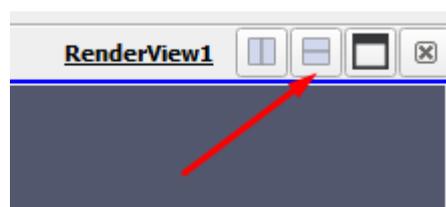
- Generar un gráfico de análisis como se muestra a continuación:



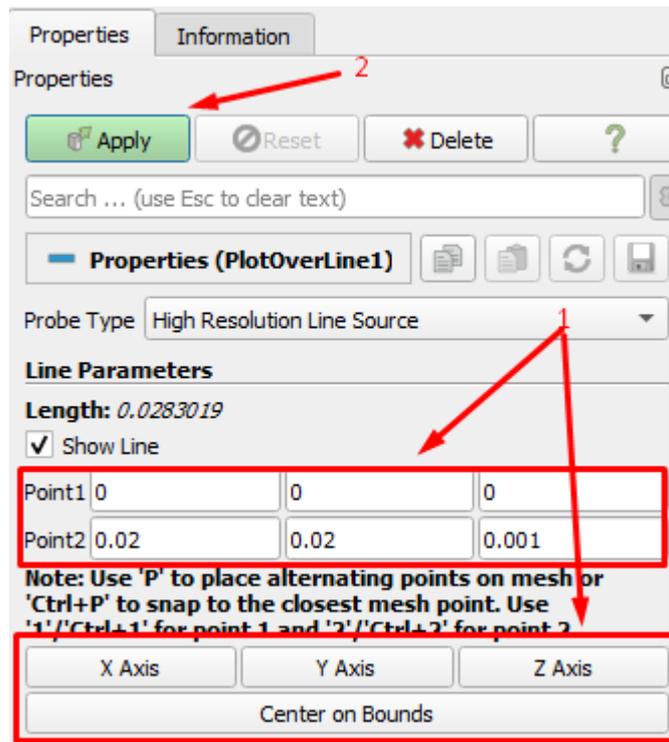
- Seleccionar el caso desde



- Generar otra vista en ParaView haciendo click sobre el siguiente botón (ubicado en la esquina superior derecha de la vista ya existente):



- Luego, en el panel de propiedades del gráfico (PlotOverLine1), se debe especificar los puntos para trazar una línea a lo largo de la cual se mide la concentración (para casos 1D simplemente usar el botón X Axis y para 2D usar Center on Bounds) y hacer click en "Apply" como se muestra a continuación:



- Esto genera gráficos para todas las especies de la simulación a lo largo del dominio. Para ver las especies individualmente, se debe seleccionar lo deseado en el panel de propiedades del gráfico (PlotOverLine1), como se muestra a continuación:

