



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
POSGRADO EN CIENCIAS DE LA TIERRA  
INSTITUTO DE GEOFÍSICA  
EXPLORACIÓN, AGUAS SUBTERRÁNEAS, MODELACIÓN Y  
PERCEPCIÓN REMOTA

MODELO DE TRANSPORTE EN MEDIOS  
POROSOS A ESCALA DE LABORATORIO  
PARA LA SIMULACIÓN DEL PROCESO DE  
TAPONAMIENTO/DESTAPONAMIENTO POR  
MICROORGANISMOS

T E S I S

QUE PARA OPTAR POR EL GRADO DE:  
**MAESTRO EN CIENCIAS DE LA TIERRA**  
(MODELACIÓN)

PRESENTA:

**Mario Noyola Rodríguez**

DIRECTOR DE TESIS:

Dr. Martín Alberto Díaz Viera  
Instituto Mexicano del Petróleo, IMP

COMITE TUTOR:

Dr. Eric Morales Casique  
Instituto de Geología  
Dr. Arturo Ortiz Tapia  
Instituto Mexicano del Petróleo

Ciudad Universitaria, Cd. México, Mayo 2017



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



# Resumen

En esta tesis se presenta un modelo de transporte en medios porosos, que fue derivado usando una metodología sistemática para la modelación matemática, numérica y computacional. El cual fue aplicado para la simulación de los procesos de taponamiento/destaponamiento en experimentos de recuperación mejorada de hidrocarburos vía microbiana (MEOR) a condiciones de laboratorio.

La recuperación mejorada de hidrocarburos vía microbiana (MEOR) es una técnica de recuperación mejorada en donde se utilizan microorganismos y sus productos para la recuperación adicional de hidrocarburos en un yacimiento. Uno de los procesos fundamentales que afectan la recuperación consiste en el taponamiento/destaponamiento del medio poroso debido a la biomasa de los microorganismos.

Desde el punto de vista metodológico, se describe cada etapa del desarrollo del modelo que consisten en los modelos conceptual, matemático, numérico y computacional.

Se obtuvo un modelo matemático general de transporte en medios porosos que incluye fenómenos físico-químicos relevantes tales como advección, difusión, dispersión, adsorción-desorción y reacciones aplicando la formulación axiomática de la modelación de medios continuos.

El sistema de ecuaciones diferenciales obtenido fue discretizado mediante el método de elemento finito en el espacio y de diferencias finitas hacia atrás en el tiempo, resultando en un esquema numérico completamente implícito.

La implementación computacional se realizó en el lenguaje de programación Python 2.7 utilizando el proyecto FEniCS 2016.1 que es un software de código abierto y multiplataforma (Windows/Linux).

El modelo de transporte resultante fue numéricamente validado para dos casos de estudio. Uno de inyección de un pulso de un trazador inerte y el otro en un proceso de taponamiento/destaponamiento por microorganismos. Los resultados se comparan con los obtenidos usando el software comercial COMSOL *Multiphysics*<sup>®</sup>.



# Abstract

In this thesis a transport model in porous media was derived using a systematic methodology approach for mathematical, numerical and computational modeling is presented. This model was applied for the simulation of the clogging/declogging processes in microbial enhanced oil recovery (MEOR) experiments at laboratory conditions.

Microbial enhanced oil recovery (MEOR) is an enhanced recovery technique where microorganisms and their products are used for additional oil recovery in a reservoir. One of the fundamental processes that affect recovery consists of the clogging/declogging of the porous medium due to the microorganisms biomass.

From the methodological point of view, each stage of the development of the model that consists of the conceptual, mathematical, numerical and computational models is described.

A general mathematical model of transport in porous media was obtained that includes relevant physical-chemical phenomena such as advection, diffusion, dispersion, adsorption-desorption and reactions applying the axiomatic formulation of continuum media modeling.

The obtained system of differential equations was discretized by the finite element method in space and backwards finite differences in time, resulting in a fully implicit numerical scheme.

The computational implementation was performed in the Python 2.7 programming language using the FEniCS 2016.1 project which is an open source and multiplatform software (Windows/Linux).

The resulting transport model was numerically validated for two case studies. One is the injection of an inert tracer pulse and the other one is a process of microorganisms clogging/declogging. The results are compared with the ones obtained using a commercial software COMSOL *Multiphysics*<sup>®</sup>.



*"Declaro conocer el Código de Ética de la Universidad Nacional Autónoma de México, plasmado en la Legislación Universitaria. Con base en las definiciones de integridad y honestidad ahí especificadas, aseguro mediante mi firma al calce que el presente trabajo es original y enteramente de mi autoría. Todas las citas de, o referencias a, la obra de otros autores aparecen debida y adecuadamente señaladas, así como acreditadas mediante los recursos editoriales convencionales".*

Mario Noyola Rodriguez, Ciudad Universitaria, Cd. México Mayo 2017.





# Índice general

<b>Resumen</b>	<b>I</b>
<b>Abstract</b>	<b>III</b>
<b>Índice general</b>	<b>X</b>
<b>Lista de figuras</b>	<b>XII</b>
<b>Lista de tablas</b>	<b>XIII</b>
<b>Introducción</b>	<b>XV</b>
0.1. Estado del arte de los modelos en medios porosos . . . . .	<b>XVI</b>
<b>1. Metodología de la Modelación Matemática, Numérica y Computacional</b>	<b>1</b>
<b>2. Introducción al Método de Elementos Finitos</b>	<b>5</b>
2.1. Reseña histórica . . . . .	<b>5</b>
2.2. La definición de elementos finitos . . . . .	<b>5</b>
2.3. Notación . . . . .	<b>6</b>
2.4. Espacios $H^1$ de elementos finitos . . . . .	<b>6</b>
2.5. El elemento de Lagrange . . . . .	<b>7</b>
2.6. La construcción de elementos finitos de referencias generales . . . . .	<b>8</b>
2.7. Cambio de base . . . . .	<b>9</b>
2.8. Espacios de polinomios . . . . .	<b>9</b>
2.9. El mapeo del elemento de referencia . . . . .	<b>10</b>
2.10. Mapeo de local a global de los grados de libertad . . . . .	<b>11</b>
2.11. Formulación variacional de Elementos Finitos . . . . .	<b>12</b>
2.12. Espacios de elementos finitos . . . . .	<b>13</b>
2.12.1. Triángulos . . . . .	<b>13</b>

<b>3. El Proyecto FEniCS</b>	<b>17</b>
3.1. Interfaz Python . . . . .	17
3.2. Componentes que forman el proyecto FEniCS. . . . .	17
3.3. Ilustración de un ejemplo del proyecto FEniCS 2016.1 . . . . .	22
<b>4. Modelo Matemático</b>	<b>29</b>
4.1. Hipótesis generales del modelo . . . . .	29
4.2. Derivación del modelo matemático . . . . .	30
4.3. Balance de masa de una componente $i$ en una fase fluida ( $f$ ) . . . . .	30
4.4. Balance de masa de una componente $i$ en la fase sólida ( $s$ ) . . . . .	31
4.5. Balance total de masa de la componente $i$ . . . . .	31
<b>5. Modelo Numérico</b>	<b>33</b>
5.1. Planteamiento del problema general . . . . .	34
5.2. Modelo numérico ecuación transporte de una componente en un medio poroso en 2D	34
<b>6. Modelo Computacional</b>	<b>37</b>
6.1. Bibliotecas utilizadas . . . . .	37
6.2. Geometría . . . . .	37
6.2.1. Dominio . . . . .	37
6.2.2. Mallado . . . . .	38
6.2.3. Espacios de funciones: . . . . .	39
6.2.4. Fronteras . . . . .	39
6.2.5. Condiciones iniciales . . . . .	41
6.2.6. La formulación variacional . . . . .	41
6.2.7. Gráficas de la solución . . . . .	41
<b>7. Validación del Modelo</b>	<b>43</b>
7.1. Problema de inyección tipo escalón . . . . .	43
7.2. Implementación computacional . . . . .	45
7.3. Número de Péclet . . . . .	49
7.3.1. Análisis numérico de la difusión usando el Número de Péclet . . . . .	49
<b>8. Caso de Estudio I</b>	<b>51</b>
8.1. Inyección de un pulso de trazador inerte. . . . .	51
8.2. Supuestos del caso de estudio . . . . .	51
8.3. Modelo matemático . . . . .	52
8.4. Implementación computacional . . . . .	52
8.4.1. Geometría de la malla . . . . .	52
8.4.2. Condiciones de fronteras . . . . .	53
8.4.3. Discusión de resultados . . . . .	55

<b>9. Caso de Estudio II</b>	<b>57</b>
9.1. Proceso de taponamiento/destaponamiento por microorganismos . . . . .	57
9.2. Supuestos generales del modelo: . . . . .	57
9.3. Modelo matemático . . . . .	59
9.4. Modelo numérico . . . . .	62
9.5. Implementación computacional . . . . .	62
9.5.1. Geometría de la malla . . . . .	62
9.5.2. Condiciones de fronteras . . . . .	63
9.6. Simulaciones numéricas . . . . .	63
9.6.1. Análisis físico del caso de estudio . . . . .	72
9.7. Discusión de resultados . . . . .	73
<b>10. Conclusiones y Trabajo Futuro</b>	<b>75</b>
<b>A. Glosario de definiciones relevantes</b>	<b>77</b>
A.0.1. Núcleos de Berea . . . . .	77
A.0.2. Inóculo microbiano . . . . .	78
A.0.2.1. Porosidad . . . . .	78
A.0.2.2. Permeabilidad . . . . .	78
A.0.3. Propiedades del fluido . . . . .	78
A.0.3.1. Fase . . . . .	78
A.0.3.2. Componente . . . . .	78
A.0.3.3. Tipos de fluidos de yacimiento . . . . .	79
A.0.3.4. Métodos de recuperación de aceite . . . . .	79
A.0.3.5. Modelos de procesos de simulación . . . . .	79
A.0.4. Propiedades de roca/fluido . . . . .	79
A.0.4.1. Saturación de fluido . . . . .	79
A.0.5. Trazador inerte . . . . .	79
A.0.6. Biomasa . . . . .	80
A.0.7. Nutrientes . . . . .	80
A.0.8. Biomasa y nutrientes . . . . .	80
A.0.9. Decaimiento . . . . .	80
A.0.10. Crecimiento . . . . .	80
A.0.11. Taponamiento/Destaponamiento . . . . .	81
<b>B. Formulación Axiomática de la Modelación de Medios Continuos</b>	<b>83</b>
B.1. Propiedades intensivas . . . . .	84
B.2. Propiedades extensivas . . . . .	85
B.3. Balance de propiedades extensivas e intensivas . . . . .	86
B.4. Ecuación de balance global . . . . .	86
B.5. Condiciones de balance local . . . . .	86

B.6. Ecuaciones de balance local . . . . .	87
B.6.1. Ejemplos de condiciones de balance local . . . . .	87
B.7. Condiciones iniciales y de frontera . . . . .	88
<b>C. Elementos de análisis funcional</b>	<b>91</b>
C.1. Teoría de integración de espacios de Lebesgue . . . . .	91
C.1.1. Espacios de Sobolev . . . . .	92
C.2. Espacios de Hilbert . . . . .	93
C.3. El Teorema de Lax-Milgram . . . . .	93
C.4. Teorema: (Lax-Milgram) . . . . .	93
C.5. Teorema de la representación de Riesz . . . . .	94
<b>D. Generalización del Tensor de dispersión Hidrodinámica</b>	<b>95</b>
<b>E. Esquema de Diferencias Finitas en el Tiempo</b>	<b>99</b>
E.0.1. Esquema Theta ( $\theta$ ) general . . . . .	99
E.0.2. Consideremos tres casos para Theta . . . . .	100
E.0.3. Ecuación parabólica semi-descretizada empleando el esquema de diferencias finitas hacia atrás . . . . .	100
<b>F. Código que se utilizó para la Ecuación de Transporte</b>	<b>101</b>
<b>Bibliografía</b>	<b>111</b>

# Índice de figuras

1.1.	Diagrama de la Metodología para generar un modelo. . . . .	3
2.1.	Elementos de Lagrange de uno y dos órdenes. . . . .	11
2.2.	Los grados de libertad para elementos $P_1(K)$ . . . . .	14
3.1.	DOLFIN funciona como la interfaz de usuario principal de FEniCS y se encarga de la comunicación entre los diversos componentes de FEniCS y software externo. . .	18
3.2.	La forma compilador FFC genera código C++ en formato UFC de una forma variacional de elementos finitos dada en formato UFL. . . . .	20
6.1.	Ejemplo del dominio de definición del modelo computacional en 2D para nuestra ecuación, donde $l = 0.08$ m y $d = 0.038$ m. . . . .	38
6.2.	Condiciones de frontera . . . . .	39
7.1.	Gráfica de la solución analítica de la concentración contra el tiempo en el problema de inyección tipo escalón a la salida. . . . .	44
7.2.	(a) Ejemplo del dominio de definición del modelo computacional en 2D para nuestra ecuación, donde $l = 0.08$ m y $d = 0.038$ m. (b) Mallado en 2D con triángulos, la malla se compone con 440 vértices y 720 celdas ordenadas. . . . .	45
7.3.	Condiciones de frontera . . . . .	46
7.4.	Gráfica de la concentración contra el tiempo del problema inyección tipo escalón realizado con el código en el Proyecto FEniCS 2016.1: . . . . .	47
7.5.	Gráfica de las comparaciones a las salidas de las concentraciones, la línea roja representa los datos en proyecto FEniCS 2016.1, la línea de estrellas los datos en COMSOL <i>Multiphysics</i> ®. . . . .	48
7.6.	Gráficas con diferentes números de Péclet . . . . .	50
8.1.	La discretización de la malla para la solución numérica, está formada por 5031 vértices 9796 elementos triangulares. Peclet Local máximo = 0.0080, hmax = 0.0027	53

8.2.	Concentración de inyección de entrada en el experimento. . . . .	54
8.3.	Ajuste de la solución numérica con el experimento. Con un error máximo = $3.8729e-05$ . . . . .	54
8.4.	Distribución espacio-temporal del ión $Cl^-$ a lo largo del núcleo. . . . .	55
9.1.	La discretización de la La malla para la solución numérica, está formada por 1712 vértices y 3224 elementos triangulares. Peclet Local máximo = 3.4816, hmax = 0.0039	63
9.2.	Perfiles de microorganismos planctónicos ( $c_w^m$ ) a lo largo del eje vertical en varias secuencias de una hora, en la plataforma proyecto FEniCS 2016.1 (a) y el software COMSOL <i>Multiphysics</i> <sup>®</sup> (b). . . . .	65
9.3.	Perfiles de nutrientes ( $c_w^n$ ) a lo largo del eje vertical en varias secuencias de una hora, en la plataforma proyecto FEniCS 2016.1 (a) y el software COMSOL <i>Multiphysics</i> <sup>®</sup> (b). . . . .	67
9.4.	Perfiles de microorganismos sésiles ( $\sigma$ ) a lo largo del eje vertical en múltiples secuencias de una hora, en la plataforma proyecto FEniCS 2016.1 (a) y el software COMSOL <i>Multiphysics</i> <sup>®</sup> (b). . . . .	69
9.5.	Perfiles de la porosidad a lo largo del eje vertical en varias secuencias de una hora.	71
9.6.	Perfiles de la permeabilidad a lo largo del eje vertical en varias secuencias de una hora. . . . .	71
9.7.	Comportamiento en las primeras 4 horas continuas de las concentraciones de los microorganismos, nutrientes y porosidad a lo largo del eje vertical en secuencias de una hora. . . . .	72
9.8.	Comportamiento de las concentraciones las 24 horas de los microorganismos, nutrientes y porosidad a lo largo del eje vertical en secuencias de una hora. . . . .	73
A.1.	Diferencia entre absorción y adsorción. . . . .	77
A.2.	Núcleo de Berea . . . . .	78
B.1.	Movimiento de partículas. . . . .	84
B.2.	Medio poroso. . . . .	88

# Índice de tablas

2.1. El elemento finito que se analizará, incluyendo el nombre completo y el respectivo (orden más alto). . . . .	6
8.1. Parámetros para el problema de inyección NaCl de un trazador a través de un núcleo de arenisca Berea. . . . .	51
9.1. Propiedades intensivas y extensivas asociadas con la masa de los componentes por fases. . . . .	59
9.2. Parámetros para el problema de inyección continua y simultánea de microorganismos y nutrientes a través de un núcleo de arenisca Berea. . . . .	64





# Introducción

Los yacimientos petroleros en la fase inicial de operación producen básicamente usando su energía natural, esta etapa se conoce como *recuperación primaria*. A medida que el yacimiento pierde energía, requiere de la inyección de gas o de agua para restablecer o mantener la presión del yacimiento, esta etapa se llama *recuperación secundaria*. Cuando los métodos de recuperación secundarios resultan ineficaces, es necesario aplicar otros métodos más sofisticados como son la inyección de vapor, de productos químicos, microorganismos, etc. Esta etapa se conoce como *recuperación mejorada o terciaria* (EOR, por sus siglas en inglés).

La recuperación mejorada de hidrocarburos vía microbiana (MEOR, por sus siglas en inglés) es una técnica de recuperación mejorada de hidrocarburos en donde se utilizan microorganismos y sus productos para la recuperación adicional de hidrocarburos en un yacimiento [Bryant y Lockhart \(2002\)](#). En principio, MEOR es un proceso que incrementa la recuperación de aceite a través de la inoculación de microorganismos en un yacimiento, con el objetivo de que las bacterias causen algunos efectos beneficiosos como, la movilización de aceite residual y el desvío de los fluidos de inyección a través de las áreas de barrido del yacimiento al obstruir las zonas de alta permeabilidad. Las tecnologías microbianas son cada vez más aceptadas en todo el mundo como métodos rentables y respetuosos con el medio ambiente para mejorar la producción de petróleo [Sarkar et al. \(1989\)](#).

Las tecnologías de recuperación mejorada de hidrocarburos consisten principalmente de procesos fisicoquímicos en su mayoría, sin embargo la aplicación de procesos biológicos es una alternativa técnica y económicamente factible. El proceso MEOR involucra el uso de microorganismos nativos del yacimiento o especialmente seleccionados (crecimiento a altas temperaturas, salinidad, presión y pH) para producir metabolitos específicos que impactan en una mejoría de la recuperación [Bryant y Lockhart \(2002\)](#).

La modelación del comportamiento de los microorganismos que influyen en la recuperación mejorada de hidrocarburos vía microbiana y de sus actividades en el yacimiento, ha atraído fuertemente el interés desde el inicio de las investigaciones sobre MEOR.

## 0.1. Estado del arte de los modelos en medios porosos

Corapcioglu Yavuz y Haridas (1985) propusieron un modelo matemático para predecir la distribución espacial y temporal de microorganismos (bacterias y virus) como una herramienta para la evaluación a largo plazo del riesgo de acumulación de contaminantes microbianos en suelos y aguas subterráneas. La ecuación gobernante para el transporte de microorganismos se acopla a una ecuación de transporte de nutrientes presentes en el agua residual; los mecanismos de taponamiento y destaponamiento son incorporados dentro del modelo. Cuando el decaimiento es considerado como una reacción de primer orden y el crecimiento bacteriano sigue la ecuación de Monod (1949), las ecuaciones del modelo no son lineales y están acopladas. Un sistema de ecuaciones simplificadas se resuelve analíticamente para verificar los resultados numéricos; se obtienen soluciones numéricas en una y dos dimensiones mediante el método de Galerkin.

Knapp (1988) desarrollaron un modelo matemático unidimensional para describir el proceso de taponado de los microorganismos. El modelo fue usado para investigar el impacto del crecimiento celular y la retención sobre la reducción espacial y temporal en la permeabilidad del medio poroso. Se supuso que el desarrollo de la fase estacionaria se debe solamente a la retención de biomasa y que el transporte convectivo es el mecanismo dominante de transporte de bacterias. Las ecuaciones gobernantes incluyen una ecuación de convección-dispersión para el transporte de microorganismos y nutrientes y una ecuación de conservación de masa para el desarrollo de la fase estacionaria.

Islam y Gianetto (1993) derivaron un modelo matemático que describe el transporte bacteriano, la propagación de nutrientes y el crecimiento microbiano en medios porosos. Ellos usaron una técnica de sobre relajación sucesiva para resolver las ecuaciones diferenciales parciales gobernantes.

Sarkar *et al.* (1991) desarrollaron un simulador numérico unidimensional de múltiples componentes en dos fases para modelar el transporte y el crecimiento de bacterias y la recuperación de aceite en un proceso MEOR. Las ecuaciones básicas que gobiernan el transporte de aceite, agua, bacterias, nutrientes y metabolitos en medios porosos son las ecuaciones de conservación de masa de los componentes. La reducción de la permeabilidad fue modelada usando la teoría de medio efectivo. El modelo de recuperación de aceite está basado en mecanismos tales como, reducción de la tensión interfacial por bio-surfactantes y el taponado selectivo por la biomasa. En su modelo un algoritmo implícito en presión explícito en concentración fue usado para resolver las ecuaciones de la presión y las de conservación de masa.

Chang *et al.* (1991) y Chang *et al.* (1992) incorporaron las ecuaciones que gobiernan el transporte de microorganismos y nutrientes en un modelo de petróleo negro trifásico en 3D. Las actividades microbianas simuladas incluyeron: el flujo neto de microorganismos por convección y dispersión, el crecimiento, el decaimiento microbianos, el consumo de nutrientes y la deposición de microorganismos sobre la superficie de la roca. La alteración en la mojabilidad de la roca durante el

tratamiento microbiano fue considerada como el mecanismo para la recuperación de aceite. Basado en resultados experimentales, las correlaciones empíricas entre las concentraciones de las células y la mojabilidad de la roca, y entre la mojabilidad de la roca y las saturaciones de la fase residual fueron establecidas.

Bryant *et al.* (1992) realizaron un conjunto de experimentos de laboratorio para obtener datos reales para alimentar un simulador de transporte microbiano. Los parámetros microbianos que determinaron fueron:

- Crecimiento y decaimiento microbiano,
- Deposición microbiana,
- Difusión microbiana,
- Dispersión convectiva,
- Consumo de nutrientes.

Zhang *et al.* (1995) desarrollaron un modelo matemático de flujo de múltiples componentes trifásico tridimensional en medios porosos que incluye reacciones físicas, químicas y biológicas. El modelo propuesto incluye: ecuaciones de transporte por componentes, las ecuaciones del modelo de petróleo negro, las ecuaciones de la cinética microbiana, un modelo de reducción de la permeabilidad, y modelos de procesos recuperación de hidrocarburos. Los autores desarrollaron un simulador numérico tridimensional trifásico de múltiples componentes para investigar el transporte y el crecimiento de microorganismos en medios porosos y el impacto de las actividades microbianas en la recuperación de aceite. Las actividades microbianas modeladas en este estudio incluyen:

- El crecimiento, retención, y productos finales de inhibición del crecimiento,
- La formación de productos metabólicos,
- El consumo de nutrientes.

Desai *et al.* (1999) propusieron un modelo estequiométrico para un proceso de fermentación con *Clostridium acetobutylicum* este modelo incorpora ecuaciones no lineales. Desarrollaron un programa de cómputo que usa un enfoque de optimización global heurística independiente del modelo para resolver el problema no lineal resultante.

Thullner *et al.* (2004) desarrollaron un modelo bidimensional para comparar una simulación de taponamiento debido a microorganismos, con una célula de flujo experimental. Ellos investigaron los cambios en el campo de flujo, lo que muy probablemente podría atribuirse a biotaponamiento.

Kim (2006) desarrolló un modelo de transporte para investigar los cambios en la porosidad y permeabilidad debido a la actividad microbiana. En este trabajo se realizó también un análisis de sensibilidad sobre la concentración microbiana de efluentes, a parámetros relativos a la fijación y desprendimiento de los mismos microorganismos.

Nielsen *et al.* (2010) desarrolló un modelo 1D, donde los microorganismos transforman los nutrientes en surfactante, el efecto de los últimos fue investigado mediante la interpolación de parámetros de curvas de permeabilidad relativa de tipo Corey para obtener un mejor ajuste con los datos experimentales. También investigaron el efecto de las concentraciones inyectadas de sustrato (nutrientes) y bacterias.

Li *et al.* (2011) investigaron la relación funcional entre la saturación de aceite residual y el número de capilaridad, a través de la solución de un modelo de elementos finitos biológicos e hidrogeológicos acoplados. Realizaron varios estudios paramétricos en 1D, incluyendo la tasa de crecimiento bacteriano, las tasas de adherencia y desprendimiento, las evoluciones de porosidad y permeabilidad, y su modelo de prueba, de la evolución de la concentración bacteriana relativa (con respecto a la concentración original) en el tiempo.

Huang y Monteagudo (2012) desarrollaron un modelo matemático que explica los procesos de Generación Microbiana de Gas (MGG) y Recuperación Mejorada de Petróleo Microbiano (MEOR), involucrando un modelo cinético que predice el crecimiento celular y la producción de metabolitos de gas y biopolímeros. El modelo también considera la reducción de la saturación de aceite residual y el cambio de viscosidad del agua por el biopolímero. Un modelo de adsorción representa la retención de solutos en la fase acuosa, alterando así la porosidad y la permeabilidad. El modelo se implementó en un simulador de reservorio de composición y de aceite negro en 3D de campo completo, utilizando el enfoque de diferencias finitas, y utilizando algunas hipótesis para tener en cuenta la producción de biomasa y bio-metabolitos en las ecuaciones discretizadas.

En el trabajo de Díaz-Viera *et al.* (2015), se derivó un modelo de transporte en medios porosos aplicando un enfoque sistemático de modelación continua, implementado utilizando el método de elementos finitos para simulación, analizar e interpretar numéricamente los procesos de recuperación mejorada de aceite vía microbiana (MEOR) bajo condiciones controladas de laboratorio y a escala de núcleo, tales como taponamiento/destaponamiento.

La motivación de este trabajo es desarrollar un modelo de transporte a escala de laboratorio para realizar simulaciones numéricas de los procesos de recuperación mejorada vía microbiana a escala de laboratorio. Para el diseño óptimo de métodos de recuperación mejorada de petróleo se requiere realizar una variedad de pruebas de laboratorio bajo condiciones controladas para entender cuáles son los mecanismos fundamentales de recuperación para un método de recuperación mejorada dado en un yacimiento específico.

Un modelo adecuado de las pruebas de laboratorio sería decisivo en la interpretación y comprensión de los mecanismos de recuperación y en la obtención de los parámetros pertinentes para la posterior implementación de procesos de recuperación mejorados en el pozo y a la escala del yacimiento.

Las pruebas de laboratorio suelen tener una serie de inconvenientes, entre los que se incluyen, que

son muy sofisticadas, caras y poco representativas de toda la gama de fenómenos involucrados.

El objetivo de la presente tesis es desarrollar un modelo de general transporte en medios porosos a escala de laboratorio usando el software FEniCS 2016.1. El modelo desarrollado es aplicado para la simulación del proceso de taponamiento/destaponamiento por microorganismos.

Resulta importante destacar que al desarrollar un modelo de transporte lo suficientemente general y flexible, éste puede ser aplicado a otros procesos de recuperación mejorada de manera casi directa. Sólo se requeriría considerar las particularidades (leyes constitutivas específicas) de cada uno de los términos involucrados en la ecuación del modelo.

Desde el punto de vista metodológico, se describe cada etapa del desarrollo del modelo que consisten en los modelos conceptual, matemático, numérico y computacional.

El modelo matemático general de transporte en medios porosos que incluye fenómenos físico-químicos relevantes tales como advección, difusión, dispersión, adsorción-desorción y reacciones es derivado aplicando la formulación axiomática de la modelación de medios continuos.

El sistema de ecuaciones diferenciales obtenido es discretizado mediante el método de elemento finito en el espacio y de diferencias finitas hacia atrás en el tiempo, resultando en un esquema numérico completamente implícito.

La implementación computacional es realizada en el lenguaje de programación Python 2.7 utilizando el proyecto FEniCS [Petter Langtangen y Logg \(2016\)](#). El proyecto FEniCS 2016.1 es un software alternativo desarrollado bajo el paradigma de código abierto, libre y multiplataforma (Windows/Linux) que permite trabajar en un entorno colaborativo, extensible y sostenible. A diferencia de otros modelos implementados usando software comerciales como [COMSOL Multiphysics \(2008\)](#), éste permitiría probar diversos modelos matemáticos y numéricos, además de poder adaptarse y extenderse a diversos problemas de recuperación de petróleo, y proponer recomendaciones en casos particulares.

El modelo de transporte resultante fue aplicado para dos casos de estudio en experimentos de transporte a través de núcleos. Uno de inyección de un pulso de un trazador inerte y el otro en un proceso de taponamiento/destaponamiento por microorganismos. Los resultados se comparan con los obtenidos usando el software comercial COMSOL *Multiphysics*<sup>®</sup>.

El contenido de esta tesis está compuesto por los siguientes capítulos:

En el capítulo 1 se muestra la metodología general de la modelación matemática, numérica y computacional.

En el capítulo 2 se presenta una introducción al método de elementos finitos, el cual será uti-

lizado para la discretización del modelo matemático.

En el capítulo 3 se presenta una descripción general del Proyecto FEniCS 2016.1, el cual se usa para la implementación computacional del modelo.

En el capítulo 4 se realiza la derivación del modelo matemático de transporte aplicando el enfoque axiomático para la modelación de los sistemas continuos.

En el capítulo 5 se muestra el modelo numérico desarrollado para el modelo de transporte.

En el capítulo 6 se presenta el modelo computacional implementado usando el Proyecto FEniCS 2016.1.

En el capítulo 7 se realiza la validación del modelo en un caso de estudio de inyección tipo escalón mediante la comparación con una solución analítica tomada del artículo de [Van Genuchten y Alves \(1982\)](#), y se muestra el impacto del número de Peclet para casos de advección dominante.

En los capítulos 8 y 9 se presentan dos casos de estudio: uno de inyección de un pulso trazador inerte y el otro de un proceso de taponamiento/destaponamiento por microorganismos.

En el capítulo 10 se presentan las conclusiones y trabajo futuro.

Adicionalmente en el *Apéndice A* se muestra un glosario con las definiciones de los parámetros para la simulación de los procesos MEOR. En el *Apéndice B* expone el enfoque axiomático para la modelación de los sistemas continuos, el cual se utiliza para la derivación de los modelos matemáticos anteriores. En los *Apéndices C* se expone los elementos de análisis funcional utilizados en la derivación del modelo de transporte. En el *Apéndice D* se muestra el desarrollo y generalización del tensor de dispersión hidrodinámica. En el *Apéndice E* se desarrolla la derivación matemática del método de diferencias finitas empleado en el modelo numérico.

En el *Apéndice F* se muestra y se documenta el código que se utilizó en el modelo computacional. Finalmente mostramos la bibliografía utilizada en la elaboración de este trabajo de tesis.

# Metodología de la Modelación Matemática, Numérica y Computacional

Para la elección de una solución, se requiere una herramienta que permita pronosticar la respuesta de un sistema dado a la implementación de un esquema propuesto. La herramienta que permite realizar estas predicciones es el modelo [Díaz-Viera \*et al.\* \(2008\)](#). Un modelo puede ser definido como una versión simplificada del sistema real que simula aproximadamente las respuestas de éste. La estructura del sistema y los procesos que ocurren dentro de éste son por consiguiente representados por modelos conceptuales, diseñados para cumplir los requisitos de cierto tipo de problemas bajo una escala dada. En este proceso, se construyen modelos que se utilizan para predecir el comportamiento de un sistema, que requiere formalmente de cuatro etapas:

- Modelo conceptual
- Modelo matemático
- Modelo numérico
- Modelo computacional

Durante el planteamiento del modelo conceptual se establecen todas las hipótesis, supuestos, condiciones, alcances y limitaciones que debe satisfacer el modelo. En consecuencia, se definen cuáles son las fases, componentes y propiedades sujetas a balance, así como las posibles relaciones de dependencia entre éstas.

La etapa de modelación matemática consiste en sentido general en obtener un modelo matemático que describa el comportamiento a través de un sistema de Ecuaciones Diferenciales Parciales (EDPs) del fenómeno a estudiar.

Para la generación de modelos matemáticos existen esencialmente dos enfoques: el determinístico y el estocástico. El enfoque determinístico consiste en establecer una ecuación o conjunto de ecuaciones cuya solución única describe de manera unívoca el comportamiento del fenómeno a modelar. Mientras que el enfoque estocástico persigue reproducir el comportamiento del fenómeno en el sentido estadístico, es decir, se obtienen múltiples soluciones (realizaciones) estadísticamente



equivalentes. Según sea la escala de la modelación se adopta el punto de vista microscópico cuando modelamos a escalas muy pequeñas y deseamos ver los detalles a nivel de partículas: se adopta el punto de vista macroscópico cuando el sistema se considera como un continuo y las propiedades son efectivas tomadas como promedio en cierto volumen representativo.

No obstante, los problemas con condiciones iniciales y de frontera que son generados usualmente no se pueden resolver de manera analítica por lo que hay que recurrir a su solución numérica.

La etapa de la modelación numérica consiste en obtener un modelo numérico a partir del modelo matemático desarrollado, el cual pueda ser implementado en un programa de cómputo. Los modelos numéricos son versiones discretas de los modelos matemáticos, cuyas soluciones son aproximadas hasta cierto nivel de error, pero a su vez deben ser consistentes con la solución del modelo matemático.

Los métodos numéricos más usados para la discretización en espacio y tiempo de sistemas de ecuaciones diferenciales parciales son de tres tipos:

- El método de diferencias finitas (FDM)
- El método de elementos finitos (FEM)
- El método de volumen finito (FVM)

Además de la discretización del sistema de EDPs se requiere en mayor o menor medida la aplicación de métodos numéricos, como son:

- Métodos de linealización del tipo de Newton-Raphson
- Métodos, directos e iterativos, para resolver el sistema de ecuaciones algebraicas lineales resultantes.
- Métodos Óptimos de Construcción de Mallas.

La modelación computacional es la etapa en la que se traduce el modelo numérico en uno computacional el cual debe estar implementado en una plataforma o equipo de cómputo específico y codificado en cierto lenguaje de programación.

En este trabajo se realizó la implementación computacional mediante el uso del software proyecto FEniCS 2016.1 que es una plataforma numérica general basada en el método de elementos finitos para la solución automatizada de ecuaciones diferenciales parciales con condiciones iniciales y de frontera. Para la validación del modelo se confirma que los resultados sean una buena aproximación del fenómeno en cuestión. Frecuentemente, esto se realiza mediante comparaciones entre los resultados del modelo computacional y soluciones semi-analíticas. Otra manera de realizar una validación es mediante la comparación con resultados publicados o resultados de pruebas de laboratorio.

En la figura (1.1) se observa el proceso de la formulación y los pasos necesarios para poder generar un modelo y obtener una solución numérica aproximada.

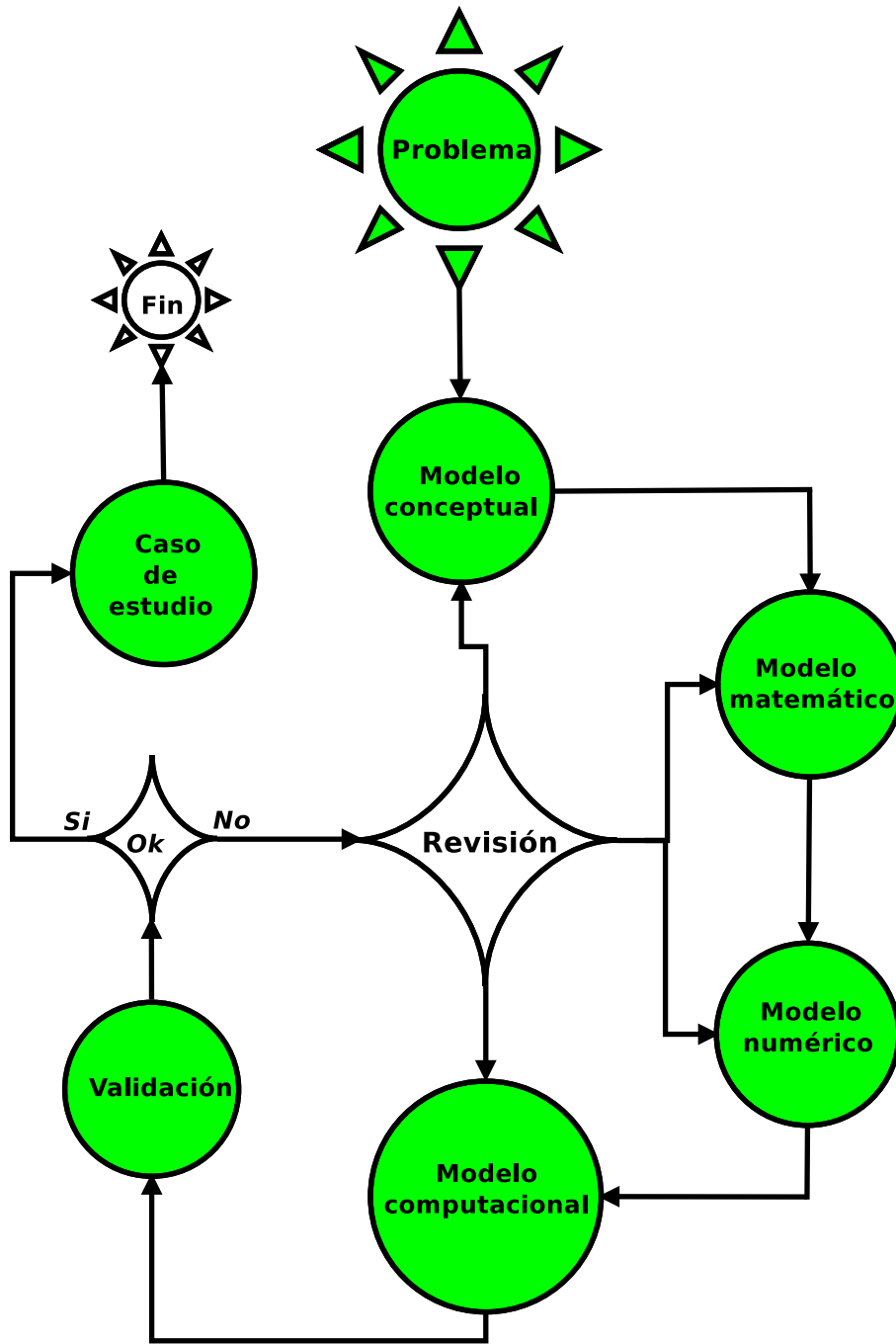


Figura 1.1: Diagrama de la Metodología para generar un modelo.

En los casos de estudio, se analizan dos situaciones para estudiar los fenómenos específicos de interés. Se proponen datos de entrada para resolver el modelo. Los resultados de los casos de estudios es la predicción del comportamiento del sistema bajo las condiciones supuestas.



# Introducción al Método de Elementos Finitos

## 2.1. Reseña histórica

Toda esta sección fue tomada de [Chen \(2005\)](#). En 1915, Boris Grigoryevich Galerkin formuló un método general para la resolución de ecuaciones diferenciales [Galerkin \(1915\)](#). Anteriormente ya se había presentado un enfoque similar por Bubnov. El método de Galerkin o método de Bubnov-Galerkin, el método se formuló originalmente con polinomios globales y se remonta a los principios variacionales de Leibniz, Euler, Lagrange, Dirichlet, Hamilton, Castigliano. El método Galerkin con espacios polinomiales a trozos  $(V_h, \tilde{V}_h)$  es conocido como el Método de los Elementos Finitos. El método de elementos finitos fue introducido por ingenieros para el análisis de estructuras mecánicas en 1950s y fue propuesto independientemente por [Courant \(1943\)](#). El uso del método de elementos finitos entre los ingenieros y matemáticos se dió en la década de 1960s desde entonces, se ha ampliado y refinado el método de los elementos finitos en un marco global para el diseño y análisis de métodos numéricos para ecuaciones diferenciales. El trabajo a posteriori de análisis de errores del método de elementos finitos se remonta a los trabajos del pionero [Babuška y Rheinboldt \(1978\)](#).

## 2.2. La definición de elementos finitos

La definición de elemento finito [Logg et al. \(2012\)](#) se introdujo por primera vez en un conjunto de notas [Ciarlet \(1975\)](#). Hasta hoy se mantiene la definición estándar [Brenner y Scott \(2008\)](#). La definición dice lo siguiente:

**Definición 1.1:** Un elemento finito [Ciarlet \(2002\)](#) se define por una tripleta  $(T, \mathcal{V}, \mathcal{L})$ , donde

- El dominio  $T$  es acotado y es subconjunto cerrado de  $\mathbb{R}^d$  (para  $d = 1, 2, 3, \dots$ ) con el límite interior no vacío y suave a tramo en la frontera;

- El espacio  $\mathcal{V} = \mathcal{V}(T)$  es un espacio funciones de dimensión finita sobre el dominio  $T$  de dimensión  $n$ ;
- El conjunto de grados de libertad (nodos)  $\mathcal{L} = \{\ell_1, \ell_2, \dots, \ell_n\}$  es una base para el espacio dual  $\hat{\mathcal{V}}$  es decir el espacio de funciones lineales acotadas en  $\mathcal{V}$ .

**Definición 1.2:** Sea  $(\hat{T}, \hat{\mathcal{V}}, \hat{\mathcal{L}})$  y  $(T, \mathcal{V}, \mathcal{L})$  elementos finitos y  $F_T : \hat{T} \rightarrow T$  sea un mapeo afín no generado.

Los elementos finitos son de equivalencia afín si  $\mathcal{F}^*(v) = \hat{\mathcal{V}}$  y  $\mathcal{F}_*(\hat{\mathcal{L}}) = \mathcal{L}$ .

Una consecuencia de la equivalencia afín es que solamente se puede construir un base nodal, y entonces se puede asignar a cada celda en una malla. Además, esta idea de equivalencia se puede extender a algunos elementos del valor del vector cuando se utilizan cierto tipo de mapeos. Los elementos de Lagrange son equivalentes afín en  $H^1$ .

## 2.3. Notación

- El espacio de los polinomios de grado  $e$  incluyendo  $q$  sobre el dominio  $T \subset \mathbb{R}^d$  es denotado por  $\mathcal{P}_q(T)$  y los correspondientes campos del vector  $d$  por  $[\mathcal{P}_q(T)]^d$ .
- Un espacio de elementos finitos  $E$  es llamado V-conforme si  $E \subseteq V$ . Si no, es llamado (V-) no conforme.
- Los elementos de  $\mathcal{L}$  se refieren generalmente a los grados de libertad de los elementos  $(T, \mathcal{V}, \mathcal{L})$ .

Al describir las familias de los elementos finitos, es habitual para ilustrar los grados de libertad con cierta notación esquemática. Como se muestra en la tabla (2.1) para el caso de esta tesis.

Elemento finito	Nombre corto	Espacio de Sobolev	Conforme ajuste
Lagrange	CG	$H^1$	si

Tabla 2.1: El elemento finito que se analizará, incluyendo el nombre completo y el respectivo (orden más alto).

## 2.4. Espacios $H^1$ de elementos finitos

El espacio  $H^1$  es fundamental en el análisis y discretización de la forma débil para los problemas elípticos de segundo orden, y los subespacios de elemento finito de  $H^1$  dan lugar a algunos de los elementos finitos más conocidos. Típicamente estos elementos utilizan espacios  $C^0$  que se aproximan desde una función suave a trozos en un dominio acotado en  $H^1$  si y sólo si es continuo Dietrich (2007). En este trabajo consideramos los elementos clásicos de Lagrange.

## 2.5. El elemento de Lagrange

El elemento finito más conocido y utilizado más ampliamente es el elemento  $\mathcal{P}_1$  de Lagrange. Este triángulo de menor grado algunas veces es llamado triángulo de Courant, después del papel influyente por Courant (1943) en el que las técnicas variacionales se utilizan con el triángulo  $\mathcal{P}_1$  para derivar método de diferencias finitas. Algunas veces este método es visto como el método de los elementos finitos, pero en realidad hay una familia de elementos parametrizados por el grado del polinomio que generaliza la interpolación univariada de polinomios simples.

Los elementos de Lagrange de grado superior ofrecen mayores propiedades de aproximación de orden. Por otra parte estos pueden ayudar al fenómeno cuando se usan elementos lineales para dar mejores propiedades de estabilidad discretos.

**Definición 1.4:** Los elementos de Lagrange ( $CG_q$ ) están definidos para  $q = 1, 2, \dots$  por

$$T \in \{\text{segmento}, \text{triangulo}, \text{tetraedros}\} \quad (2.1)$$

$$\mathcal{V} = \mathcal{P}_q(T), \quad (2.2)$$

$$\ell_i(v) = v(x^i), \quad i = 1, \dots, n(q), \quad (2.3)$$

donde  $\{x^i\}_{i=1}^{n(q)}$  es una enumeración de puntos en  $T$  definida por

$$x = \begin{cases} \frac{i}{q}, & 0 \leq i \leq q, \quad T \text{ intervalo}, \\ \left(\frac{i}{q}, \frac{j}{q}\right) & 0 \leq i + j \leq q, \quad T \text{ triangulo}, \\ \left(\frac{i}{q}, \frac{j}{q}, \frac{k}{q}\right) & 0 \leq i + j + k \leq q, \quad T \text{ tetraedro}. \end{cases}$$

La dimensión de estos elementos finitos de Lagrange corresponde a los grados completos de polinomios de grado  $q$  sobre  $T$  y es

$$n(q) = \begin{cases} q + 1, & T \text{ intervalo}, \\ \binom{q+1}{2} (q+1)(q+2), & T \text{ triangulo}, \\ \binom{q+1}{3} (q+1)(q+2)(q+3), & T \text{ tetraedro}. \end{cases}$$

La definición anterior presenta una opción para el conjunto de puntos  $\{x^i\}$ . Sin embargo, está no es la única opción posible. En general, basta con que el conjunto de puntos  $\{x^i\}$  que no tiene solución y que en los puntos de frontera son localizados afín de permitir el ensamble en  $C^0$ . El conjunto de puntos debe incluir los vértices  $q - 1$  puntos en cada borde,  $\frac{(q-1)(q-2)}{2}$  puntos para cada cara y así sucesivamente. Los puntos de las fronteras deben ser colocados de forma simétrica de modo que los puntos sobre las celdas adyacentes coincidan. Mientras que el acondicionamiento numérico y las propiedades de interpolación se puede mejorar de buena manera por la elección de estos puntos de manera inteligente.

Dejando  $\prod_q^T$  denote el interpolador definido por los grados de libertad por encima de los elementos

de Lagrange de grado  $q$  tenemos de [Brenner y Scott \(2008\)](#) que:

$$\left\| u - \Pi_q^T u \right\|_{H^1(T)} \leq Ch_T^q |u|_{H^{q+1}(T)}, \quad \left\| u - \Pi_q^T u \right\|_{L^2(T)} \leq Ch_T^{q+1} |u|_{H^{q+1}(T)} \quad (2.4)$$

donde, y en todas partes  $C$  denota una constante positiva genérica que no depende de  $h_T$ , pero depende del grado de  $q$ , y la proporción de aspecto simple, y  $u$  es una función (ó vector de campo) regular. Los valores vectoriales o tensoriales de los elementos de Lagrange generalmente se construyen mediante el uso de elementos de Lagrange para cada componente.

## 2.6. La construcción de elementos finitos de referencias generales

En esta definición el término de *elementos finitos* se refiere no sólo a una celda en particular en la malla, sino también al espacio funcional asociado y a los grados de libertad. Típicamente el dominio  $T$  es de forma poligonal o poliédrica simple, y el espacio de funciones  $\mathcal{V}$  consta de polinomios. Dado un elemento finito, una base concreta, a menudo llamada la base nodal, para este elemento puede ser calculado usando la siguiente definición.

*Definición 2.2:* La base nodal para un elemento finito  $(T, \mathcal{V}, \mathcal{L})$  es el conjunto de funciones  $\{\phi\}_{i=1}^n$ , tal que para todo  $1 \leq i, j \leq n$ ,

$$l_i(\phi_j) = \delta_{ij}, \quad (2.5)$$

donde  $\delta_{ij}$  denota la función delta de Kronecker.

El problema principal que nos ocupa es la construcción de esta base nodal. Para cualquier elemento finito dado, uno puede construir la base nodal de forma explícita con el álgebra elemental. Los grados de libertad de un elemento en particular a menudo se eligen de manera que se cumplan estos requisitos de continuidad. Además de calcular la base nodal, se ha simplificado para las siguientes tareas:

1. Evaluar las funciones de base y sus derivadas en los puntos.
2. Asociar las funciones de base (o grados de libertad) con la topología de las caras  $T$  como sus vértices, aristas y colocarlas en los bordes.
3. Asociar a cada función base los metadatos adicionales que describen la correlación que se debe utilizar para la evaluación de las funciones base y sus derivadas.
4. Establecer normas aplicables a la evaluación de los grados de libertad aplicada a funciones arbitrarias (necesario para las condiciones de frontera tipo Dirichlet).

El primero de ellos es relativamente simple en el marco del cálculo simbólico, pero requieren más cuidado si una aplicación utiliza la aritmética numérica. Los dos de en medio codifican la información necesaria para un código para transformar la base de referencia y un ensamble global de grados de libertad cuando el elemento finito es más o menos  $C^0$  continua.

## 2.7. Cambio de base

La idea fundamental en la construcción de una base nodal es de álgebra lineal elemental: se construye la base deseada (nodal) como una combinación lineal de otra base disponible. Vamos a empezar con algunos  $\{\psi\}_{i=1}^n$  que se extiende por  $\mathcal{V}$ . De esta forma, construimos cada base nodal y las funciones  $\phi_j$  como

$$\phi_j = \sum_{k=1}^n \alpha_{kj} \psi_k, \quad (2.6)$$

La tarea consiste en calcular la matriz  $\alpha$ , cada uno fija  $\phi_j$  debe satisfacer

$$\ell_i(\phi_j) = \delta_{ij} \quad (2.7)$$

y el uso de la expansión anterior para  $\phi_j$ , obtenemos

$$\delta_{ij} = \sum_{k=1}^n \ell_i(\alpha_{jk} \psi_k) = \sum_{k=1}^n \alpha_{jk} \ell_i(\psi_k) \quad (2.8)$$

Por lo tanto, para un  $j$  fijo tenemos un sistema de  $n$  ecuaciones

$$\sum_{k=1}^n B_{ik} \alpha_{jk} = \delta_{ij} \quad (2.9)$$

donde

$$B_{ik} = \ell_i(\psi_k) \quad (2.10)$$

es un tipo de matriz generalizada de Vandermonde, claro podemos escribir la ecuación anterior como

$$B\alpha^T = I \quad (2.11)$$

y obtenemos

$$\alpha = B^{-T} \quad (2.12)$$

En la práctica, esto supone que uno tiene una implementación de la base original para que las acciones de los grados de libertad pueden ser calculados fácilmente. Los grados de libertad implica típicamente evaluación, diferenciación, integración y así sucesivamente.

## 2.8. Espacios de polinomios

Los elementos finitos en términos de una función de dimensión finita del espacio  $\mathcal{V}$ . Aunque no es estrictamente necesario, las funciones utilizadas en elementos finitos son típicamente polinomios. Si bien nuestra estrategia general será, dar cabida a bases no polinómicas, sólo nos ocuparemos



de polinomios. El espacio más común es  $[\mathcal{P}_q]^d$ . El espacio polinomial de grado  $q$  en  $\mathbb{R}^d$ . Hay muchas maneras diferentes para representar  $[\mathcal{P}_q]^d$ . Los más comunes que se utilizan son *Power basis*, *Legendre basis*, *Jacobi basis*, *Dubiner basis*, *Bernstein basis*, *Homogeneous polynomials*, *Vector or tensor-valued polynomials*. En 2D sobre triángulos,  $[\mathcal{P}_q]^2$  es expandido por funciones de la forma  $\{x^i y^j\}_{i,j=0}^{i+j \leq q}$ , con una definición similar en tres dimensiones.

## 2.9. El mapeo del elemento de referencia

Una práctica común, empleada en el software FEniCS y en muchos otros códigos de elementos finitos, es mapear las funciones de base nodales de la celda de referencia a cada celda de una malla. A veces, esto es tan simple como un cambio de coordenadas afín; en otros casos es más complicado. Para completar, se describen brevemente los conceptos básicos de la creación de los elementos finitos globales en términos de un elemento de referencia asignada. Así por lo tanto  $T$  es un polígono global en la malla y  $\hat{T}$  es el correspondiente polígono de referencia.

$$x = F_T(\hat{x}) = A_T(\hat{x}) + x_0 \quad (2.13)$$

El jacobiano de este mapeo es:

$$J(\hat{x}) = \frac{\partial x}{\partial \hat{x}} = \frac{\partial A_T(\hat{x})}{\partial \hat{x}} \quad (2.14)$$

Actualmente, FEniCS sólo es compatible con los mapas afines entre  $T$  y  $\hat{T}$ , lo que significa que  $x = F_T \hat{x} + x_0$  y  $J = A_T$ . Para los elementos isoparamétricos una función de base se define en términos de la función de base correspondiente en el elemento de referencia como:

$$\phi(x) = \hat{\phi}(\hat{x}) \quad (2.15)$$

A continuación la integral, se puede realizar en el polígono de referencia,

$$\int_T \phi(x) dx = \int_{\hat{T}} \hat{\phi}(\hat{x}) \det J d\hat{x} \quad (2.16)$$

y las derivadas espaciales se definen por las derivadas en el elemento de referencia y la asignación de la geometría mediante el uso de la regla de la cadena,

$$\frac{\partial \phi}{\partial x_i} = \sum_j \frac{\partial \hat{\phi}}{\partial \hat{x}_j} \frac{\partial \hat{x}_j}{\partial x_i} \quad (2.17)$$

El mapeo por encima de funciones de base es común para las aproximaciones en  $H^1$ . Para aproximaciones en  $H(\text{div})$  o  $H(\text{curl})$  es necesario utilizar el mapeo Piola, donde la asignación para las

funciones de base se diferencia de la asignación de la geometría. Esto es para elementos  $H(div)$ , el mapeo Piola se lee

$$\phi(x) = \frac{1}{|\det J|} J \hat{\phi}(\hat{x}) \quad (2.18)$$

Cuando se utiliza la numeración de las entidades de malla, es ventajoso utilizar  $\frac{1}{\det J}$  en lugar de  $\frac{1}{|\det J|}$  ya que el signo del determinante en relación con la señal de la vector normal.

## 2.10. Mapeo de local a global de los grados de libertad

Una alternativa es relacionar cada grado de libertad en la celda de referencia a un punto en la celda de referencia. El mapeo de la geometría a continuación, da un punto global en la malla, Por la ecuación (2.13), que identifica el grado de libertad; es decir, los grados de libertad en diferentes elementos son compartidos si corresponden al mismo punto global en la malla. Alternativamente, cada grado de libertad puede estar relacionado con una entidad de red local, como un vértice, borde o cara, sobre el elemento de referencia. después de mapear el elemento, el grado de libertad a continuación, estará relacionado con la entidad de red correspondiente en la malla global.

**Ejemplo: Los elementos de Lagrange.** *Los elementos de Lagrange se muestran en la figura 2.1 estos son los elementos más comunes. Los grados de libertad están representados por puntos negros, los cuales representan el punto de evaluación. El orden del primer elemento se muestra en el triángulo de la izquierda, sus grados de libertad consisten en una evaluación punto en cada uno de los vértices. Es decir, los grados de libertad  $\ell_i : \mathcal{V} \rightarrow \mathbb{R}$*

$$\ell_i(v) = \int_T v \delta_{x_i} dx = v(x_i) \quad (2.19)$$

donde  $x_i$  son los vértices  $(0,0)$ ,  $(1,0)$ ,  $(0,1)$  y  $\delta$  es la función delta de Dirac. Las funciones de base correspondientes son  $1 - x - y$ ,  $x$ ,  $y$  y. El elemento de segundo orden se muestra en el triángulo de la derecha. Este tiene seis grados de libertad.

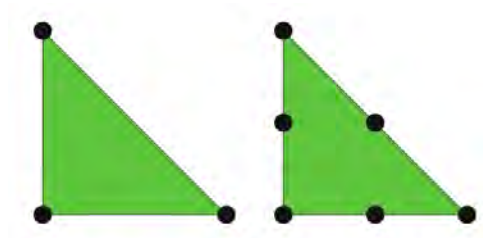


Figura 2.1: Elementos de Lagrange de uno y dos órdenes.

## 2.11. Formulación variacional de Elementos Finitos

Para la formulación y su análisis teórico tomado de [Chen \(2005\)](#), particularmente en los espacios de Sobolev [Ciarlet \(2002\)](#),  $H^r(\Omega)$ ,  $r = 1, 2, \dots$ . Un espacio lineal  $V$ , junto con un producto interior  $(\cdot, \cdot)$ , definido sobre éste, es llamado espacio de producto interior y es representado por  $(V, (\cdot, \cdot))$ . Con el producto interior  $(\cdot, \cdot)$  existe una norma asociada definida en  $V$ :

$$\|v\| = \sqrt{(v, v)} \quad v \in V. \quad (2.20)$$

Por lo tanto, un espacio con producto interno puede ser siempre hecho para ser un espacio lineal normado. Si el espacio lineal correspondiente  $(V, (\cdot, \cdot))$  es completo, entonces  $(V, (\cdot, \cdot))$  es llamado espacio de Hilbert. Los espacios de Hilbert  $H^r(\Omega)$ , ( $r = 1, 2, \dots$ ), con producto interior

$$(u, v)_{H(\Omega)} = \sum_{|\alpha| \leq r} \int_{\Omega} D^{\alpha} u(x) D^{\alpha} v(x) dx \quad u, v \in H^r(\Omega) \quad (2.21)$$

y su norma correspondiente  $\|\cdot\|_{H^r(\Omega)}$ , es un espacio de Hilbert. Ahora suponemos que  $V$  es un espacio de Hilbert con el producto escalar  $(\cdot, \cdot)$  y su norma correspondiente  $\|\cdot\|_V$ . Así  $a(\cdot, \cdot) : V \times V \rightarrow \mathbb{R}$  y esto es su forma bilineal en sentido que

$$\begin{aligned} a(u, \alpha v + \beta w) &= \alpha a(u, v) + \beta a(u, w), \\ a(\alpha u + \beta v, w) &= \alpha a(u, w) + \beta a(v, w) \end{aligned} \quad (2.22)$$

para  $\alpha, \beta \in \mathbb{R}$ ,  $\alpha, \beta \in \mathbb{R}$ . Ahora asumimos que  $L : V \rightarrow \mathbb{R}$  es un funcional lineal. Vamos a definir el funcional  $\alpha, \beta \in \mathbb{R}$  por:

$$F(v) = \frac{1}{2} a(v, v) - L(v), \quad v \in V. \quad (2.23)$$

Consideremos ahora el problema de minimización abstracta

$$\text{Hallar } p \in V \text{ tal que } F(p) \leq F(v) \quad \forall v \in V \quad (2.24)$$

y el problema variacional abstracta es

$$\text{Hallar } p \in V \text{ tal que } a(p, v) = L(v) \quad \forall v \in V \quad (2.25)$$

Para analizar las ecuaciones (2.24) y (2.25) necesitamos algunas propiedades de  $a$  y  $L$ :

- $a(\cdot, \cdot)$  es simétrica si

$$a(u, v) = a(v, u) \quad \forall u, v \in V \quad (2.26)$$

- $a(\cdot, \cdot)$  es continua o acotada en la norma  $\|\cdot\|_v$  si existe una constante  $a^* > 0$  tal que

$$|a(u, v)| \leq a^* \|u\|_V \|v\|_V \quad \forall v, u \in V \quad (2.27)$$

- $a(\cdot, \cdot)$  es  $V$ -elíptica o coactivo si existe una constante  $a_* > 0$  tal que

$$|a(v, v)| \geq a_* \|v\|_V^2 \quad \forall v, u \in V \quad (2.28)$$

- $L$  es acotada en la norma  $\|\cdot\|_V$ :

$$|L(v)| \leq \hat{L} \|v\|_V \quad \forall v \in V. \quad (2.29)$$

Por lo tanto, de acuerdo al teorema la representación de Riesz, existe una única  $p \in V$  tal que es decir

$$a(p, v) = L(v) \quad \forall v \in V, .$$

## 2.12. Espacios de elementos finitos

Los elementos finitos definido se ensamblan para construir *espacios de elementos finitos* generando aproximaciones de funciones definidas en principio en dominios arbitrarios tomado [Logg et al. \(2012\)](#).

Sea  $\Omega$  el dominio en donde está definida la EDPs a la partición (triangulación) del dominio la cual está formada por pequeños intervalos llamados *elementos*.

### 2.12.1. Triángulos

Consideramos espacios generales de elementos finitos. Primero trataremos el caso  $\Omega \subset \mathbb{K}$  es un dominio poligonal en el plano. Así  $K_h$  es la triangulación del dominio  $\Omega$  dentro de los triángulos  $K$ . Introduciremos la notación

$$P_r(K) = \{v : v \text{ es un polgono de grado a lo mas } r \text{ en } K\}.$$

donde  $r = 1, 2, \dots$ . Para  $r = 1$   $P_1(K)$  es un espacio de funciones lineal, de la forma

$$v(\mathbf{x}) = v_{00} + v_{10}x_1 + v_{01}x_2 \quad \mathbf{x} = (x_1, x_2) \in K, v \in P_1(K)$$

donde  $v_{ij} \in \mathbb{R}$ ,  $i, j = 0, 1$ .

Note que la dimensión  $(P_1(K)) = 3$ ; es decir, esta dimensión es tres.

Para  $r = 2$ ,  $P_2(K)$  es el espacio de funciones cuadráticas de  $K$ :

$$v(\mathbf{x}) = v_{00} + v_{10}x_1 + v_{01}x_2 + v_{20}x_1^2 + v_{11}x_1x_2 + v_{02}x_2^2, \quad v \in P_2(K)$$

donde  $v_{ij} \in \mathbb{R}$ ,  $i, j = 0, 1, 2$ . Vemos que la dimensión  $(P_2(K)) = 6$ .

En general, tenemos

$$P_r(K) = \left\{ v : v(\mathbf{x}) = \sum_{0 \leq i+j \leq r} v_{ij} x_1^i x_2^j, \quad \mathbf{x} \in K, \quad v_{ij} \in \mathbb{R} \right\}, \quad r \geq 0,$$

Así que

$$\dim(P_r(K)) = \frac{(r+1)(r+2)}{2}$$

*ejemplo 1.* Definir

$$V_h = \{v : v \text{ es continua en } \Omega \text{ y } v|_K \in P_1(K), K \in K_h\},$$

donde  $v|_K$  representa la restricción  $v$  en  $K$ . Como parámetros o grados globales de libertad, para describir las funciones de  $V_h$ , utilizamos los valores en los vértices (nodos) de  $K_h$ . Se puede demostrar que esta es una opción legítima; es decir una función de  $V_h$  es determinada únicamente por estos grados de libertad global. Para ver esto, para cada triángulo  $K \in K_h$ , estos vértices son indicados por  $\mathbf{m}_1, \mathbf{m}_2$  y  $\mathbf{m}_3$ ; ver figura(2.2) Además, y las funciones (local) de bases sean  $\lambda_i$ ,  $i = 1, 2, 3$ , que se definen por

$$\lambda_i(\mathbf{m}_j) = \begin{cases} 1, & \text{si } i = j, \\ 0, & \text{si } i \neq j \quad i, j = 1, 2, 3. \end{cases}$$

Estas funciones de base se pueden determinar en el siguiente enfoque:

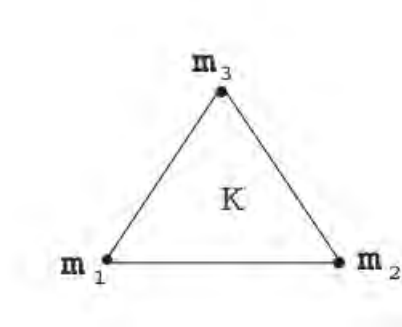


Figura 2.2: Los grados de libertad para elementos  $P_1(K)$

Dada la ecuación de la línea recta que pasa por los vértices  $\mathbf{m}_2$  y  $\mathbf{m}_3$  dada por

$$c_0 + c_1x_1 + c_2x_2 = 0$$

y luego definimos

$$\lambda_1(\mathbf{x}) = \gamma(c_0 + c_1x_1 + c_2x_2), \quad \mathbf{x} = (x_1, x_2),$$

donde la constante  $\gamma$  se elige de modo que  $\lambda_1(\mathbf{m}_1) = 1$ . Las funciones  $\lambda_2$  y  $\lambda_3$  se puede determinar con el mismo enfoque. Estas funciones  $\lambda_1$ ,  $\lambda_2$  y  $\lambda_3$  a veces se llaman las coordenadas baricéntricas de un triángulo. si  $K$  es el triángulo de referencia con vértices  $(1,0)$ ,  $(0,1)$  y  $(0,0)$ ,  $\lambda_1$ ,  $\lambda_2$  y  $\lambda_3$

son respectivamente  $x_1$ ,  $x_2$  y  $1 - x_1 - x_2$ . Ahora cualquier función  $v \in P_1(K)$  tienen una única representación

$$v(\mathbf{x}) = \sum_{i=1}^3 v(\mathbf{m}_i) \lambda_i(\mathbf{x}), \quad \mathbf{x} \in K.$$

Así  $v \in P_1(K)$  se determina de forma única por sus valores en los tres vértices. Por lo tanto, en cada triángulo  $K \in K_h$ , los grados de libertad de los elementos, que pueden ser estos valores (nodales). Estos grados de libertad son los mismos que los grados globales de libertad y se utilizan para construir las funciones de base en  $V_h$ . Afirmamos que para  $v$  tal que  $v|_K \in P_1(K)$ ,  $K \in K_h$ , si es continua en los vértices internos, entonces  $v \in C^0(\bar{\Omega})$ . Obviamente, es suficiente para mostrar que  $v$  es continua a través de todos los bordes entre los elementos. Así los triángulos  $K_1, K_2 \in K_h$  comparten un borde común  $e$  con los puntos finales  $\mathbf{m}_1$  y  $\mathbf{m}_2$ . Y el conjunto  $v_i = v|_{K_i} \in P_1(K_i)$ ,  $i = 1, 2$ . Entonces la diferencia  $v_1 - v_2$  definida sobre  $e$  se anula en  $\mathbf{m}_1$  y  $\mathbf{m}_2$ . Porque  $v_1 - v_2$  es lineal sobre  $e$ , se anula en todo el borde  $e$ . Por lo tanto  $v$  es continua en  $e$  y se prueba la afirmación de que  $v \in C^0(\bar{\Omega})$ . Para un problema con una condición de frontera esencial, esta condición debe ser incorporado en la definición de  $V_h$ .



## El Proyecto FEniCS

El Proyecto FEniCS es un proyecto de investigación y software libre destinado a crear métodos matemáticos y software para el modelado matemático computacional automatizado. Esto significa crear software fácil, intuitivo, eficiente y flexible para resolver ecuaciones diferenciales parciales (EDPs) usando métodos de elementos finitos. FEniCS consta de varios bloques de construcción (componentes de software) que juntos forman el software el proyecto FEniCS: DOLFIN, FFC, FIAT, UFL, mshr entre otros.

### 3.1. Interfaz Python

Python ha surgido como una opción atractiva para el rápido desarrollo de programas de simulación para la computación científica. Es sección tomada para ilustración de la tesis de [Logg \*et al.\* \(2012\)](#). Python ofrece los beneficios de un lenguaje de programación de alto nivel, la fuerza de un lenguaje orientado a objetos y una gran cantidad de bibliotecas para el cálculo numérico. La mayor parte de la interfaz DOLFIN en Python se genera automáticamente a partir de la interfaz de C++. Dado que la funcionalidad tanto de C++ y las interfaces de Python se implementan como parte de la biblioteca DOLFIN C++, DOLFIN es igualmente eficiente a través de C++ y las interfaces de Python para la mayoría de operaciones. En particular, la forma de lenguaje UFL está perfectamente integrado en la interfaz de Python y la generación de código se maneja automáticamente en tiempo de ejecución.

### 3.2. Componentes que forman el proyecto FEniCS.

El proyecto FEniCS busca automatizar la resolución de ecuaciones diferenciales, utilizando el método de los elementos finitos. FEniCS proporciona una interfaz en Python para todos estos paquetes, la mayoría escritos en Fortran o C++, y eso le da una sencillez de uso inigualable.



La figura (3.1) presenta una visión general de las relaciones entre los componentes de FEniCS y software externo.

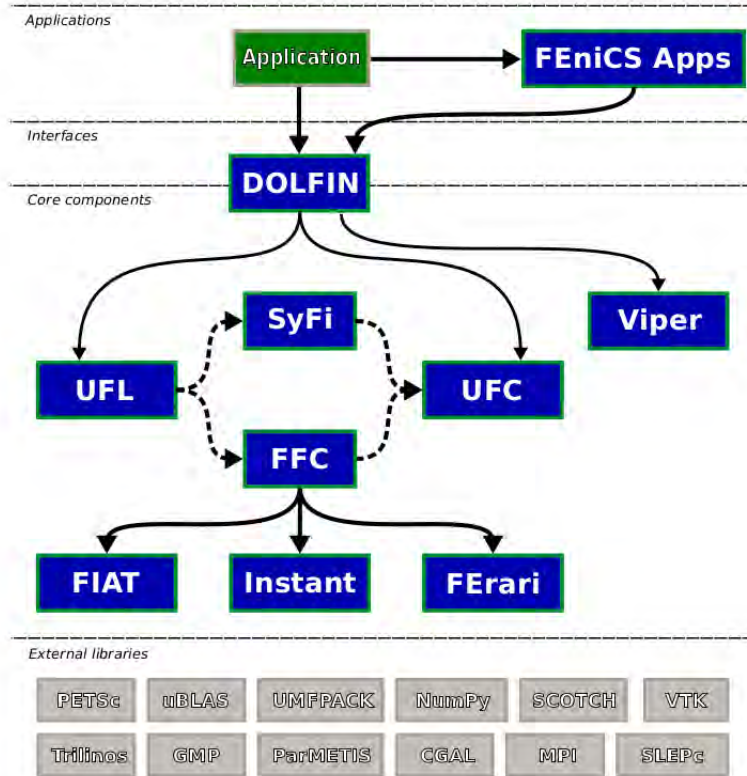


Figura 3.1: DOLFIN funciona como la interfaz de usuario principal de FEniCS y se encarga de la comunicación entre los diversos componentes de FEniCS y software externo.

En la siguiente figura (3.1) se presenta una visión general de las relaciones entre los componentes de FEniCS y las bibliotecas externas. El esquema del software presentado en la figura muestra una aplicación de usuario que en este caso es el proyecto FEniCS implementada en la parte superior de la interfaz de usuario DOLFIN, desarrollada ya sea en C++ o en Python. Las aplicaciones de usuario también pueden desarrollarse utilizando FEniCS Apps, que es una colección de solucionadores implementadas en la parte superior de FEniCS/DOLFIN. DOLFIN a su vez funciona como una interfaz de usuario y un componente central de FEniCS. Toda la comunicación entre un programa de usuario, y otros componentes básicos de FEniCS y las bibliotecas externas se dirigen a través de capas de envoltura que se implementan como parte de la interfaz de usuario de DOLFIN. En particular, las formas variacionales expresadas en el lenguaje de formulario UFL (Código de ensamblaje unificado) se pasan al compilador de formularios FFC (Compilador para formas variacionales de elementos finitos) o SFC (SFC traduce el código UFL al código UFC). para generar el código UFC (interfaz de generación de código de elementos finitos), que puede ser usado por DOLFIN para ensambles lineales. En el caso de FFC, esta generación de código

depende del motor de elementos finitos FIAT (paquete de Python para la generación automática de funciones de base de elementos finitos), de la utilidad de recopilación en tiempo real Instant (es utilizado por FFC y DOLFIN para just-in-time (JIT) compilación de formas variacionales y expresiones) y de la opción optima del motor de FErari (proporcionan una aceleración modesta a considerable en la evaluación en tiempo de ejecución de las formas variacionales). Por último, las capacidades para graficar proporcionadas por DOLFIN son implementadas por Viper (Viper es un graficador científico minimalista y un módulo de visualización en tiempo de ejecución). Parte de esta comunicación está expuesta a los usuarios de la interfaz DOLFIN C++, que requiere que un usuario genere explícitamente código UFC desde un archivo de formulario UFL llamando a un compilador de formularios en la línea de comandos. DOLFIN también se basa en software externo para funciones importantes como el álgebra lineal. Las bibliotecas PETSc (conjunto de herramientas portables y extensibles para computo científico), Trilinos, uBLAS, y las bibliotecas de partición de malla ParMETIS y SCOTCH (Pellegrini).

En informática, una biblioteca es una colección de recursos utilizados por programas de computadora, a menudo para desarrollar software. Estos pueden incluir datos de configuración, documentación, datos de ayuda, plantillas de mensajes, código y subrutinas preescritas, clases, valores o tipo especificaciones.

## DOLFIN: Una biblioteca de elementos finitos

DOLFIN es una biblioteca de C++/Python que funciona como la interfaz de usuario principal de FEniCS. Se proporciona una resolución de problemas de medio ambiente para los modelos basados en ecuaciones diferenciales parciales. DOLFIN implementa estructuras de datos tales como mallas, espacios de funciones y algoritmos de cálculo intensivo tales como el ensamble de elementos finitos y el refinamiento de malla, e interfaces con solucionadores de álgebra lineal y estructuras de datos como PETSc (the Portable, Extensible Toolkit for Scientific Computation), conjunto de herramientas portables y extensibles para computo científico. Por último, las capacidades de graficación proporcionadas por DOLFIN son implementados por Viper (es un graficador científico minimalista y un módulo de visualización en tiempo de ejecución. Viper tiene soporte para visualizar mallas y soluciones en DOLFIN). DOLFIN también se basa en un software externo para las funcionalidades importantes, tales como las bibliotecas de álgebra lineal PETSc, Trilinos, uBLAS. Una de las características clave de FEniCS es que está automatizado para la generación de código para la solución general y eficiente de problemas variacionales de elementos finitos. Esta generación automática de código se basa en un compilador para la compilación de código en tiempo de ejecución.

## UFL (El lenguaje de formulario unificado)

UFL es un lenguaje específico de dominio para la declaración de discretización de elementos finitos de las formas variacionales. Más precisamente, se define una interfaz flexible para elegir espacios de elementos finitos y la definición de las expresiones de las formas débiles en una notación matemática.

### SyFi: elementos finitos simbólicos y forma de compilación

Paquete SyFi es una biblioteca de C++ construida sobre la biblioteca matemática simbólica GiNaC. El nombre SyFi significa Elementos Finitos Simbólicos. El paquete proporciona dominios poligonales, espacios polinomiales y grados de libertad como expresiones simbólicas que son fácilmente manipulables. Esto facilita la definición de elementos finitos y formas variacionales. Estos elementos y formularios se utilizan para generar código C++ eficiente.

## FFC (Compilador para formas variacionales de elementos finitos)

En términos simples, la solución de los problemas variacionales de elementos finitos se basa en dos principios: el ensamble de sistemas lineales o no lineales de ecuaciones y la solución de esas ecuaciones. En particular, una parte central de la mayoría de los códigos de elementos finitos es el conjunto de matrices dispersas de formas bilineales elemento finito. FFC toma como entrada una forma variacional se especifica en el lenguaje de formas UFL y genera como salida código C++ que se ajusta a la interfaz de UFC.

La figura (3.2) nos dice que a partir de una entrada que describe un problema variacional de elementos finitos en la notación matemática, el compilador de formularios FFC genera código para el cálculo eficiente, más específicamente, FFC toma como entrada una forma variacional especificada en el lenguaje de formulario UFL y genera como salida un código C++ que se ajusta a la interfaz UFC.



Figura 3.2: La forma compilador FFC genera código C++ en formato UFC de una forma variacional de elementos finitos dada en formato UFL.

## UFC: una interfaz de generación de código de elementos finitos

UFC (Código de ensamblaje unificado) es un marco unificado para el ensamble de elementos finitos. Se define una interfaz fija para la comunicación de rutinas de bajo nivel (funciones) para la evaluación y el ensamble de las formas variacionales de elementos finitos. Un componente central de FEniCS es la interfaz de UFC (Código unificado del formulario de ensamble). UFC es una interfaz entre los componentes de problemas específicos y de uso general de programas de elementos finitos. La interfaz de UFC se aplica a una amplia gama de problemas de elementos finitos (incluyendo elementos finitos mixtos y métodos de Galerkin discontinuos) y puede ser utilizado con las bibliotecas que difieren ampliamente en su diseño. La interfaz de UFC consiste en un único archivo cabeceras (`ufc.h`) que especifica una interfaz de C++ que debe ser implementado por el código que cumpla con la especificación de UFC.

## FIAT

FIAT es un paquete de Python para la generación automática de funciones de base de elementos finitos. Trabaja principalmente en términos de álgebra lineal numérica. Es capaz de generar funciones de base de elementos finitos para una amplia gama de familias de elementos finitos en simples (líneas, triángulos y tetraedros), incluyendo los elementos de Lagrange, y los elementos de Raviart-Thomas, Brezzi-Douglas-Marini y Nedelec. También es capaz de generar elementos tensor-producto y un número de elementos más exóticos, como los elementos Argyris, Hermite y Morley.

## Instant

Instant es un módulo de Python que permite la expansión en línea instantánea de código C y C++ en Python. Instant es utilizado por FFC y DOLFIN para just-in-time (JIT) compilación de formas variacionales y expresiones en tiempo de ejecución.

## FErari

En la mayoría de los casos, las optimizaciones basadas en FErari proporcionan una aceleración modesta a considerable en la evaluación en tiempo de ejecución de las formas variacionales. Puede aumentar considerablemente el tiempo que FFC requiere para generar código y, por lo tanto, son menos adecuados para una fase de desarrollo o una estrategia de compilación just-in-time. Como regla general, también se puede afirmar que la cuadratura se hace más eficiente en relación con la contracción del tensor cuando la complejidad de una forma aumenta según se mide en el número de

coeficientes y el número de operadores diferenciales, mientras que el enfoque tensor de contracción es relativamente más eficiente para formas y polinomios de alto orden.

### 3.3. Ilustración de un ejemplo del proyecto FEniCS 2016.1

Para la ilustración de toda esta sección se tomo del Tutorial Solving PDEs in Python The FeniCS Tutorial 1. Capitulo 2 pag.11 [Petter Langtangen y Logg \(2016\)](#).

Al aplicar los métodos de elementos finitos para EDPs, lo fundamental debe ser resolver las ecuaciones diferenciales.

Por tanto, se resuelve el siguiente problema de valor inicial y de frontera unidimensional:

$$-\nabla^2 u(\mathbf{x}) = f(\mathbf{x}), \quad \mathbf{x} \in \Omega, \quad (3.1)$$

$$u(\mathbf{x}) = u_D(\mathbf{x}), \quad \mathbf{x} \text{ sobre } \partial\Omega \quad (3.2)$$

donde  $u = u(\mathbf{x})$  es la función desconocida,  $f = f(\mathbf{x})$  es una función prescrita,  $\nabla^2$  es el operador Laplaciano,  $\Omega$  es el dominio espacial y  $\partial\Omega$  es la frontera del dominio  $\Omega$ .

En un espacio de dos dimensiones con coordenadas  $x$  y  $y$  podemos escribir esta misma ecuación como:

$$-\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = f(x, y). \quad (3.3)$$

donde  $u$  es ahora la nueva función desconocida de dos variables,  $u = u(x, y)$ , definido en un dominio espacial ( $\Omega$ ) de dos dimensiones.

La ecuación de Poisson surge en numerosos contextos físicos, incluyendo conducción de calor, electrostática, difusión de sustancias, torsión de barras elásticas, flujo de fluido no viscoso y ondas de agua. Además, la ecuación aparece en las estrategias de división numérica para sistemas más complicados de EDPs, en particular las ecuaciones de Navier-Stokes.

Resolver una EDPs tal como la ecuación de Poisson en FEniCS consiste en los siguientes pasos: Identificar el dominio para la implementación computacional ( $\Omega$ ), la EDPs, las condiciones de frontera y los términos fuentes ( $f$ ).

Discretizar (EDPs) como un problema variacional de elementos finitos.

Realizar la implementación en Python que defina el dominio computacional, el problema variacional, las condiciones de frontera y los términos fuente.

Usar FEniCS para resolver la EDPs y, opcionalmente, extienda el programa para calcular cantidades derivadas tales como flujos y promedios, y visualice los resultados.

## Formulación variacional de elementos finitos

FEniCS se basa en el método de elementos finitos, que es una maquinaria matemática general y eficiente para la solución numérica de EDPs. El punto de partida para los métodos de elementos

finitos es un EDPs expresado en forma variacional. La receta básica para convertir un EDPs en un problema variacional es multiplicar el EDPs por una función  $v$ , integrar la ecuación resultante sobre el dominio  $\Omega$ , y realizar la integración por partes de términos con derivadas de segundo orden. La función  $v$  por la cual multiplicamos la EDPs es llamada *función de prueba*. La función desconocida  $u$  que se aproxima se denomina *función de base*. Las funciones de base y prueba pertenecen a ciertos espacios de funciones que especifican las propiedades de las funciones. En el presente caso, primero multiplicamos la ecuación de Poisson por la función de prueba  $v$  (test) e integramos sobre el dominio  $\Omega$ :

$$-\int_{\Omega} (\nabla^2 u) v dx = \int_{\Omega} f v dx \quad (3.4)$$

Aquí  $dx$  denota el elemento diferencial para la integración sobre el dominio  $\Omega$ . Posteriormente,  $ds$  denotará el elemento diferencial para la integración sobre el límite de  $\Omega$ .

Una regla común cuando derivamos formulaciones variacionales es que tratamos de mantener el orden de las derivadas de  $u$  y  $v$  lo más bajo posible (esto ampliará la colección de elementos finitos que se pueden usar en el problema). Aquí, tenemos una derivada espacial de segundo orden de  $u$ , que puede transformarse en una primera derivada de  $u$  y  $v$  aplicando la técnica de integración por partes. La fórmula es

$$-\int_{\Omega} (\nabla^2 u) v dx = \int_{\Omega} \nabla u \cdot \nabla v dx - \int_{\partial\Omega} \frac{\partial u}{\partial n} v ds \quad (3.5)$$

Aquí  $\frac{\partial u}{\partial n} = \nabla u \cdot n$  es la derivada de  $u$  en dirección normal hacia afuera  $n$  sobre la frontera. Otra característica de las formulaciones variacionales es que se requiere que la función de prueba  $v$  se anule en las partes de la frontera donde se conoce la solución  $u$ . En el presente problema, esto significa que  $v = 0$  en toda la frontera  $\partial\Omega$ . Por lo tanto, el segundo término en el lado derecho de la ecuación (3.5) desaparece. De la ecuación (3.4) y la ecuación (3.5) se deduce que

$$\int_{\Omega} \nabla u \cdot \nabla v dx = \int_{\Omega} f v dx \quad (3.6)$$

Si se requiere que esta ecuación se mantenga para todas las funciones de prueba  $v$  en algún espacio adecuado  $\hat{V}$ , llamado espacio de prueba. Obtenemos un problema matemático bien definido que determina de manera única la solución  $u$  que se encuentra en algún espacio de función  $V$ , el llamado espacio de prueba. Nos referimos a la ecuación (3.6) como la forma débil o la forma variacional del problema de valor de límite original las ecuaciones (3.1-3.2). La declaración apropiada de nuestro problema variacional ahora va como sigue: Encontrar  $u \in V$  tal que:

$$\int_{\Omega} \nabla u \cdot \nabla v dx = \int_{\Omega} f v dx \quad \forall v \in \hat{V} \quad (3.7)$$

Los espacios de prueba y los espacios base  $V$  y  $\hat{V}$  presentados en el problema se definen como:

$$V = \{v \in H^1(\Omega) : v = u_D \in \partial\Omega\},$$

$$\hat{V} = \{v \in H^1(\Omega) : v = 0 \in \partial\Omega\},$$

En resumen,  $H^1(\Omega)$  es el espacio matemáticamente bien conocido de Espacios de Sobolev que contiene funciones  $v$  tales que  $v^2$  y  $|\nabla v|^2$  tienen integrales finitas sobre  $\Omega$ . La solución de la EDP subyacente debe encontrarse en un espacio de función donde las derivadas son también continuas, pero el espacio de Sobolev  $H^1(\Omega)$  permite funciones con derivadas discontinuas. El problema variacional ecuación (3.7) es un problema continuo: define la solución  $u$  en el espacio funciones de dimensión infinita  $V$ . El método de elementos finitos para la ecuación de Poisson encuentra una solución aproximada del problema variacional ecuación (3.7) reemplazando los espacios funcionales de dimensión infinita  $V$  y  $\hat{V}$  por un espacio de prueba y base de dimensión finita discreto  $V_h \subset V$  y  $\hat{V}_h \subset \hat{V}$ . El problema variacional discreto consiste: Encontrar  $u_h \in V_h \subset V$  tal que

$$\int_{\Omega} \nabla u_h \cdot \nabla v dx = \int_{\Omega} f v dx \quad \forall v \in \hat{V}_h \subset \hat{V}. \quad (3.8)$$

Este problema variacional, junto con la definición adecuada de los espacios de funciones  $V_h$  y  $\hat{V}_h$ , definen de manera única nuestra solución numérica aproximada de la ecuación de Poisson, ecuación (3.1)

## Forma Variacional Abstracta de Elementos Finitos

Resulta conveniente introducir la siguiente notación canónica para los problemas variacionales: Encontrar  $u \in V$  tal que

$$a(u, v) = L(v) \quad \forall v \in \hat{V} \quad (3.9)$$

para la ecuación de Poisson, tenemos:

$$a(u, v) = \int_{\Omega} \nabla u \cdot \nabla v dx \quad (3.10)$$

$$L(v) = \int_{\Omega} f v dx \quad (3.11)$$

De la literatura matemática,  $a(u, v)$  es conocido como *la forma bilineal* y  $L(v)$  como *la forma lineal*. En cada problema lineal que resolvamos, identificaremos los términos desconocido  $u$  y los recogeremos en  $a(u, v)$ , y similarmente recogemos los términos conocidos en  $L(v)$ . Las fórmulas que corresponden a  $a$  y  $L$  se codifican directamente en el programa. FEniCS proporciona toda la notación matemática necesaria para expresar el problema variacional  $a(u, v) = L(v)$ .

Para resolver una EDP lineal en FEniCS, como la ecuación de Poisson, se necesita realizar tan sólo dos pasos:

1. Elegir los espacios de elementos finitos  $V$  y  $\hat{V}$ , especificando el dominio (la malla) y el tipo de espacio de función (grado polinomial y tipo).
2. Expresar la PDEs como un problema variacional (discreto): Encontrar  $u \in V$  tal que  $a(u, v) = L(v)$  para toda  $v \in \hat{V}$ .

## Estructura del programa

El listado del programa FEniCS define una malla de elementos finitos, un espacio de funciones de elementos finitos  $V$  sobre esta malla, las condiciones de frontera para  $u$  (la función  $u_D$ ) y las formas bilineal y lineal  $a(u, v)$  y  $L(v)$ . Posteriormente, se calcula la función de prueba desconocida  $u$ . Entonces podemos comparar la solución numérica y la exacta así como visualizar la solución calculada  $u$ .

## Generación de mallas

Una parte central de DOLFIN es su biblioteca de mallas. La biblioteca de mallas proporciona estructuras de datos y algoritmos para mallas computacionales, incluyendo el cálculo de conectividad de malla (relaciones de incidencia), el refinamiento de la malla, separación de la malla y la intersección de la malla.

**La creación de una malla.** DOLFIN proporciona funcionalidad para la creación de mallas simples, tales como mallas de cuadrados unitarios y cubos unitarios, esferas, rectángulos y cajas.

```
mesh = UnitSquareMesh(8, 8)
```

Define una malla uniforme de elementos finitos sobre el cuadrado unitario  $[0, 1][0, 1]$ . La malla consiste en celdas, que son triángulos en  $2D$  con lados rectos.

**La lectura de una malla a partir de archivos.** Una aplicación típica tendrá que leer desde el archivo de una malla que se ha sido generada por un generador de malla externa. Para leer una malla de archivo, simplemente suministrar el nombre del archivo al constructor de la clase de malla:

```
mesh = Mesh("mesh.xml")
```

## Espacio de funciones de elementos finitos

Con una malla, podemos definir un espacio de función de elementos finitos  $V$  sobre esta malla:

```
V = FunctionSpace(mesh, "P", 1)
```

El segundo argumento "P", especifica el tipo de elemento, mientras que el tercer argumento es el grado de las funciones de base del elemento. El tipo de elemento aquí es "P", lo que implica la familia de elementos de Lagrange estándar. También puede utilizar "Lagrange" para especificar este tipo de elemento. El tercer argumento 1 especifica el grado del elemento finito. En este caso, el elemento Lagrange lineal  $P_1$  estándar, que es un triángulo con nodos en los tres vértices.

## Definición de las funciones de base y prueba

En matemáticas, distinguimos entre los espacios de base y los espacios de prueba  $V$  y  $\hat{V}$ . La única diferencia en el problema actual son las condiciones de frontera. En FEniCS no es-



pecificamos las condiciones de frontera como parte del espacio de funciones, por lo que basta con trabajar con un espacio común  $V$  para las funciones de prueba y de base en el programa:

```
u = TrialFunction(V)
v = TestFunction(V)
```

## Definiendo las condiciones de frontera

El siguiente paso es especificar la condición de frontera:  $u = u_D$  en  $\partial\Omega$ .

Esto es realizado por

```
bc = DirichletBC(V, u_D, boundary)
```

Donde  $u_D$  es una expresión que define los valores de la solución en la frontera, y `boundary` es una función (u objeto) que define qué puntos pertenecen a la frontera. La variable  $u_D$  se refiere a un objeto de expresión, que se utiliza para representar una función matemática.

La construcción típica es

```
u_D = Expression(formula, degree=1)
```

donde `formula` es una cadena que contiene una expresión matemáticas.

La función `boundary` especifica qué puntos que pertenecen a la parte de la frontera donde debe aplicarse la condición de frontera:

```
def boundary(x, on boundary):
    return on boundary
```

## El término fuente

Antes de definir las formas bilineal y lineal  $a(u, v)$  y  $L(v)$  tenemos que especificar el término fuente  $f$ :

```
f = Expression("6", degree=0)
```

Cuando  $f$  es constante sobre el dominio,  $f$  puede representarse más eficientemente como una constante:

```
f = Constant(-6)
```

## Funciones

La clase de función representa a los elementos finitos es una función  $u$  en un espacio de elementos finitos  $V$ .

**Creación de funciones.** Para crear una función en un espacio funcional, uno simplemente llama al constructor de la clase de funciones con el espacio funcional como el argumento.

```
u = Function(V)
```

Si dos o más s función se crean en el mismo espacio funcional, el espacio funcional es compartida

entre las funciones. Una función se utiliza normalmente para mantener la solución computarizada para una ecuación diferencial parcial.

## El problema variacional

DOLFIN se basa en la cadena de herramientas FEniCS FIAT-UFL-FFC / SFC-UFC para la evaluación de las formas variacionales elemento finito. Formas variacionales expresadas en el lenguaje de forma UFL se compilan utilizando una de las formas compiladores FFC o SFC, y el código de UFC generada es utilizado por DOLFIN para evaluar (ensamblar) formas variacionales. El lenguaje de formas UFL permite una amplia gama de formas variacionales que debe expresarse en un lenguaje de cerca de la notación matemática, como se ejemplifica por las siguientes expresiones definen las formas bilineales y formas lineales para la discretización de un problema:

```
a = inner(sigma(u), epsilon(v))*dx
L = dot(f, v)*dx
```

## Ensamblaje de elementos finitos

Una funcionalidad básica de DOLFIN es el conjunto de formas variacionales elemento finito. Dada una forma variacional (a), DOLFIN ensambla el operador discreto correspondiente (A). El código siguiente muestra cómo montar un escalar (m), un vector (b) y una matriz (A) de un funcional (M), una forma lineal (L) y una forma bilineal (a), respectivamente:

```
m = assemble(M)
b = assemble(L)
A = assemble(a)
```

El conjunto de formas variacionales de la interfaz de Python automáticamente provoca la generación de código, el código generado se almacena en caché para su posterior reutilización.

## Graficando la solución

Una vez calculada la solución, se puede visualizar mediante el comando plot:

```
plot(u)
plot(mesh)
interactive()
```

**Formato DOLFIN XML.** XML es el formato nativo de DOLFIN. Una ventaja de XML es que es un formato robusto y legible. Si se comprimen los archivos, también hay poca sobrecarga en términos de tamaño de archivo en comparación con un formato binario.

El programa completo lo podemos ver a continuación:

```
from dolfin import *
# Creamos la malla y los espacios de funciones
mesh = UnitSquare(6, 4)
V = FunctionSpace(mesh, "Lagrange", 1)
# Definimos las condiciones de frontera
u0 = Expression("1 + x[0]*x[0] + 2*x[1]*x[1]")
def u0_boundary(x, on_boundary):
    return on_boundary
bc = DirichletBC(V, u0, u0_boundary)
# Se define el problema variacional
u = TrialFunction(V)
v = TestFunction(V)
f = Constant(-6.0)
a = inner(nabla_grad(u), nabla_grad(v))*dx
L = f*v*dx
# Calculamos la solución
u = Function(V)
solve(a == L, u, bc)
# Graficamos la solución y la malla
plot(u)
plot(mesh)
# Guardamos la solución con un archivo en formato VTK
file = File("poisson.pvd")
file << u
interactive()
```

## Modelo Matemático

Vamos a derivar el modelo considerando las hipótesis generales, usando el enfoque sistemático para la modelación de sistemas continuos, derivaremos a continuación las ecuaciones que constituyen el modelo matemático de transporte a través de un medio poroso [Díaz-Viera \*et al.\* \(2015\)](#). Tenemos que el modelo de transporte se le puede incorporar la hipótesis de que puede haber adsorción, y advección y difusión.

### 4.1. Hipótesis generales del modelo

Se consideran las siguientes hipótesis generales en el modelo.

1. Se tiene al menos una fase sólida ( $s$ ) y una fase fluida ( $f$ ).
2. Existe una componente  $i$  disuelta en la fase fluida que puede ser transportada por procesos de advección y de difusión.
3. La componente  $i$  se considera que no ocupa volumen en la fase sólida.
4. La superficie es energéticamente homogénea y las moléculas absorbidas no interactúan.
5. No hay transiciones de fase.
6. La componente  $i$  puede pasar de la fase fluida a la fase sólida y viceversa mediante procesos de adsorción (desorción).
7. Todos los sitios de adsorción son equivalentes, y cada sitio puede ocupar una sola molécula.
8. La adsorción es monocapa en la máxima adsorción.
9. El estado inicial de la porosidad es constante ( $\phi = constante$ ), pero se permite la variación dinámica de la porosidad debido a los procesos de taponamiento/destaponamiento.
10. La permeabilidad inicial es constante y es la misma en todas las direcciones, pero se puede modificar en función con la porosidad por los procesos (adsorción/desorción).

## 4.2. Derivación del modelo matemático

Aplicando y considerando las suposiciones establecidas en el modelo conceptual, podemos obtener el sistema de ecuaciones que gobiernan para el modelo de transporte usando la formulación axiomática previamente descrita para sistemas continuos [Herrera y Pinder \(2012\)](#)

## 4.3. Balance de masa de una componente $i$ en una fase fluida ( $f$ )

La propiedad extensiva masa de una componente  $i$  en una fase fluida  $f$  se puede expresar como sigue:

$$M_f^i(t) = \int_{B(t)} \phi S_f c_f^i d\underline{x} \quad (4.1)$$

donde  $\phi$  es la porosidad del medio poroso,  $S_f$  es la saturación de la fase fluida y  $c_f^i$  es la concentración de masa de la componente  $i$  en la fase fluida. A partir de la relación entre propiedad extensiva e intensiva dada en la ecuación (4.1) se pueden escribir las ecuaciones de balance global (4.2) y local (4.3) respectivamente.

Ecuación de balance global

$$\frac{d}{dt} M_f^i(t) = \int_{B(t)} g_f^i(\underline{x}, t) d\underline{x} + \int_{\partial B(t)} \underline{\tau}_f^i(\underline{x}, t) \cdot \underline{n} d\underline{x} \quad (4.2)$$

Ecuación de balance local

$$\frac{\partial(\phi S_f c_f^i)}{\partial t} + \nabla \cdot (\phi S_f c_f^i \underline{v}_f) = g_f^i + \nabla \cdot \underline{\tau}_f^i \quad (4.3)$$

se considera

- $\underline{u}_f \equiv \phi S_f \underline{v}_f$  es la velocidad de Darcy para la fase fluida,
- $\underline{\tau}_f^i = \phi S_f \underline{D}_f^i \cdot \nabla c_f^i$  es el flujo dispersivo de la componente  $i$  (Ley de Fick).
- $g_f^i = (g_f^i)_{ad} + (g_f^i)_{\Omega}$  es el término fuente, que está compuesto por:
  1.  $(g_f^i)_{ad}$  es la adsorción de la componente  $i$  en la fase sólida  $s$ .
  2.  $(g_f^i)_{\Omega}$  es un término fuente genérico diferente de la adsorción.

Entonces, se obtiene la ecuación de balance de masa de la componente  $i$  en la fase fluida ( $f$ )

$$\frac{\partial(\phi S_f c_f^i)}{\partial t} - \nabla \cdot (\phi S_f \underline{D}_f^i \cdot \nabla c_f^i) + \nabla \cdot (\underline{u}_f c_f^i) = (g_f^i)_{ad} + (g_f^i)_{\Omega} \quad (4.4)$$

## 4.4. Balance de masa de una componente $i$ en la fase sólida ( $s$ )

La propiedad extensiva masa de una componente  $i$  en una fase sólida  $s$  se puede expresar como sigue:

$$M_s^i(t) = \int_{B(t)} (1 - \phi) c_s^i d\mathbf{x} \quad (4.5)$$

donde  $c_s^i$  es la concentración de masa de la componente adsorbida  $i$  en la fase sólida. A Partir de la relación entre propiedad extensiva e intensiva dada la ecuación (4.5) se pueden escribir las ecuaciones de balance global (4.6) y (4.7) respectivamente.

Ecuación de balance global

$$\frac{d}{dt} M_s^i(t) = \int_{B(t)} g_s^i(\mathbf{x}, t) d\mathbf{x} + \int_{\partial B(t)} \underline{\tau}_s^i(\mathbf{x}, t) \cdot \underline{n} d\mathbf{x} \quad (4.6)$$

ecuación de balance local

$$\frac{\partial((1 - \phi)c_s^i)}{\partial t} + \nabla \cdot ((1 - \phi)c_s^i \underline{v}_s) = g_s^i + \nabla \cdot \underline{\tau}_s^i \quad (4.7)$$

donde se considera

- $\underline{v}_s = 0$  ya que la fase sólida no se mueve.
- $\underline{\tau}_s^i = 0$  ya que no hay dispersión en la fase sólida.
- $g_s^i = (g_s^i)_{ad}$  es el término fuente, que se puede definir como la adsorción de surfactante en la fase sólida.

Queda entonces la ecuación de balance local de la masa para la componente  $i$ , en la fase sólida.

$$\frac{\partial((1 - \phi)c_s^i)}{\partial t} = (g_s^i)_{ad} \quad (4.8)$$

## 4.5. Balance total de masa de la componente $i$

Sumando las ecuaciones (4.4) y (4.8) y tomando en cuenta  $(g_f^i)_{ad} = -(g_s^i)_{ad}$ , resulta:

$$\frac{\partial(\phi S_f c_f^i + (1 - \phi)c_s^i)}{\partial t} - \nabla \cdot (\phi S_f \underline{D}_f^i \cdot \nabla c_f^i) + \nabla \cdot (\underline{u}_f c_f^i) = (g_f^i)_\Omega \quad (4.9)$$

Al desarrollar el término del tiempo de la ecuación anterior (4.9), se tiene que

$$\frac{\partial(\phi S_f c_f^i + (1 - \phi)c_s^i)}{\partial t} = (\phi S_f) \frac{\partial c_f^i}{\partial t} + \frac{\partial c_f^i}{\partial t} (\phi S_f) + (1 - \phi) \frac{\partial c_s^i}{\partial t} + \frac{\partial c_s^i}{\partial t} (1 - \phi) \quad (4.10)$$

Aquí resulta conveniente expresar el término de la concentración de la componente  $i$  en la fase sólida en función de la concentración en la fase fluida. Esto se puede hacer mediante la introducción de una isoterma de absorción  $F$ , es decir esta relación implica un equilibrio instantáneo entre la concentración en el sólido y el líquido

$$c_s^i(c_f^i) = \rho_r F(c_f^i) \quad (4.11)$$

Nótese que se tiene que introducir la densidad aparente de la roca  $\rho_r$ , a fin de que la concentración adsorbida quede referida con respecto al sitio donde el adsorbato va a quedar adherido. Entonces, al derivar la concentración de la componente  $i$  en la fase sólida respecto al tiempo resulta

$$\frac{\partial c_s^i}{\partial t} = \frac{\partial c_s^i}{\partial c_f^i} \frac{\partial c_f^i}{\partial t} \quad (4.12)$$

Al recibir la ecuación anterior mediante la función de la isoterma de adsorción, se tiene

$$\frac{\partial c_s^i}{\partial t} = \rho_r \frac{\partial F(c_f^i)}{\partial c_f^i} \frac{\partial c_f^i}{\partial t} \quad (4.13)$$

En lo sucesivo se nombrará como  $H$  a la derivada parcial de la isoterma de adsorción con respecto a la concentración de la fase fluida, es decir

$$H(c_f^i) = \frac{\partial F(c_f^i)}{\partial c_f^i} \quad (4.14)$$

Por lo que podemos expresar la ecuación (4.12) como sigue

$$\frac{\partial c_s^i}{\partial t} = \rho_r H(c_f^i) \frac{\partial c_f^i}{\partial t} \quad (4.15)$$

Finalmente la ecuación (4.9) se puede escribir de la siguiente manera, siendo esta la ecuación que vamos a trabajar para el desarrollo del trabajo.

$$\boxed{[\phi S_f + (1 - \phi)\rho_r H] \frac{\partial c_f^i}{\partial t} - \nabla \cdot (\phi S_f \underline{D}_f^i \cdot \nabla c_f^i) + \nabla \cdot (\underline{u}_f c_f^i) = (g_f^i)_\Omega - \frac{\partial c_f^i}{\partial t} (\phi S_f) - \frac{\partial F \rho_r}{\partial t} (1 - \phi)} \quad (4.16)$$

# Capítulo 5

## Modelo Numérico

Para el desarrollo del modelo numérico adoptamos y replanteamos nuestra ecuación de transporte del capítulo anterior, para la discretización en el tiempo utilizamos el método de diferencias finitas hacia atrás, se aplicará el método de elementos finitos para la solución de la ecuación.

$$[\phi S_f + (1 - \phi)\rho_r H] \frac{\partial c_f^i}{\partial t} - \nabla \cdot (\phi S_f \underline{\underline{D}}_f^i \cdot \nabla c_f^i) + \nabla \cdot (\underline{u}_f c_f^i) = (g_f^i)_\Omega - \frac{\partial c_f^i}{\partial t} (\phi S_f) - \frac{\partial F \rho_r}{\partial t} (1 - \phi)$$

Donde

$R = [\phi S_f + (1 - \phi)\rho_r H]$ , es el término de reacción,

$c = c_f^i$ , es la concentración a calcular,

$\underline{u} = \underline{u}_f$ , es el vector velocidad,

$\underline{\underline{D}} = \phi S_f \underline{\underline{D}}_f^i$ , corresponde al tensor de dispersión hidrodinámica o término difusivo,

$\nabla \cdot (\underline{u}_f c_f^i)$ , es el término advectivo,

$r = \frac{\partial(\phi S_f)}{\partial t}$ , el término de reacción,

$f = (g_f^i)_\Omega - \frac{\partial F \rho_r}{\partial t} (1 - \phi)$ , siendo este el término fuente.

Quedando finalmente la ecuación de la forma, siendo esta ecuación mas general que la anterior pues ya incluye el término de reacción:

$$R \frac{\partial c}{\partial t} - \nabla \cdot (\underline{\underline{D}} \cdot \nabla c) + \nabla \cdot (\underline{u}c) + rc = f \quad (5.1)$$



## 5.1. Planteamiento del problema general

Para el desarrollo de la formulación del modelo numérico consideremos el siguiente problema con condiciones iniciales y condiciones de frontera:

$$\mathcal{L}_{\underline{x},t}c = f, \quad \forall(\underline{x}, t) \in \Omega \quad (5.2)$$

$$c(\underline{x}, t_0) = c_0(\underline{x}), \quad \forall \underline{x} \in \Omega \quad (5.3)$$

$$c = c_\partial(\underline{x}, t), \quad \forall \underline{x} \in \partial_D \Omega \quad y \quad \forall t > t_0 \quad (5.4)$$

$$\underline{n} \cdot (\underline{D} \cdot \nabla c - \underline{u}c) = g_\partial(\underline{x}, t), \quad \forall \underline{x} \in \partial_N \Omega \quad y \quad \forall t > t_0 \quad (5.5)$$

donde  $\mathcal{L}_{\underline{x},t}c$  es un operador parabólico general de segundo orden y se expresa como:

$$\mathcal{L}_{\underline{x},t}c = R \frac{\partial c}{\partial t} - \nabla \cdot (\underline{D} \cdot \nabla c) + \nabla \cdot (\underline{u}c) + rc = f \quad (5.6)$$

donde  $\underline{D}$  es un tensor simétrico, positivo definido, acotado y suave a tramos en  $\Omega$  de la forma

$$\underline{D} = \begin{pmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{32} & D_{33} \end{pmatrix}, \quad \text{con } D_{ij} \equiv D_{ij}(\underline{x}, t), \quad \underline{u} \text{ es el vector de la forma } \underline{u} = (u_1, u_2, u_3), \quad \text{con}$$

$u_i \equiv u_i(\underline{x}, t)$ , mientras que  $c \equiv c(\underline{x}, t)$  y  $f \equiv f(\underline{x}, t)$ .  $\Omega$  es un dominio acotado con frontera  $\partial\Omega$ , a su vez la frontera está formada por dos partes, una con condiciones de tipo Dirichlet  $\partial_D\Omega$  y otra con condiciones de tipo Neumann  $\partial_N\Omega$ , donde  $\partial\Omega = \partial_D\Omega \cup \partial_N\Omega$ , con  $\partial_D\Omega \cap \partial_N\Omega = \emptyset$ .

Resulta de utilidad considerar al operador parabólico previamente definido como la suma de dos operadores de la siguiente forma:

$$\mathcal{L}_{\underline{x},t}c \equiv R \frac{\partial c}{\partial t} + \tilde{\mathcal{L}}_{\underline{x},t}c \quad (5.7)$$

donde  $R \frac{\partial c}{\partial t}$  es un operador diferencial en términos de la derivada del tiempo  $t$ , mientras que  $\tilde{\mathcal{L}}_{\underline{x},t}$  es un operador elíptico general de segundo orden y se define como

$$\tilde{\mathcal{L}}_{\underline{x},t}c \equiv -\nabla \cdot (\underline{D} \cdot \nabla c) + \nabla \cdot (\underline{u}c) + rc \quad (5.8)$$

La derivación matemática se encuentra en el apéndice E.

## 5.2. Modelo numérico ecuación transporte de una componente en un medio poroso en 2D

Como el método numérico a aplicar en este caso es el de elementos finitos, usualmente se prefiere para los casos no estacionarios realizar una semidiscretización en el tiempo de la ecuación

diferencial. Aquí se usará por simplicidad y por razones de estabilidad un esquema de diferencias finitas hacia atrás para el término del tiempo:

$$\frac{\partial}{\partial t}c^n = \frac{c^n - c^{n-1}}{\Delta t_n} + O(\Delta t_n) \quad (5.9)$$

donde  $c^n \equiv c(\underline{x}, t_n)$ ,  $\Delta t_n$  es el paso del tiempo, es decir  $\Delta t_n \equiv t_n - t_{n-1}$  con  $n = 0, 1, \dots, N$ .

Entonces, para un tiempo  $t_n$  dado podemos reescribir la ecuación como sigue:

$$R^n \frac{c^n - c^{n-1}}{\Delta t_n} = \nabla \cdot (\underline{\underline{D}}^n \cdot \nabla c^n) - \nabla \cdot (\underline{u}^n c^n) - r^n c^n + f^n \quad (5.10)$$

con  $\underline{\underline{D}}^n \equiv \underline{\underline{D}}(\underline{x}, t_n)$ ,  $\underline{u}^n \equiv \underline{u}(\underline{x}, t_n)$ ,  $c^n \equiv c(\underline{x}, t_n)$ ,  $R^n \equiv R(\underline{x}, t_n)$  y  $f^n(\underline{x}, t)$ .

Si multiplicamos ambos miembros por  $\Delta t_n$  resulta:

$$R^n(c^n - c^{n-1}) = \Delta t_n(\nabla \cdot (\underline{\underline{D}}^n \cdot \nabla c^n) - \nabla \cdot (\underline{u}^n c^n) - r^n c^n + f^n) \quad (5.11)$$

$$R^n c^n - R^n c^{n-1} = \Delta t_n \nabla \cdot (\underline{\underline{D}}^n \cdot \nabla c^n) - \Delta t_n \nabla \cdot (\underline{u}^n c^n) - \Delta t_n r^n c^n + \Delta t_n f^n \quad (5.12)$$

Reacomodando términos tenemos:

$$R^n c^n - \Delta t_n \nabla \cdot (\underline{\underline{D}}^n \cdot \nabla c^n) + \Delta t_n \nabla \cdot (\underline{u}^n c^n) + \Delta t_n r^n c^n = R^n c^{n-1} + \Delta t_n f^n \quad (5.13)$$

Si multiplicamos por una función de peso  $w$  e integramos en  $\Omega$  resulta:

$$\int_{\Omega} (R^n c^n - \Delta t_n \nabla \cdot (\underline{\underline{D}}^n \cdot \nabla c^n - \underline{u}^n c^n + \Delta t_n r^n c^n) w d\underline{x} = \int_{\Omega} (R^n c^{n-1} + \Delta t_n f^n) w d\underline{x} \quad (5.14)$$

de aquí:

$$\begin{aligned} \int_{\Omega} R^n c^n w d\underline{x} - \Delta t_n \int_{\Omega} \nabla \cdot (\underline{\underline{D}}^n \cdot \nabla c^n - \underline{u}^n c^n) w d\underline{x} \\ \Delta t_n \int_{\Omega} r^n c^n w d\underline{x} = \int_{\Omega} (R^n c^{n-1} + \Delta t_n f^n) w d\underline{x} \end{aligned} \quad (5.15)$$

Aplicando el teorema de la divergencia obtenemos:

$$\begin{aligned} - \int_{\Omega} \nabla \cdot (\underline{\underline{D}}^n \cdot \nabla c^n - \underline{u}^n c^n) w d\underline{x} = \int_{\Omega} \nabla w \cdot (\underline{\underline{D}}^n \cdot \nabla c^n - \underline{u}^n c^n) d\underline{x} \\ - \int_{\partial\Omega} w (\underline{\underline{D}}^n \cdot \nabla c^n - \underline{u}^n c^n) \cdot \underline{n} d\underline{x} \end{aligned} \quad (5.16)$$

Aplicando las condiciones de frontera obtenemos:

$$\begin{aligned} \int_{\partial\Omega} w(\underline{D}^n \cdot \nabla c^n - \underline{u}^n c^n) \cdot \underline{n} d\underline{x} &= \int_{\partial_D\Omega} w(\underline{D}^n \cdot \nabla c^n - \underline{u}^n c^n) \cdot \underline{n} d\underline{x} \\ &+ \int_{\partial_N\Omega} w(\underline{D}^n \cdot \nabla c^n - \underline{u}^n c^n) \cdot \underline{n} d\underline{x} \end{aligned} \quad (5.17)$$

$$\int_{\partial\Omega} w(\underline{D}^n \cdot \nabla c^n - \underline{u}^n c^n) \cdot \underline{n} d\underline{x} = \int_{\partial_N\Omega} w g_{\partial}^n d\underline{x} \quad (5.18)$$

La formulación variacional resulta de la siguiente manera:

$$\begin{aligned} \int_{\Omega} R^n c^n w d\underline{x} + \Delta t_n \int_{\Omega} \nabla w \cdot (\underline{D}^n \cdot \nabla c^n - \underline{u}^n c^n) d\underline{x} - \Delta t_n \int_{\partial_N\Omega} w g_{\partial}^n d\underline{x} \\ + \Delta t_n \int_{\Omega} r^n c^n w d\underline{x} = \int_{\Omega} (R^n c^{n-1} + \Delta t_n f^n) w d\underline{x} \end{aligned} \quad (5.19)$$

El resultado de la forma débil o variacional es conveniente reescribirlo en su notación estándar

$$a(c, w) = L(w) \quad (5.20)$$

donde  $a(c, w)$  se le conoce como forma bilineal y  $L(w)$  se conoce como forma lineal en la literatura matemática. De aquí podemos escribir la siguiente notación en forma bilineal y lineal:

$$a(c, w) = \int_{\Omega} R^n c^n w d\underline{x} + \Delta t_n \int_{\Omega} \nabla w \cdot (\underline{D}^n \cdot \nabla c^n - \underline{u}^n c^n) d\underline{x} + \Delta t_n \int_{\Omega} r^n c^n w d\underline{x} \quad (5.21)$$

y

$$L(w) = \int_{\Omega} (R^n c^{n-1} + \Delta t_n f^n) w d\underline{x} + \Delta t_n \int_{\partial_N\Omega} w g_{\partial}^n d\underline{s} \quad (5.22)$$

La formulación variacional discreta para el método de elementos finitos, se expresa como:  
Encontrar  $u_h \in V_h \subset V$  tal que

$$a(u_h, w_h) = L(w_h) \quad \forall w_h \in \hat{V}_h \subset \hat{V} \quad (5.23)$$

donde

$V = \{v \in H^1(\Omega) : v = u_{\partial} \text{ en } \partial_D\Omega\}$  es el espacio de las funciones de prueba,  
 $\hat{V} = \{v \in H^1(\Omega) : v = 0 \text{ en } \partial_D\Omega\}$  es el espacio de las funciones de base y  
 $H^1$  es el espacios de Sobolev.

# Capítulo 6

## Modelo Computacional

La implementación computacional de este trabajo se realizó en el proyecto FEniCS 2016.1 que es una biblioteca de código abierto bajo la plataforma de Python 2.7, se usó la el software editor interprete de Spyder (Scientific Python Development Environment), (Entorno de Desarrollo Científico de Python). El trabajo se fue realizado en el sistema operativo *Ubuntu Linux*, en una computadora *Acer* de 64 bits, con una memoria RAM de 4 GB y 1 GB de DD.

Vamos a mostrar el modelo computacional considerando las hipótesis generales, discretización de la malla y del dominio, condiciones de frontera y su implementación en un código computacional.

### 6.1. Bibliotecas utilizadas

Vamos mostrar las bibliotecas más utilizadas en la parte computacional del trabajo y que además son necesarias para poder implementar la solución computacional.

```
from fenics import *
from mshr import *
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

### 6.2. Geometría

#### 6.2.1. Dominio

El dominio de definición de nuestro modelo computacional es un rectángulo en posición vertical que equivale, una sección en 2D de un cilindro en 3D, con longitud  $l$  y ancho  $d$ , siendo esté en la implementación computacional lo tomamos de la siguiente manera,  $\Omega = [x_{min}, y_{min}] \times [x_{max}, y_{max}]$ , quedando en el código de la siguiente manera:

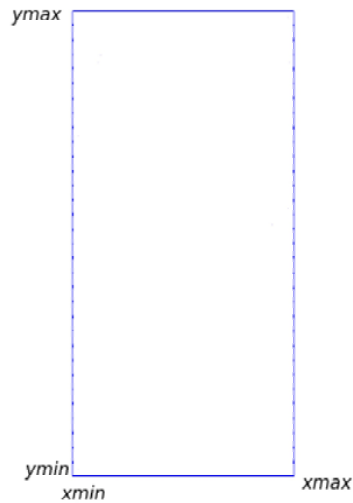


Figura 6.1: Ejemplo del dominio de definición del modelo computacional en 2D para nuestra ecuación, donde  $l = 0.08$  m y  $d = 0.038$  m.

```

l=0.08 # core length L=0.08 [m]
d=0.038 # core diameter d=0.038 [m]
r=d/2 # core radius
xmin=0.
xmax=d
ymin=0.
ymax=l
domain = Rectangle(dolfin.Point(xmin, ymin), dolfin.Point(xmax, ymax))

```

## 6.2.2. Mallado

A continuación se muestra el mallado del dominio:

```

# Creating a mesh generator object gives access to parameters of the
generator = CSGCGALMeshGenerator2D()
generator.parameters["cell_size"] = 0.25
generator.parameters["mesh_resolution"] = 30.0
generator.parameters["edge_truncate_tolerance"] = -1.0
generator.parameters["triangle_shape_bound"] = 0.125
# Generate and plot mesh
mesh = generate_mesh(domain, 64)
plot(mesh, "2D mesh")
interactive()

```

### 6.2.3. Espacios de funciones:

```
fs_order = 1 # function space order
V = FunctionSpace(mesh, 'Lagrange', fs_order)
```

Para crear los espacios de funciones necesitamos la malla (*mesh*), los espacios de polinomios que en éste caso estamos utilizando polinomios de Lagrange y el grado del polinomio, que para nuestro trabajo utilizamos el polinomio de grado 1.

### 6.2.4. Fronteras

Mostraremos la figura con sus respectivas condiciones de frontera que se utilizan en el problema. Las condiciones de frontera se colocaron en base a la numeración como se muestra en la figura (6.2) y se utilizarán de la siguiente manera:

1. Para la frontera número (1) se utilizó una condición de frontera tipo Dirichlet, con una inyección de  $c = \text{constante}$ .
2. Para la frontera (2) y (4) se utilizará condiciones de frontera tipo Neumann, ya que son fronteras de no flujo es decir  $\underline{n} \cdot (\underline{D} \cdot \nabla u - \underline{b}u) = 0$ .
3. Para la frontera (3) también utilizaremos una condición tipo Neumann donde la derivada normal sea igual a cero, ya que esa frontera será la salida del flujo  $\frac{\partial c}{\partial n} = 0$ . Aquí el flujo dispersivo es cero, pues ya no hay medio poroso en el exterior.

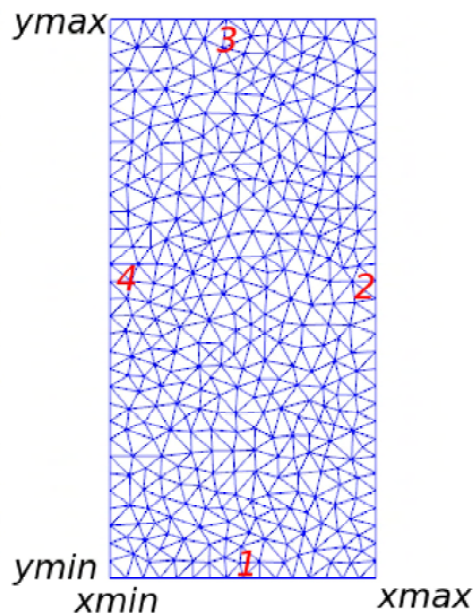


Figura 6.2: Condiciones de frontera

La cual su implementación computacional se ve como sigue:

```
#Boundary condition at the bottom
class BottomBoundary(SubDomain):
    def inside(self, x, on_boundary):
        tol = 1E-14 # tolerance for coordinate comparisons
        return on_boundary and abs(x[1] - ymin) < tol

Gamma_1 = BottomBoundary()
Gamma_1.mark(boundary_parts, 1)

#If Dirichlet condition
c_B = Constant(c_in)
bc1=DirichletBC(V, c_B, boundary_parts, 1)

#Boundary condition at the right
class RightBoundary(SubDomain):
    def inside(self, x, on_boundary):
        tol = 1E-14 # tolerance for coordinate comparisons
        return on_boundary and abs(x[0] - xmax) < tol
Gamma_2 = RightBoundary()
Gamma_2.mark(boundary_parts, 2)

#Boundary condition at the top
class TopBoundary(SubDomain):
    def inside(self, x, on_boundary):
        tol = 1E-14 # tolerance for coordinate comparisons
        return on_boundary and abs(x[1] - ymax) < tol

Gamma_3 = TopBoundary()
Gamma_3.mark(boundary_parts, 3)
#Boundary condition at the left
class LeftBoundary(SubDomain):
    def inside(self, x, on_boundary):
        tol = 1E-14 # tolerance for coordinate comparisons
        return on_boundary and abs(x[0] - xmin) < tol

Gamma_4 = LeftBoundary()
Gamma_4.mark(boundary_parts, 4)
bcs = [bc1]
```

### 6.2.5. Condiciones iniciales

La implementación de las condiciones iniciales se muestran a continuación:

```
c_0 = Expression('c0 + t',c0=c0,t=t_0, degree=1)
c_1 = project(c_0, V)
```

### 6.2.6. La formulación variacional

```
# Variational forms and problem
a = R*c*w*dx + dt*inner(nabla_grad(w), dot(D,nabla_grad(c))-u*c)*dx \
    + dt*u_y*c*w*ds(3)
L = (R*c_1 + dt*q_c)*w*dx #- dt*g_in*w*ds(1)
```

Donde  $a$  es la forma bilineal y  $L$  es la forma lineal de la discretización variacional con elementos finitos de la ecuación de transporte.

### 6.2.7. Gráficas de la solución

```
xlab='y-coordinate [m]'
ylab='C [ppm]'
nt=NT
nx=NY
x_section=y_section
y_time=cy_time
x_sec_min=ymin
x_sec_max=ymax
fig_title='conc vs time'
legend=False
XYplot_profiles(nt, time, y_time, x_sec_min, x_sec_max,
nx, xlab, ylab, fig_title, legend)

xlab='Time [s]'
ylab='C [ppm]'
xmin=t_0
xmax=t_N
symbol_color='r-'
XYplot(time, c_t_in, xmin, xmax, xlab, ylab, symbol_color)
XYplot(time, c_t_out, xmin, xmax, xlab, ylab, symbol_color)
```

El código completo se puede ver el en apéndice F.





## Validación del Modelo

### 7.1. Problema de inyección tipo escalón

Para la validación se utilizó el problema de inyección de un escalón, se tomaron los datos del artículo [Van Genuchten y Alves \(1982\)](#) quienes listan una serie de soluciones a la ecuación unidimensional de transporte convectivo-dispersivo de solutos, la cual se tomo del articulo de la pag. 11 siendo la ecuación A3. y la comparación con las gráficas del código hecho con el software proyecto FEniCS 2016.1 bajo la plataforma de Python 2.7. Este trabajo se realizó con la ecuación de transporte en una componente en un medio poroso en 2D. Con condiciones de frontera tipo Neumann, y una condición de frontera tipo Dirichlet (inyección constante) y un dominio en  $\Omega=[x_{min},y_{min}]x[x_{max}, y_{max}]$ . Se utilizó la discretización con el método de elemento finito para pasar la ecuación a su forma variacional, de donde se obtuvo la forma bilineal y lineal, para la discretización en el tiempo se utilizó el método de diferencias finitas hacia atrás por ser el más estable. Las características de la malla que se utilizó es un rectángulo con las siguientes dimensiones tienen una longitud de  $l = 0.08$ , un ancho de  $d = 0.038$ .

Datos utilizados en el problema:

CO =	1326.2	mmol/L	Concentración inicial
Ciny =	13.26	mmol/L	Concentración de inyección
L=	0.08	m	Longitud del núcleo
VO=	$2.58E - 7$	m/s	velocidad
D=	$2.0648E - 7$	$m^2/s$	Coefficiente de difusión
tp=	500000	s	Duración periodo de inyección

Concentración evaluada en  $z = 0.08m$   $t \in [0 - 400000] s$ .

## Ecuación de inyección tipo escalón

La validación se realizó con la ecuación de transporte en una componente en 2D.

**Ecuación de transporte de inyección tipo escalón:**

$$\frac{\partial}{\partial t}(c_w^t) + \nabla \cdot \{c_w^t \underline{u}_w - D_w^t \nabla c_w^t\} = 0 \quad (7.1)$$

1. **Condición inicial:**

$$c_w^t(x, t_0) = 0 \quad (7.2)$$

2. **Frontera de entrada (1):**

$$c_{w_{iny}}^t(x, 0) = \begin{cases} c_{w_{iny}} & \text{para } t_0 < t < t_{iny} \\ 0 & \text{para } t > t_{iny} \end{cases}$$

3. **Frontera de salida (3):**

$$\frac{\partial c_w^t}{\partial x}(L, t_0) = 0 \quad (7.3)$$

donde L es el tamaño de la columna.

4. **Condiciones sin flujo para las otras fronteras (2) y (4).**

A continuación mostramos las gráficas obtenidas en el artículo de Van Genuchten y W. J. Alves, para la ecuación de transporte tipo escalón. Siendo el resultado de la solución analítica la siguiente:

$$A(x, t) = 1 - \sum_{m=1}^{\infty} \frac{2\beta_m \sin(\frac{\beta_m x}{L}) \exp[\frac{v x}{2D} - \frac{v^2 t}{4DR} - \frac{\beta_m^2 D t}{L^2 R}]}{[\beta_m^2 + (\frac{vL}{2D})^2 + \frac{vL}{2D}]} \quad (7.4)$$

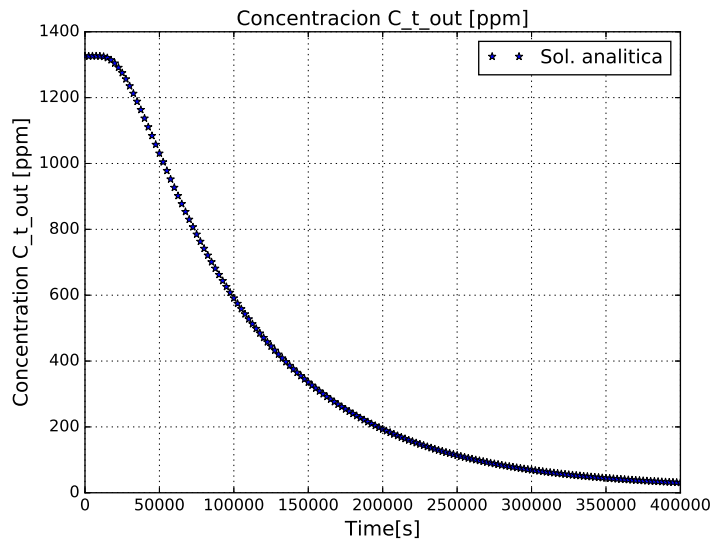


Figura 7.1: Gráfica de la solución analítica de la concentración contra el tiempo en el problema de inyección tipo escalón a la salida.

## 7.2. Implementación computacional

Una vez que se ha establecido el procedimiento numérico para la solución del modelo matemático, se requiere su implementación computacional.

### Geometría de la malla.

Ahora podemos ver los resultados obtenidos con el software Proyecto FEniCS 2016.1 en el lenguaje de programación Python.

### Dominio

El dominio del modelo computacional es un rectángulo en posición vertical con longitud  $l$  y ancho  $d$ , donde  $\Omega = [x_{min}, y_{min}] \times [x_{max}, y_{max}]$  ver figura (a).

### Mallado

El mallado del dominio utilizado en el modelo computacional ver figura (b):

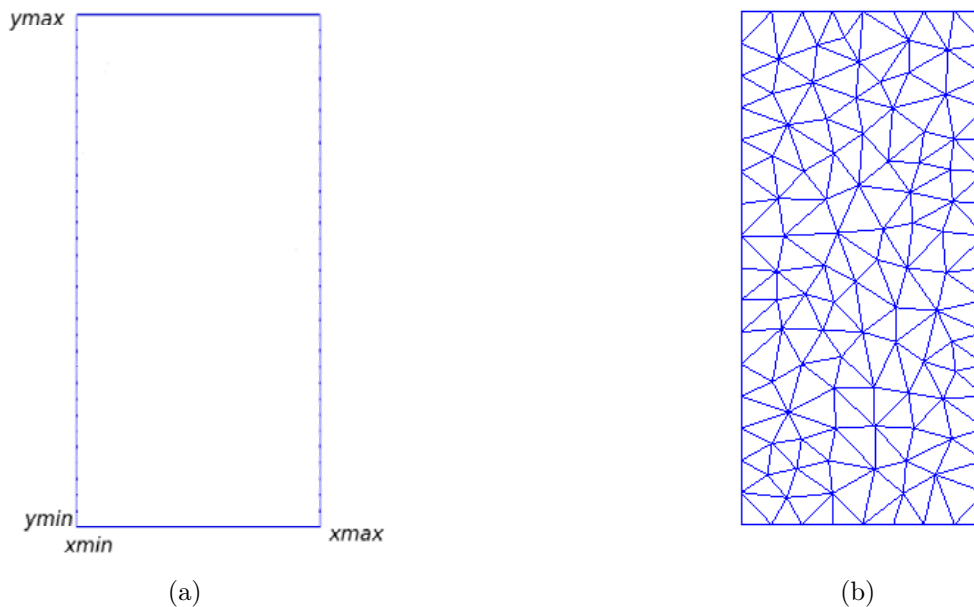


Figura 7.2: (a) Ejemplo del dominio de definición del modelo computacional en 2D para nuestra ecuación, donde  $l = 0.08$  m y  $d = 0.038$  m.

(b) Mallado en 2D con triángulos, la malla se compone con 440 vértices y 720 celdas ordenadas.

## Condiciones de Fronteras

Mostraremos la figura con sus respectivas condiciones de frontera que se utilizan en el problema. Las condiciones de frontera se colocaron en base a la numeración dada en la figura (7.3), que se utilizarán de la siguiente manera:

1. Para la frontera número (1) se utilizó una condición de frontera tipo Dirichlet, con una inyección de  $c = \text{constante}$ .
2. Para la frontera (2) y (4) se utilizará condiciones de frontera tipo Neumann, ya que son fronteras de no flujo es decir  $\underline{n} \cdot (\underline{D} \cdot \nabla u - \underline{b}u) = 0$ .
3. Para la frontera (3) también utilizaremos una condición tipo Neumann donde la derivada normal sea igual a cero, ya que esa frontera será la salida del flujo  $\frac{\partial c}{\partial n} = 0$ . Aquí el flujo dispersivo es cero, pues ya no hay medio poroso en el exterior.

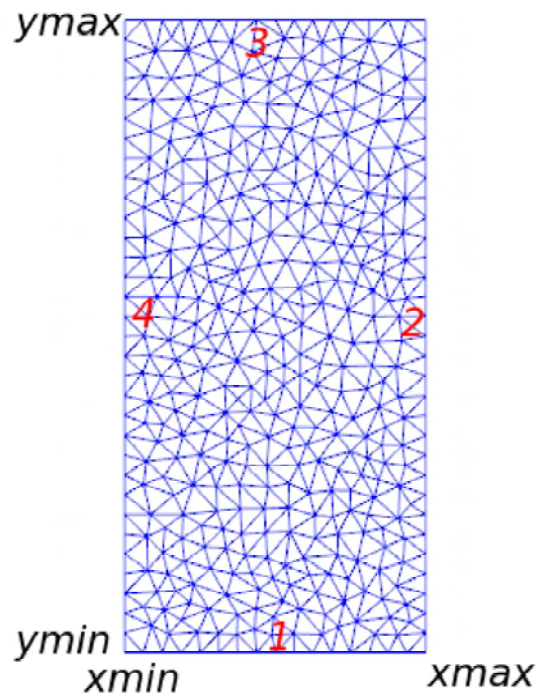


Figura 7.3: Condiciones de frontera

Datos del problema.

Mostraremos los datos utilizados en el programa:

Datos utilizados en el problema:	
Término de adsorción, $R =$	1
Velocidad de inyección, $u\text{-in} =$	$2.5810e - 07$ [m/s]
Velocidad de Darcy en la dirección x, $u\text{-x} =$	0 [m/s]
Velocidad de Darcy en la dirección y, $u\text{-y} =$	$2.5810e - 07$ [m/s]
Coefficiente de dispersión longitudinal, $\alpha\text{-L} =$	0.08 [m]
Coefficiente de dispersión transversal, $\alpha\text{-T} =$	0 [m]
$D_{xx} =$	$2.0648e - 08$ [m <sup>2</sup> /s]
Término de la fuente, $q\text{-c} =$	0 [ppm/s]
Concentración inicial, $c_0 =$	1326.2 [ppm]
Concentración inyectada, $c\text{-in} =$	13.262 [ppm]

Información del Tiempo:		
Tiempo inicial,	$t\text{-0} =$	0[s]
Tiempo final,	$t\text{-N} =$	400000 [s]
Paso de tiempo,	$dt =$	500[s]
Número de pasos de tiempo,	$NT =$	160

Ahora mostraremos los resultados de graficar con el programa del Proyecto FEniCS 2016.1.

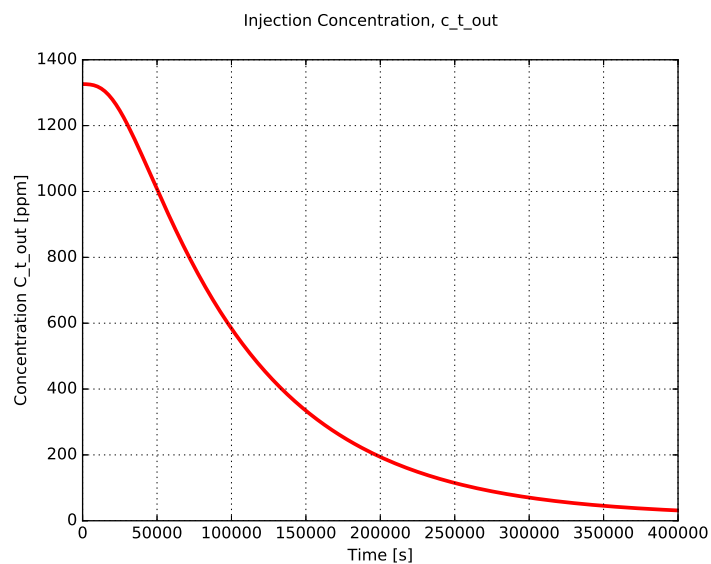


Figura 7.4: Gráfica de la concentración contra el tiempo del problema inyección tipo escalón realizado con el código en el Proyecto FEniCS 2016.1:

Graficando los datos del problema de Van Genuchten y W. J. Alves y las soluciones obtenidas con el proyecto FEniCS 2016.1 podemos ver que las gráficas son similares.

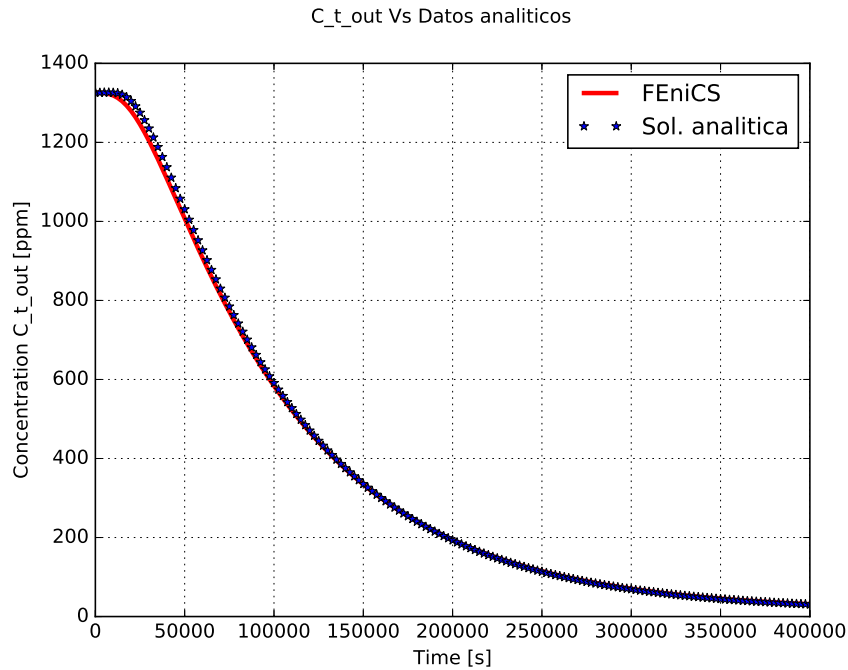


Figura 7.5: Gráfica de las comparaciones a las salidas de las concentraciones, la línea roja representa los datos en proyecto FEniCS 2016.1, la línea de estrellas los datos en COMSOL *Multiphysics*<sup>®</sup>.

**Discusión de resultados:** Observando la figura (7.5) vemos que en cada caso donde se obtuvieron las concentraciones a la salida tanto en la simulación con el proyecto FEniCS línea roja continua, como con la obtenida con la solución analítica denotadas por las estrellas, se puede ver que los resultados obtenidos en el proyecto FEniCS, con los expuestos en el artículo muestran mucha congruencia, ya que las gráficas de la concentración prácticamente una esta encima de la otra. El error relativo que se obtuvo en la realización de la validación del problema fue de  $3.8729e-05$ , cualitativamente son iguales y con el error que se obtuvo podemos decir que la validación del problema es satisfactorio.

## 7.3. Número de Péclet

El número de Péclet es de gran importancia en la solución numérica de ecuaciones diferenciales parciales en problemas de advección-difusión, en donde existen dificultades cuando se presenta casos de advección dominante sobre la difusión .

Este número adimensional lo vamos abordar de dos formas de manera global y local:

- **Número de Péclet global:** Es el análisis del dominio (longitud total) donde se encuentra definida la ecuación.

*Se define como:*

$$\mathbb{P}_{eG} = \frac{|u|}{D} L \quad (7.5)$$

*donde  $L$  es el tamaño del dominio.*

El número de Péclet mide la dominancia del término advectivo  $u$  sobre el difusivo  $D$ .

- **Número de Péclet local.** Analiza los subintervalos de la partición del dominio.

*Se define como:*

$$\mathbb{P}_{eL} = \frac{|u|}{D} h \quad (7.6)$$

*donde  $h$  es el tamaño del subintervalo de la partición del dominio.*

### 7.3.1. Análisis numérico de la difusión usando el Número de Péclet

El número de Péclet se define a partir del término advectivo, difusivo y longitud del intervalo a analizar. La relevancia del número de Péclet, es que a través de él, se puede hacer un análisis de la relación que existe entre los términos advectivo y difusivo. Para analizar la solución obtenida con el método de elemento finito, y la solución obtenida de manera analítica de la ecuación diferencial parcial. Los métodos numéricos para hallar la solución de la ecuación se basan fundamentalmente en discretizar el dominio, y es aquí donde el número de Péclet local está involucrado, es decir al obtener la solución de nuestro problema de interés; nuestro mayor interés está enfocado en saber cuál es la mejor aproximación numérica a la solución analítica, y con la relación que existe entre los métodos numéricos y el número de Péclet [Rosas-Medina y Herrera \(2005\)](#).



## Gráficas con diferentes Número de Péclet Local.

Casos con el mismo el mismo número de Péclet global  $P_{eG} = 100$

- a)  $P_{eL} = 21.5$   $h_{max} = 0.07$
- b)  $P_{eL} = 11$   $h_{max} = 0.009$
- c)  $P_{eL} = 5.5$   $h_{max} = 0.004$
- d)  $P_{eL} = 3.7$   $h_{max} = 0.001$

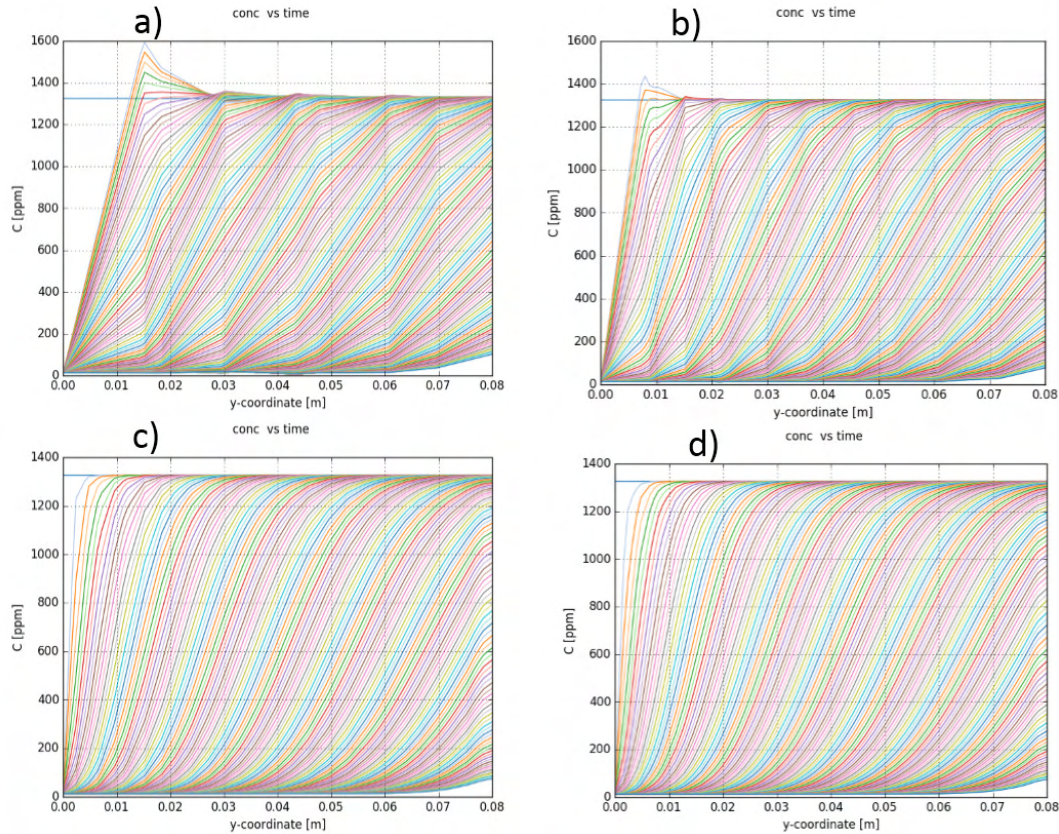


Figura 7.6: Gráficas con diferentes números de Péclet

Dado un número de Péclet global y modificando el número de Péclet local y refinando la malla para diferentes casos como podemos observar en las figura (7.6). La figura del inciso (a) muestra oscilaciones espurias de la solución que propaga a diferentes velocidades de la velocidad real mostrando el salto de la concentración en la gráfica, disminuyendo el número de Péclet y refinando la malla para el inciso (b) las oscilaciones en la solución va disminuyendo y así sucesivamente como vamos disminuyendo el número de Péclet y refinando la malla la solución se va suavizando lo cual podemos ver en el inciso (d). Este fenómeno se da cuando tenemos frentes abruptos o gradientes muy pronunciados. En resumen al disminuir el número de Péclet y disminuir el  $\Delta x$  lo que hacemos es hacer que la velocidad se mantenga lo mas suave posible.

## Caso de Estudio I

### 8.1. Inyección de un pulso de trazador inerte.

Se realiza experimento de inyección de un trazador inerte para caracterizar el coeficiente de dispersión asociado con el medio poroso de interés. Dada la dependencia de este coeficiente con la velocidad del fluido, se recomienda realizar varias pruebas a diferentes tasas de inyección [Díaz-Viera y Moctezuma-Berthier \(2012\)](#).

### 8.2. Supuestos del caso de estudio

Realizamos experimentos para investigar las propiedades de dispersión de uno de los medios porosos de nuestro interés arenisca Berea (ver Apéndice A). Empleamos un núcleo de  $14.18\text{ cm}$  de largo, con un diámetro interno de  $10.16\text{ cm}$ , porosidad de  $0.19575$ . El núcleo se coloca verticalmente y se inyecta por el extremo inferior agua hasta asegurar una saturación total de agua al antes de inyectar el trazador. Se elige una condición de frontera tipo pulso para el experimento. El pulso se obtiene reemplazando el flujo de agua por el trazador (una solución de cloruro de sodio a  $1\text{ g/L}$ , equivalente a  $617\text{ mg/L}$  de ión cloruro) durante  $3.16$  horas. La tasa de inyección del trazador fue de  $100\text{ ml/h}$ . Se toman muestras cada  $6$  minutos del efluente. Los datos Utilizados los podemos ver en la tabla (8.1).

Propiedad (Símbolo)	Valor [Unidad]
Porosidad ( $\phi$ )	0.1957
Tasa de inyección ( $u_w$ )	$1.7512e - 5\text{ [m/s]}$
Coefficiente de dispersión del cloruro ( $D_{cl}$ )	$3e - 6\text{ [m}^2\text{/s]}$
Longitud del núcleo ( $x_L$ )	$0.1418\text{ [m]}$

Tabla 8.1: Parámetros para el problema de inyección NaCl de un trazador a través de un núcleo de arenisca Berea.

### 8.3. Modelo matemático

Para modelar las curvas efluentes del trazador inerte, la ecuación de transporte que se utilizó fue:

*Ecuación de transporte de trazador inerte:*

$$\frac{\partial}{\partial t}(\phi c_w^t) + \nabla \cdot \{c_w^t \underline{u}_w - \phi D_w^t \nabla c_w^t\} = 0 \quad (8.1)$$

*Condición inicial:*

$$c_w^t(t_0) = 0 \quad (8.2)$$

*Condiciones de frontera de entrada (1):*

$$-D_w^t \frac{\partial c_w^t}{\partial x} + u_w c_w^t = u_w c_{w_{iny}}^t \quad (8.3)$$

donde

$$c_{w_{iny}}^t = \begin{cases} c_{w_{iny}} & \text{para } t_0 < t < t_{iny} \\ 0 & \text{para } t > t_{iny} \end{cases}$$

es decir vale 617 mg/L durante el tiempo de inyección y luego se anula.

*Condiciones de frontera de salida (3):*

$$\frac{\partial c_w^t}{\partial x} = 0 \quad (8.4)$$

*Condiciones sin flujo para las otras fronteras (2) y (4).*

Como el medio poroso está completamente saturado con agua, por lo que la  $S_w = 1$ , además se supone que el medio poroso es isotrópico y homogéneo, por lo que el tensor de dispersión hidrodinámico se simplifica a una constante  $-\underline{\underline{D}}_w^t = -D_w^t \underline{\underline{I}}$ .

### 8.4. Implementación computacional

Una vez que se ha establecido el procedimiento numérico para la solución del modelo matemático, se requiere su implementación computacional.

#### 8.4.1. Geometría de la malla

La implementación computacional del modelo de transporte también se llevó a cabo en software proyecto FEniCS 2016.1.

#### Dominio

El dominio del modelo computacional en 2D es un rectángulo en posición vertical con las siguientes dimensiones:

Longitud  $l = 0.1418\text{m}$

ancho  $d = 0.1016\text{m}$

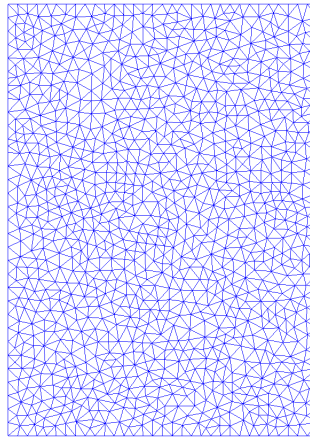


Figura 8.1: La discretización de la malla para la solución numérica, está formada por 5031 vértices 9796 elementos triangulares. Peclet Local máximo = 0.0080, hmax = 0.0027

## Mallado

### 8.4.2. Condiciones de fronteras

Mostraremos la figura con sus respectivas condiciones de frontera que se utilizan en el problema. Las condiciones de frontera se colocaron en base a la numeración dada en la (figura 7.3) que se utilizarán de la siguiente manera:

1. Para la frontera número (1) se utilizó una condición de frontera tipo Dirichlet, con una inyección de  $c = \text{constante}$ .
2. Para la frontera (2) y (4) se utilizará condiciones de frontera tipo Neumann, ya que son fronteras de no flujo es decir  $\underline{n} \cdot (\underline{D} \cdot \nabla u - \underline{bu}) = 0$ .
3. Para la frontera (3) también utilizaremos una condición tipo Neumann donde la derivada normal sea igual a cero, ya que esa frontera será la salida del flujo  $\frac{\partial c}{\partial n} = 0$ . Aquí el flujo dispersivo es cero, pues ya no hay medio poroso en el exterior.

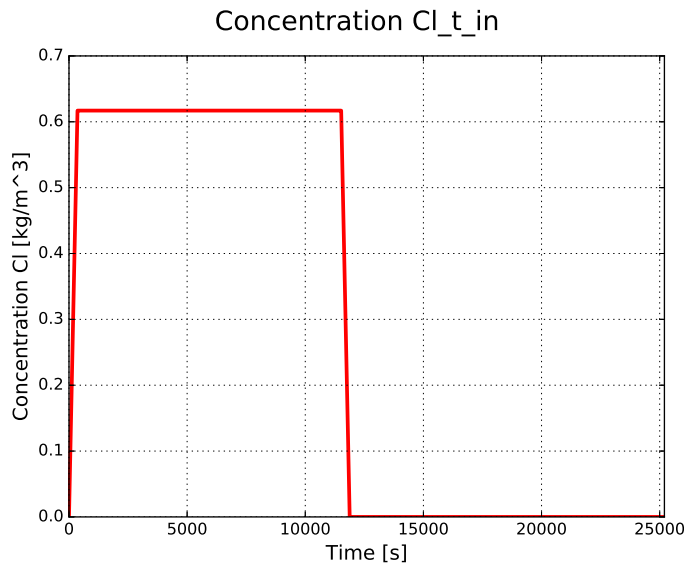


Figura 8.2: Concentración de inyección de entrada en el experimento.

La figura (8.2) se muestra la concentración en la entrada del núcleo del experimento del pulso de inyección tipo escalón, hasta alcanzar su punto máximo, luego se mantiene estacionario hasta llegar a las 2.7 horas cae hasta cero y se mantiene estacionario.

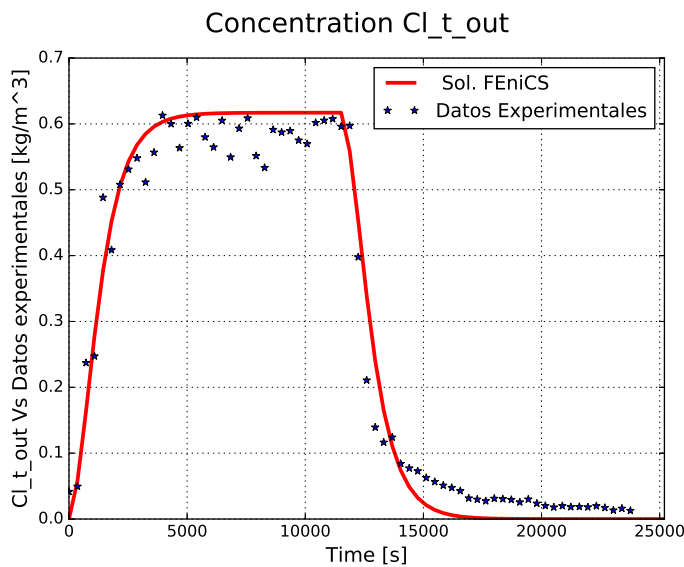


Figura 8.3: Ajuste de la solución numérica con el experimento. Con un error máximo =  $3.8729e - 05$

En la figura (8.3) se muestra la concentración a la salida del núcleo del experimento del pulso de inyección tipo escalón, podemos observar los datos del experimento comparados con la solución numérica, los datos del experimento se muestran con las estrellas de color azul, y los resultados de la simulación se muestran con la línea

roja continua, aquí se muestra un ajuste manual tomando en consideración un valor del coeficiente de dispersión de  $D_w^t = 3 \text{ m}^2/\text{s}$ , la idea fundamental es el ajuste de la concentración máxima y los valores de ajuste alrededor del pulso los iniciales y los de la caída, no así el comportamiento final de la cola, El objetivo no es buscar el mejor ajuste, si no que la solución numérica cualitativamente reproduce el comportamiento de los datos experimentales.

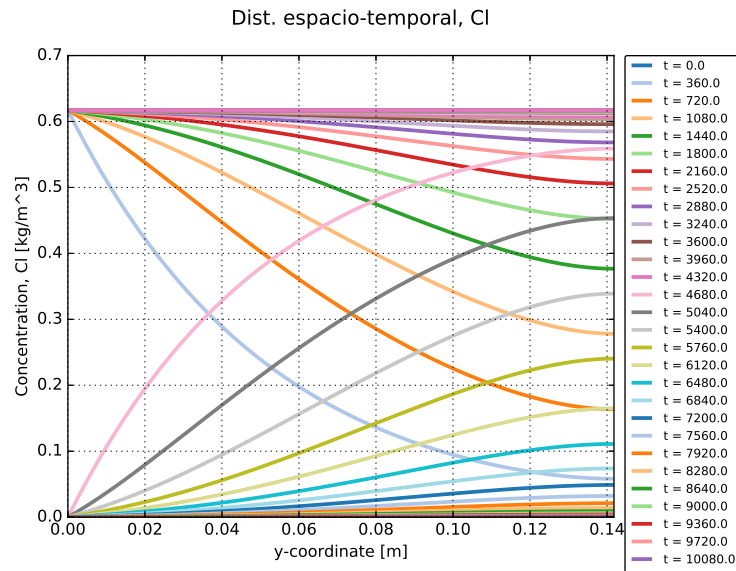


Figura 8.4: Distribución espacio-temporal del ión  $Cl^-$  a lo largo del núcleo.

Adicionalmente, en la figura (8.4) se puede observar la evolución de la concentración a lo largo del núcleo en el espacio y en el tiempo de las curvas de concentración de la sal de cloruro de sodio para algunos tiempos seleccionados (en segundos), se puede ver como en la curvas de la concentración inicia con la que fue inyectada en el tiempo inicial y va decayendo con el paso del tiempo y cuando se produce el tiempo del salto, vemos como decae hasta que se vuelve estacionario, el espaciamiento de cada efluente es de 6 segundo en un tiempo total de 7 horas.

### 8.4.3. Discusión de resultados

En este experimento se analiza el transporte de un trazador inerte a través de un núcleo de arenisca Berea. En la tabla (8.1) se muestran los datos medidos de la concentración de ión cloruro de sodio a la salida del sistema poroso, en función del tiempo. Así como el resultado de la simulación numérica correspondiente, ajustando un valor de  $D_w^t = 3 \text{ m}^2/\text{s}$  para el coeficiente de dispersión hidrodinámica a estas condiciones del experimento, se muestran tres figuras que se realizaron con el proyecto FeniCS en la simulación del experimento, la figura (8.2) se muestra inyección inicial de la concentración al inicio del núcleo, en la figura (8.3) podemos ver la concentración de salida al final del núcleo y también el ajuste con los datos experimentales, por ultimo la figura (8.4) muestra la curvas de concentración del efluente del ión de cloruro del caso de estudio tipo escalón.



## Caso de Estudio II

Se presenta un modelo transporte que se implementó utilizando el método de elementos finitos en el entorno proyecto FEniCS 2016.1, para simular, analizar e interpretar numéricamente los procesos de transporte multi-componente en medios porosos a escala de núcleo. El modelo de transporte se realizó en una sola fase (agua) y tres componentes (organismos planctónicos, microorganismos sésiles y nutrientes). Este modelo incluye fenómenos físico-químico-biológicos tales como advección, difusión, dispersión, adsorción, desorción, crecimiento y decaimiento de microorganismos.

### 9.1. Proceso de taponamiento/destaponamiento por microorganismos

En este caso de estudio se inyectan de manera simultánea y continua microorganismos y nutrientes a un núcleo de arenisca de Berea. A través de la fase agua durante 24 horas hasta que se alcanza el estado estacionario. El objetivo es evaluar el efecto de la porosidad y la permeabilidad debido al proceso de taponamiento/destaponamiento producido por el crecimiento microbiano. Los datos utilizados en el experimento son tomados [Díaz-Viera y Moctezuma-Berthier \(2012\)](#).

### 9.2. Supuestos generales del modelo:

1. Existen dos fases: una fase de roca sólida ( $s$ ) y una fase de agua fluida ( $w$ )
2. Hay cuatro componentes: agua ( $w$ ) sólo en la fase agua, microorganismos ( $m$ ) repartidos entre la fase agua (planctónicos) y la fase biopelícula (sésiles), y nutrientes ( $n$ ) también repartidos entre la fase agua (fluyentes).
3. El medio poroso es isotrópico y está completamente saturado de agua ( $V_w = V_{poroso}$ ) tal que,  $S_w = 1$ , ya que

$$\phi \equiv \frac{V_{poroso}}{V_{total}} \quad (9.1)$$

es la porosidad del medio (fracción de volumen poroso en el volumen total) y

$$S_w \equiv \frac{V_w}{V_{poroso}} \quad (9.2)$$

es la saturación del agua en el medio (fracción de volumen ocupado por el agua en el volumen poroso).



4. Se considera que el flujo de los microorganismos y nutrientes siguen la ley de Fick:

$$\tau^\gamma = \phi S_w \underline{D}_w^\gamma \cdot \nabla c_w^\gamma, \quad \gamma = m, n, \text{ donde}$$

$$(D_w^\gamma)_{ij} = (\alpha_T)^\gamma |\underline{u}| \delta_{ij} + ((\alpha_L)^\gamma - (\alpha_T)^\gamma) \frac{u_{wi} u_{wj}}{|\underline{u}_w|} + \tau (D_m)^\gamma \delta_{ij} \quad (9.3)$$

donde la ecuación (9.3) es el tensor de dispersión hidrodinámica,  $(\alpha)^\gamma_w$  y  $(\alpha)^\gamma_w$  son los coeficientes de dispersividad mecánica transversal y longitudinal, respectivamente  $\delta_{ij}$  es la función delta de Kronecker,  $\tau < 1$  es la tortuosidad y  $(D_m)^\gamma_w$  es la difusión molecular.

5. La roca (matriz porosa) y los fluidos son incompresibles.  
 6. El medio poroso está totalmente saturado, es decir, el volumen poroso está completamente lleno.  
 7. Las fases fluidas permanecen separadas dentro de los poros.  
 8. Todas las fases están en equilibrio termodinámico.  
 9. El medio poroso se considera homogéneo e isotrópico, lo que significa que sus propiedades son invariantes con ubicación y tamaño, y dirección independiente, respectivamente. En particular, en el estado inicial la porosidad es constante ( $\phi = \text{constante}$ ) y su permeabilidad es la misma en todas las direcciones ( $\underline{k} = k\underline{I}$ ), pero se permite la porosidad dinámica y la variación de permeabilidad debido a los procesos de taponamiento/destaponamiento (adsorción/desorción).  
 10. Los microorganismos y los nutrientes interactúan biológicamente entre sí, ya que tienen una cinética de crecimiento tipo Monod

$$\mu = \mu_{max} \left( \frac{c_w^n}{K_{m/n} + c_w^n} \right) \quad (9.4)$$

donde  $\mu_{max}$  es la tasa de crecimiento específico máxima,  $K_{m/n}$  es la constante de afinidad de los microorganismos por los nutrientes,  $c_w^n$  es la concentración de nutrientes en el agua.

Se utilizan modelos de decaimiento lineal para microorganismos tanto planctónicos como sésiles. Y se expresan por

$$k_d \phi S_w c_w^m \quad (9.5)$$

$$k_d \rho_b^m \sigma \quad (9.6)$$

respectivamente, donde  $k_d$  es la tasa de decaimiento específico de las células y suponemos que es la misma para ambas fracciones de microorganismos,  $\rho_b^m$  es la densidad de los microorganismos,  $\sigma$  es la fracción de volumen poroso ocupado por microorganismos y  $c_w^m$  es la concentración de microorganismos planctónicos.

11. Los microorganismos forman una biopelícula. Con una adsorción quasilineal [Corapcioglu Yavuz y Haridas \(1984\)](#) y se utilizan procesos de desorción limitada irreversibles [Lappan y Fogler \(1996\)](#). Los términos de adsorción y desorción empleados son

$$k_a^m \phi S_w c_w^m \quad (9.7)$$

y

$$k_r^m \rho_b^m (\sigma - \sigma_{irr}) \quad (9.8)$$

respectivamente. Aquí  $k_a^m$  y  $k_r^m$  Son los correspondientes coeficientes de adsorción y desorción, y  $\sigma_{irr}$  es la concentración mínima de células sésiles, que nos dice las células que están irreversiblemente adsorbidas dentro de la fase.

12. No existe ningún tipo de discontinuidad en el sistema (no hay saltos).

Fase( $\alpha$ )	Componente	Prop. extensiva	Prop. intensiva
Agua ( $\alpha = w$ )	Agua( $w$ )	$M_w^w(t)$	$\phi S_w \rho_w^w$
	Microorganismo( $m$ )	$M_w^m(t)$	$\phi S_w c_w^m$
	Nutrientes( $n$ )	$M_w^n(t)$	$\phi S_w c_w^n$

Tabla 9.1: Propiedades intensivas y extensivas asociadas con la masa de los componentes por fases.

### 9.3. Modelo matemático

Las ecuaciones de transporte: El modelo de transporte para una componente se derivará en base a la revisión de los modelos anteriores. El modelo de transporte, que se presenta aquí, está constituido por un sistema de tres ecuaciones diferenciales parciales no lineales completamente acoplados [Chang et al. \(1992\)](#). Para modelar las curvas efluentes de la solución de cultivo bacteriano con nutriente, las ecuaciones de transporte que se necesitan resolver son:

**Ecuación de microorganismos Planctónicos ( $c_w^m$ ):**

$$\frac{\partial}{\partial t}(\phi S_w c_w^m) + \nabla \cdot \{c_w^m \underline{u}_w - \phi S_w \underline{D}_w^m \cdot \nabla c_w^m\} = (\mu - k_d - k_a^m) \phi S_w c_w^m + k_r^m \rho_b^m (\sigma - \sigma_{irr}) \quad (9.9)$$

**Ecuación de microorganismos sésiles ( $\sigma$ ) adimensional:**

$$\frac{\partial}{\partial t}(\rho_b^m \sigma) = (\mu - k_d - k_r^m) \rho_b^m \sigma + k_a^m \phi S_w c_w^m + k_r^m \rho_b^m \sigma_{irr} \quad (9.10)$$

**Ecuación de nutrientes ( $c_w^n$ ):**

$$\frac{\partial}{\partial t}(\phi S_w c_w^n) + \nabla \cdot \{c_w^n \underline{u}_w - \phi S_w \underline{D}_w^n \cdot \nabla c_w^n\} = -\frac{\mu}{Y_{\frac{m}{n}}}(\phi S_w c_w^m + \rho_b^m \sigma) \quad (9.11)$$

En resumen de las ecuaciones contienen los siguientes parámetros,  $c_w^\gamma$ ;  $\gamma = m, n$  son las concentraciones de las componentes microorganismos y nutriente en la fase agua,  $\underline{D}_w^\gamma$ ,  $\gamma = m, n$  son los tensores de dispersión hidrodinámica de microorganismos y nutrientes en agua,  $\mu$  es la función de cinética de crecimiento tipo Monod,  $k_d$  es la tasa de decaimiento de células,  $k_a^m$  es la tasa de adsorción de microorganismos,  $k_r^m$  es la tasa de desorción de microorganismos,  $\rho_b^m$  es la densidad de microorganismos,  $\sigma$  es la fracción de volumen poroso ocupado por biomasa,  $\sigma_{irr}$  es la fracción de volumen poroso ocupado irreversiblemente por biomasa,  $Y_{\frac{m}{n}}$  es el coeficiente de producción.

■ **Relaciones de Porosidad y Permeabilidad:**

La modificación de la porosidad debida a los procesos de taponamiento/destaponamiento está teniendo en cuenta la siguiente expresión dada en [Chang et al. \(1992\)](#).

$$\phi = \phi_0 - \sigma \quad (9.12)$$

donde  $\phi$  es la porosidad actual,  $\phi_0$  es la **porosidad** inicial y  $\sigma$  es la fracción de volumen ocupada por los microorganismos sésiles (adimensional).

Considerando que la modificación de la **permeabilidad** se expresa en función de la porosidad mediante la ecuación de [Carman \(1956\)](#).

$$k = k_0 \frac{(1 - \phi_0)^2}{\phi_0^3} \frac{\phi^3}{(1 - \phi)^2} \quad (9.13)$$

donde  $k$  y  $k_0$  son la permeabilidad actual e inicial, respectivamente.

■ **Condiciones iniciales y de frontera**

**Condición inicial:**

$$S_w(t_0) = S_{w0}, c_w^m(t_0) = c_{w0}^m, c_w^n(t_0) = c_{w0}^n, \sigma(t_0) = \sigma_0$$

**Condiciones de frontera:**

Las condiciones de contorno especificadas a continuación consideran un dominio particular, que es una columna porosa cilíndrica:

■ **Condiciones de entrada (velocidad constante) en la parte inferior del núcleo:**

$$\underline{u}_0 \cdot \underline{n} = \underline{n}_w \cdot \underline{n} = \underline{u}_w^{in} \cdot \underline{n}, \quad -[c_w^\gamma \underline{u}_w - \phi S_w \underline{D}_w^\gamma \cdot \nabla c_w^\gamma] \cdot \underline{n} = c_{w_{iny}}^\gamma \underline{u}_w^{in} \cdot \underline{n}; \quad \gamma = m, n \quad (9.14)$$

donde  $\underline{u}_w^{in}$  es la velocidad de inyección de agua, y  $c_{w_{iny}}^\gamma \neq 0$ , son concentraciones inyectadas para microorganismos ( $\gamma = m$ ), y para nutrientes ( $\gamma = n$ ), respectivamente.

■ **Condiciones de salida en la parte superior del núcleo**

$$\frac{\partial S_w}{\partial \underline{n}} = 0, \quad \frac{\partial c_w^\gamma}{\partial \underline{n}} = 0. \quad \gamma = m, n$$

■ **Condiciones sin flujo para otras fronteras.**

Dada las ecuaciones anteriores vamos a replantear esas ecuaciones para poder hacer la implementación computacional.

## Ecuación de microorganismos planctónicos ( $c_w^m$ ):

$$\frac{\partial}{\partial t}(\phi S_w c_w^m) + \nabla \cdot \{c_w^m \underline{u}_w - \phi S_w \underline{D}_w^m \cdot \nabla c_w^m\} = (\mu - k_d - k_a^m) \phi S_w c_w^m + k_r^m \rho_b^m (\sigma - \sigma_{irr})$$

Dado que la saturación para la fase agua ( $w$ ), es  $S_w = 1$ , la ecuación nos queda como:

$$\frac{\partial}{\partial t}(\phi S_w c_w^m) + \nabla \cdot \{c_w^m \underline{u}_w - \phi S_w \underline{D}_w^m \cdot \nabla c_w^m\} = (\mu - k_d - k_a^m) \phi S_w c_w^m + k_r^m \rho_b^m (\sigma - \sigma_{irr})$$

Esta misma ecuación la podemos ver como:

$$\phi \frac{\partial c_w^m}{\partial t} + \nabla \cdot \{c_w^m \underline{u}_w - \phi \underline{D}_w^m \cdot \nabla c_w^m\} - (\mu - k_d - k_a^m) \phi S_w c_w^m = k_r^m \rho_b^m (\sigma - \sigma_{irr})$$

Ahora vamos a adaptar esta ecuación a la forma de nuestra ecuación del operador elíptico de segundo orden que hemos estado trabajando:

$$R \frac{\partial c}{\partial t} - \nabla \cdot (\underline{D} \cdot \nabla c) + \nabla \cdot (c \underline{u}) + cr = q$$

de aquí tenemos:

$$R_1 = \phi$$

$$c_1^m = c_w^m$$

$$u_1 = u_w$$

$$\underline{D}_1 = \phi S_w \underline{D}_w^m$$

$$r_1 = (\mu - k_d - k_a^m) \phi S_w$$

$$q_1 = k_r^m \rho_b^m (\sigma - \sigma_{irr})$$

quedando la ecuación de la siguiente manera:

$$R_1 \frac{\partial c_1^m}{\partial t} - \nabla \cdot (\underline{D}_1 \cdot \nabla c_1^m) + \nabla \cdot (c_1^m \underline{u}_1) + c_1^m r_1 = q_1$$

## Ecuación de microorganismos sésiles ( $\sigma$ ) adimensional:

$$\frac{\partial}{\partial t}(\rho_b^m \sigma) = (\mu - k_d - k_r^m)\rho_b^m \sigma + k_a^m \phi S_w c_w^m + k_r^m \rho_b^m \sigma_{irr}$$

así la ecuación nos queda como:

$$\rho_b^m \frac{\partial \sigma}{\partial t} - (\mu - k_d - k_r^m)\rho_b^m \sigma = k_a^m \phi S_w c_w^m + k_r^m \rho_b^m \sigma_{irr}$$

de aquí tenemos:

$$R_2 = \rho_b^m$$

$$c_2 = \sigma$$

$$r_2 = (\mu - k_d - k_r^m)\rho_b^m$$

$$q_2 = k_a^m \phi S_w c_1^m + k_r^m \rho_b^m \sigma_{irr}$$

$$R_2 \frac{\partial c_2}{\partial t} - r_2 c_2 = q_2$$

## Ecuación de nutrientes ( $c_w^n$ ):

$$\frac{\partial}{\partial t}\{(\phi S_w + h\bar{\rho}_r^r)c_m^n\} + \nabla \cdot \{c_w^n \underline{u}_w - \phi S_w \underline{D}_w^n \cdot \nabla c_w^n\} = -\frac{\mu}{Y_n^m}(\phi S_w c_w^m + \rho_b^m \sigma)$$

Como el termino  $h\bar{\rho}_r^r = 0$  y haciendo la manipulación correspondiente la ecuación nos queda:

$$\frac{\partial}{\partial t}(\phi S_w c_w^n) + \nabla \cdot \{c_w^n \underline{u}_w - \phi S_w \underline{D}_w^n \cdot \nabla c_w^n\} = -\frac{\mu}{Y_n^m}(\phi S_w c_w^m + \rho_b^m \sigma)$$

ahora tenemos:

$$\phi \frac{\partial c_w^n}{\partial t} + \nabla \cdot \{c_w^n \underline{u}_w - \phi S_w \underline{D}_w^n \cdot \nabla c_w^n\} + c_w^n \frac{\partial \phi}{\partial t} = -\frac{\mu}{Y_n^m}(\phi c_w^m + \rho_b^m \sigma)$$

así obtenemos:

$$R_3 = \phi$$

$$c_3^n = c_w^n$$

$$\underline{D}_3 = \phi S_w \underline{D}_w^n$$

$$r_3 = \frac{\partial \phi}{\partial t} = 0$$

$$q_3 = -\frac{\mu}{Y_n^m}(\phi S_w c_1^m + \rho_b^m \sigma)$$

La cual podemos ver como:

$$R_3 \frac{\partial c_3^n}{\partial t} - \nabla \cdot (\underline{D}_3 \cdot \nabla c_3^n) + \nabla \cdot (c_3^n \underline{u}_1) = q_3$$

## 9.4. Modelo numérico

El modelo numérico consiste en hacer la elección apropiada de los métodos numéricos en términos de precisión y eficiencia para la solución del modelo matemático. En este caso el problema resultante es un sistema no lineal de ecuaciones diferenciales parciales con condiciones iniciales y de frontera.

Para la solución numérica aplicamos los siguientes métodos:

- Para las derivadas temporales se utilizó una discretización de diferencias finitas hacia atrás de segundo orden, resultando en un esquema completamente implícito en el tiempo.  
Para el resto de operadores diferenciales en las derivadas espaciales se aplicó una discretización estándar de elementos finitos, en la que se emplearon los polinomios cuadráticos de Lagrange para funciones de base y de peso.
- Se utilizó una malla no estructurada con elementos triangulares en 2D.
- Para la linealización del sistema de ecuaciones no lineales se aplicó el método iterativo de Newton-Raphson.
- Para la solución del sistema algebraico de ecuaciones resultante se utilizó una variante del método LU con el paquete UMFPACK para matrices no simétricas y ralas.
- El procedimiento general para acoplar las ecuaciones de transporte es secuencial y se ejecuta iterativamente de la siguiente manera:
  1. Se introducen las velocidades constantes de fase  $u_\alpha$  para  $\alpha = w$ .
  2. El modelo de transporte se resuelve y se obtiene: concentraciones de los componentes  $c_w^m$ ,  $c_w^n$  y  $\sigma$ .
  3. La porosidad y la permeabilidad se modifican de acuerdo con las ecuaciones (9.12) y (A.2).
  4. Cuando se alcanza una precisión determinada, el procedimiento se detiene, pero en el caso contrario el procedimiento iterativo debe continuar pasando al Paso 1.

## 9.5. Implementación computacional

Una vez que se ha establecido el procedimiento numérico para la solución del modelo matemático, se requiere su implementación computacional.

### 9.5.1. Geometría de la malla

La implementación computacional del modelo de transporte también se llevó a cabo en el software proyecto FEniCS 2016.1.

### Dominio

El dominio del modelo computacional en 2D es un rectángulo en posición vertical con las siguientes dimensiones: Longitud  $l = 0.25$  m ancho  $d = 0.04$  m

## Mallado

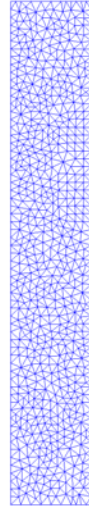


Figura 9.1: La discretización de la La malla para la solución numérica, está formada por 1712 vértices y 3224 elementos triangulares. Peclet Local máximo = 3.4816, hmax = 0.0039

### 9.5.2. Condiciones de fronteras

Mostraremos la figura con sus respectivas condiciones de frontera que se utilizan en el problema. Las condiciones de frontera se colocaron en base a la numeración dada en la (figura 7.3) que se utilizarán de la siguiente manera:

1. Para la frontera número (1) se utilizó una condición de frontera tipo Dirichlet, con una inyección de  $c = \text{constante}$ .
2. Para la frontera (2) y (4) se utilizará condiciones de frontera tipo Neumann, ya que son fronteras de no flujo es decir  $\underline{n} \cdot (\underline{D} \cdot \nabla u - \underline{b}u) = 0$ .
3. Para la frontera (3) también utilizaremos una condición tipo Neumann donde la derivada normal sea igual a cero, ya que esa frontera será la salida del flujo  $\frac{\partial c}{\partial n} = 0$ . Aquí el flujo dispersivo es cero, pues ya no hay medio poroso en el exterior.

## 9.6. Simulaciones numéricas

El caso de estudio es un ensayo de inundación de agua en un núcleo, el experimento que se realiza se inyecta agua con microorganismos y nutrientes. Chang *et al.* (1991), realizaron un experimento de inyección de cultivo bacteriano y nutriente en núcleo de arenisca Berea. Utilizamos sus datos (ver tabla 9.2) para simular un experimento de inyección continua de solución de cultivo bacteriano con nutriente.

Los resultados obtenidos en el proyecto FEniCS se validan comparados con los obtenidos con el software comercial COMSOL *Multiphysics*®.

Propiedad (Símbolo)	Valor [Unidad]
Porosidad ( $\phi$ )	0.2295
Densidad de microorganismos ( $\rho_b^m$ )	1000 [kg/m]
Fracción de biomasa irreducible ( $\sigma_{irr}$ )	0.0
Velocidad de inyección del agua ( $u_w$ )	$3.53e - 6$ [m/s]
Coefficiente de dispersión de nutrientes ( $D_w^n$ )	$8.9247e - 9$ [m <sup>2</sup> /s]
Coefficiente de dispersión de microorganismos ( $D_w^m$ )	$5.9139e - 9$ [m <sup>2</sup> /s]
Tasa de crecimiento específico máxima ( $\mu_{max}$ )	$9.7222e - 5$ [1/s]
Coefficiente de afinidad ( $k_{m/n}$ )	8.0092 [kg/m <sup>3</sup> ]
Coefficiente de decaimiento ( $k_d$ )	$2.5429e - 6$ [1/s]
Coefficiente de producción ( $Y_{m/n}$ )	0.5
Coefficiente de desorción ( $k_r^m$ )	4.2824 [1/s]
Coefficiente de adsorción ( $k_a^m$ )	$2.8935e - 4$ [1/s]
Concentración inyectada de nutrientes ( $c_{winy}^n$ )	40.0461 [kg/m <sup>3</sup> ]
Concentración inyectada de microorganismos ( $c_{winy}^m$ )	30.0346 [kg/m <sup>3</sup> ]
Longitud del núcleo ( $x_L$ )	0.25 [m]

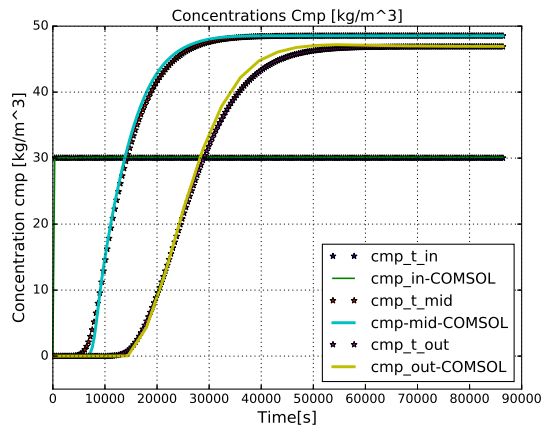
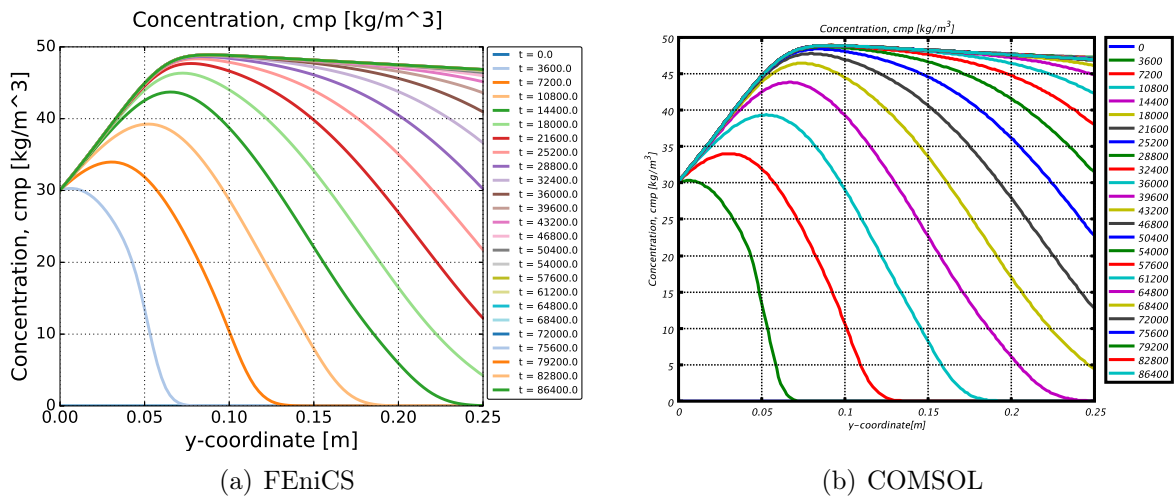
Tabla 9.2: Parámetros para el problema de inyección continua y simultánea de microorganismos y nutrientes a través de un núcleo de arena Berea.

## COMSOL *Multiphysics*<sup>®</sup>

Para la validación de este caso de estudio se comparan los resultados obtenidos con resultados ya obtenidos con el software comercial COMSOL *Multiphysics*<sup>®</sup>, de cual damos una breve introducción de su uso.

COMSOL *Multiphysics*<sup>®</sup> fue fundado en 1986 por Svante Littmarck y Farhad Saeidi (estudiantes de posgrado de Germund Dahlquist), basado en el código desarrollado para un curso de posgrado en el Royal Institute of Technology (KTH) en Estocolmo, Suecia.

El software COMSOL *Multiphysics*<sup>®</sup> es un paquete de análisis de elementos finitos, y simulación para diversas aplicaciones de física e ingeniería, especialmente fenómenos acoplados o multifísicos. El paquete es multiplataforma (Windows, Mac, Linux). Además de las interfaces de usuario físicas convencionales, COMSOL *Multiphysics*<sup>®</sup> también permite introducir sistemas acoplados de ecuaciones diferenciales parciales (PDEs). Las PDEs se pueden introducir directamente o utilizando la denominada forma débil. Desde la versión 5.0 (2014), COMSOL *Multiphysics*<sup>®</sup> también se utiliza para crear aplicaciones basadas en la física [COMSOL Multiphysics](#) (2008).



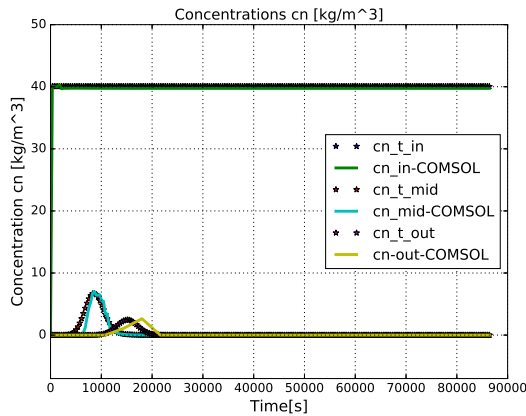
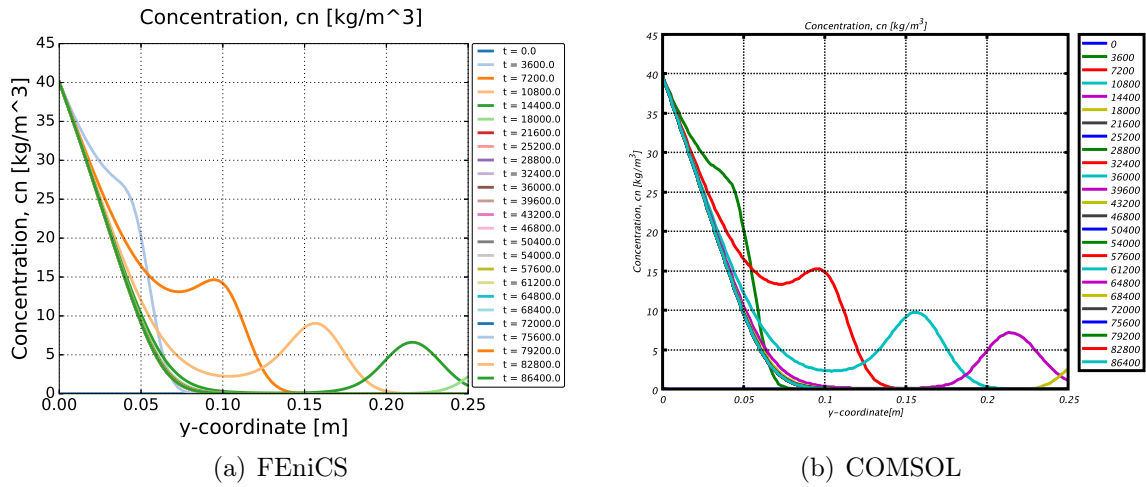
(c) Comparación de las concentraciones en tres puntos, al inicio, en un punto medio y al final del núcleo, donde las estrellas representan las concentraciones en Proyecto FEniCS 2016.1, y la línea continua las concentraciones de los valores de COMSOL *Multiphysics*®.

Figura 9.2: Perfiles de microorganismos planctónicos ( $c_w^m$ ) a lo largo del eje vertical en varias secuencias de una hora, en la plataforma proyecto FEniCS 2016.1 (a) y el software COMSOL *Multiphysics*® (b).

Las figuras (9.2) (a) y (b) se muestra la comparación de las distribuciones espacio-temporal de la concentraciones de los microorganismos planctónicos  $c_w^m$  a lo largo del núcleo para un tiempo de 24 horas, con secuencias de una hora, se puede observar que la concentración máxima alcanzada por los microorganismos es de  $c_w^m = 48.85 \text{ kg/m}^3$  y esto se alcanza en 0.74 m del núcleo hasta llegar a un estado estacionario después de la concentración inicial, como se muestra en la gráfica del inciso (a) obtenida en el proyecto FEniCS en la cual nos muestra la semejanza cualitativamente con la gráfica obtenida en el software COMSOL, como se puede observar en la comparación de las dos gráficas las concentraciones inician el en mismo punto  $c_w^m = 30.034619 \text{ kg/m}^3$ , tanto en la gráfica con el



proyecto FEniCS y como la obtenida con el software comercial COMSOL *Multiphysics*<sup>®</sup>(b), el espaciamiento del effluente es de una hora en ambas gráficas distribuidas a lo largo de núcleo del mismo tamaño para ambos casos. En la figura (c) para realizar el análisis del error se seleccionaron tres puntos a lo largo de núcleo, un punto al inicio, un punto en medio y un punto al final. Podemos observar las concentraciones de los microorganismos obtenidas en el proyecto FEniCS y las concentraciones obtenidas en el software COMSOL *Multiphysics*<sup>®</sup>, son prácticamente las mismas obteniendo así un error relativo de 1.5753, con esto se considera validado el caso de estudio.

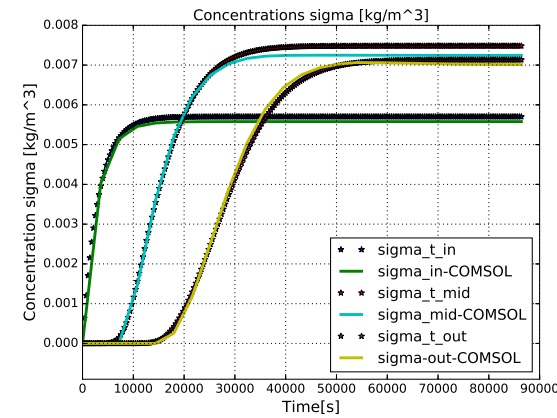
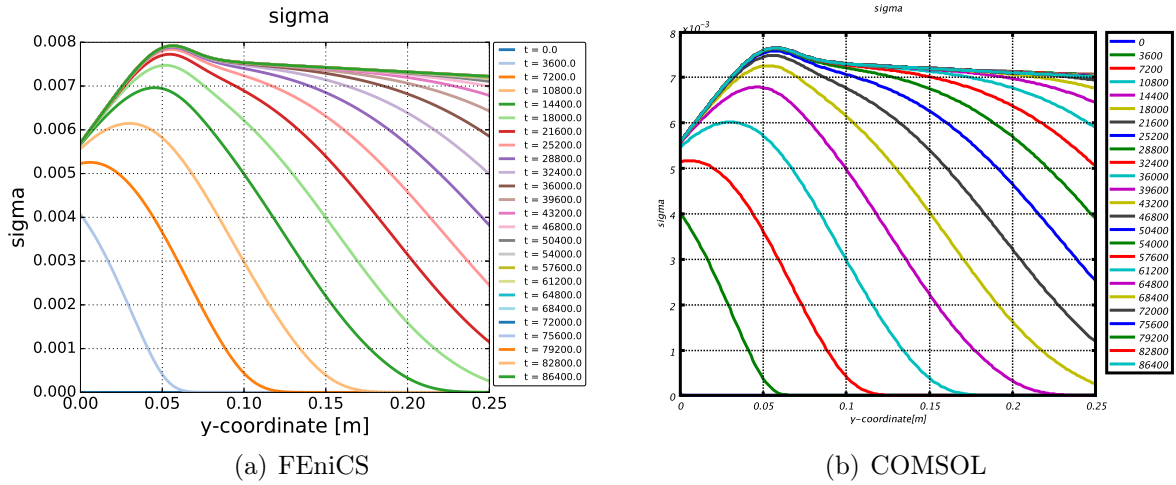


(c) Comparación de las concentraciones en tres puntos, al inicio, en un punto medio y al final del núcleo, donde las estrellas representan las concentraciones en Proyecto FEniCS 2016.1, y la línea continua las concentraciones de los valores de COMSOL *Multiphysics*®.

Figura 9.3: Perfiles de nutrientes ( $c_w^n$ ) a lo largo del eje vertical en varias secuencias de una hora, en la plataforma proyecto FEniCS 2016.1 (a) y el software COMSOL *Multiphysics*® (b).

Las figuras (9.3) (a) y (b) se muestra la comparación de las distribuciones espacio-temporal de las concentraciones de los nutrientes  $c_w^n$  a lo largo del núcleo para un tiempo de 24 horas, con secuencias de una hora, se puede observar que los nutrientes se han consumido por completo tomando un comportamiento asintótico al final la de las 24 horas, lo cual se puede ver en la gráfica del inciso (a) obtenida en el proyecto FEniCS en la cual podemos ver la semejanza cualitativamente, con la gráfica obtenida con el software COMSOL, como se puede observar en la comparación de las dos gráficas las concentraciones inician el en mismo punto  $c_w^m = 40.046158 \text{ kg/m}^3$ , tanto en la gráfica con el proyecto FEniCS y como la obtenida con el software comercial COMSOL *Multiphysics*®, el espaciamiento del efluente es de una hora en ambas gráficas distribuidas a lo largo de núcleo del mismo tamaño

para ambos casos, las ondas que se muestran en ambas comparaciones creo que se debe a algunos nutrientes que se le están escapando a los microorganismos en el proceso. En la figura (c) para el análisis del error se seleccionaron tres puntos a lo largo de núcleo, un punto al inicio, un punto en medio y un punto al final. Podemos observar que las concentraciones de los microorganismos obtenidas en el proyecto FEniCS y las obtenidas en el software COMSOL *Multiphysics*<sup>®</sup> son prácticamente las mismas obteniendo así un error relativo máximo de 1.7328.



(c) Comparación de las concentraciones en tres puntos, al inicio, en un punto medio y al final del núcleo, donde las estrellas representan las concentraciones en Proyecto FEniCS 2016.1, y la línea continua las concentraciones de los valores de COMSOL Multiphysics.

Figura 9.4: Perfiles de microorganismos sésiles ( $\sigma$ ) a lo largo del eje vertical en múltiples secuencias de una hora, en la plataforma proyecto FEniCS 2016.1 (a) y el software COMSOL *Multiphysics*<sup>®</sup> (b).

Las figuras (9.4) (a) y (b) se muestra la comparación de las distribuciones espacio-temporal de los organismos sésiles  $\sigma$  a lo largo del núcleo para un tiempo de 24 horas, con secuencias de una hora, la variación de los cambios de  $\sigma$  se refleja directamente en la porosidad y la permeabilidad se puede observar que el valor máximo es  $\sigma = 0.0079$  obtenida en el proyecto FEniCS (a) y la comparación muestra la semejanza cualitativamente con la gráfica obtenida con el software comercial COMSOL *Multiphysics*<sup>®</sup> (b). En la figura (c) se realiza el análisis del error para lo cual se seleccionaron tres puntos a lo largo de núcleo, se toma el primer punto al inicio, un punto en medio y al final. Podemos observar que las distribuciones obtenidas en el proyecto FEniCS y las obtenidas en el software COMSOL *Multiphysics*<sup>®</sup> son prácticamente las mismas, muestran mucha congruencia, prácticamente son las mismas y en

la salida se puede ver como varían un poco las concentraciones de FEniCS y COMSOL obteniendo así un error relativo máximo de 0.002.

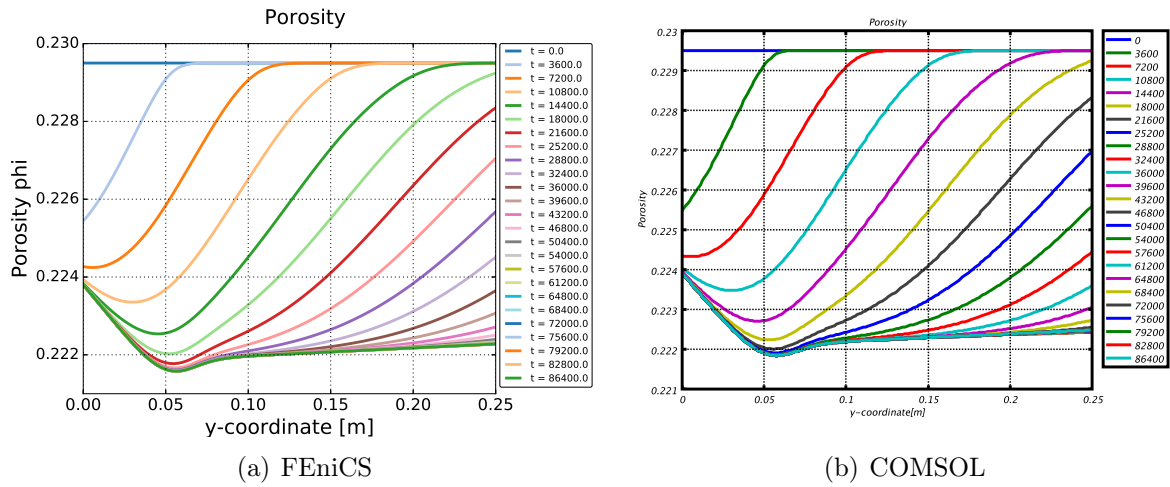


Figura 9.5: Perfiles de la porosidad a lo largo del eje vertical en varias secuencias de una hora.

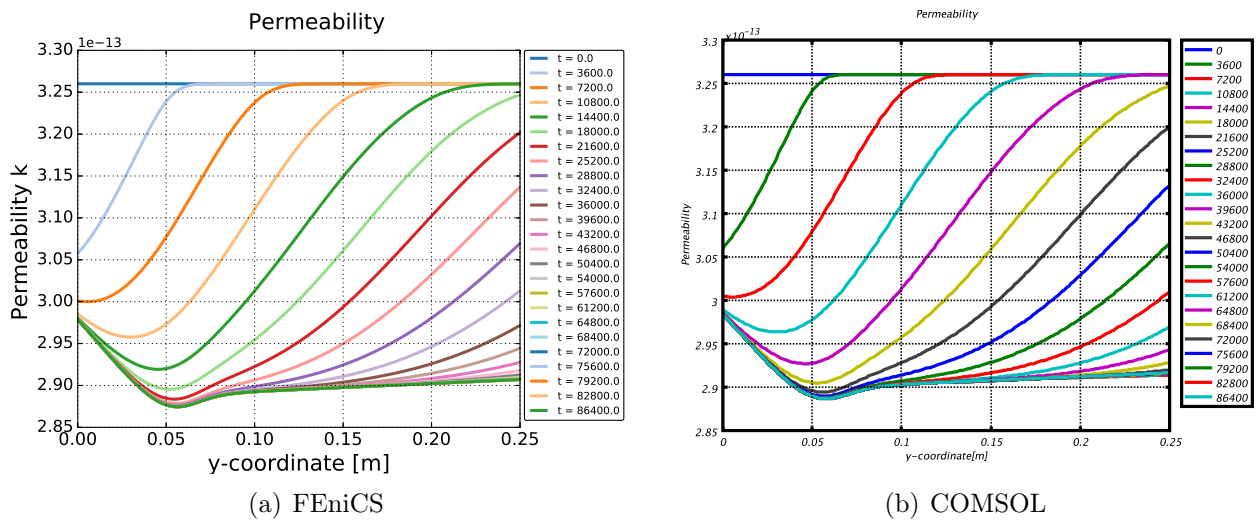


Figura 9.6: Perfiles de la permeabilidad a lo largo del eje vertical en varias secuencias de una hora.

En este trabajo, se investigó la modificación de porosidad y la permeabilidad debida a la variación en la distribución de biomasa a lo largo del núcleo. La figura (9.5) muestra el resultado de la modificación de la porosidad debida a la variación de la biomasa y la comparación de la porosidad obtenida en el software COMSOL *Multiphysics*<sup>®</sup> en la cual podemos ver que los resultados tienen una similitud cualitativa, utilizando tiempos de 24 horas en las dos simulaciones y el espaciamiento de una hora para ambos casos. La figura (9.6) muestra el resultado de la modificación de la permeabilidad debida a la variación de la biomasa a lo largo del núcleo y las comparaciones de la permeabilidad obtenida en el proyecto FEniCS y la permeabilidad obtenida en el software COMSOL *Multiphysics*<sup>®</sup> las cuales presentan una similitud cualitativa, en ambas simulaciones realizadas.

La semejanza de la porosidad ( $\phi$ ) y la permeabilidad ( $k$ ) de las distribuciones espacio-temporales de la forma de la concentración de microorganismos sésiles ( $\sigma$ ), puede atribuirse a la simplicidad de las relaciones de porosidad y permeabilidad aplicadas. Las gráficas describen el comportamiento de la biomasa con la relación de la porosidad y la permeabilidad podemos observar que en la figuras tanto de la porosidad como la de la permeabilidad se aprecia que hay una zona donde alcanzan un valor máximo, ahí podría haber un mayor crecimiento de microorganismos y luego se produce un decaimiento de nutrientes, por eso las gráficas se van volviendo estacionarias, se puede pensar que los nutrientes están siendo consumidos por los microorganismos y los esta moviendo el flujo de agua y el transporte difusivo, podemos ingerir que hay una parte taponada y luego hay destaponamiento, cuando hay mayor crecimiento ahí se puede estar taponando y luego se produce un decaimiento, esa relación taponamiento/destaponamiento es lo que hace que se muestren esos patrones en los resultados obtenidos que se pueden manipular con los parámetros de los coeficientes de desorción  $k_r^m$  y el coeficiente de adsorción  $k_a^m$  (taponamiento/destaponamiento) claro todo depende de la velocidad de inyección  $u_w$  del experimento, para este trabajo se utilizó una velocidad del agua de  $3.53e - 6$  m/s, obteniendo estos resultados.

### 9.6.1. Análisis físico del caso de estudio

Para realizar un análisis mas profundo del comportamiento del proceso de taponamiento/destaponamiento en nuestro caso de estudio realizamos la ilustración para algunas concentraciones de microorganismos y nutrientes juntos y ver como se esta comportando la porosidad y la permeabilidad en el proceso, recordando que es una sola fase, el medio esta completamente saturado e inyectamos microorganismos y nutrientes simultáneamente a una concentración constante, como se muestran en las figuras siguientes:

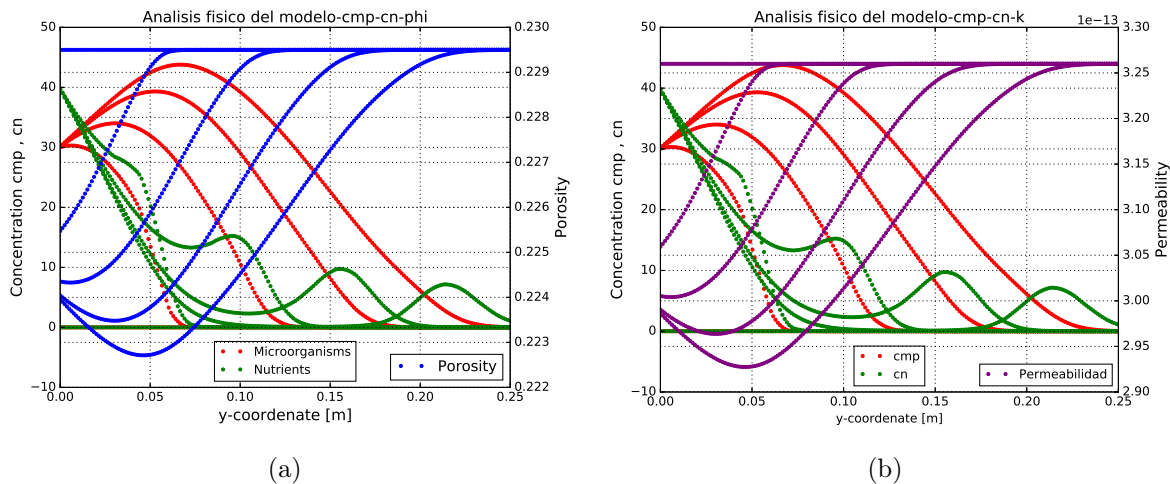


Figura 9.7: Comportamiento en las primeras 4 horas continuas de las concentraciones de los microorganismos, nutrientes y porosidad a lo largo del eje vertical en secuencias de una hora.

En las gráficas podemos como se inyecta la concentración inicial y como va aumentando la concentración de microorganismos y estos van consumiendo a los nutrientes, pero se ve que hay nutrientes que ya avanzaron sin consumirse, eso se debe a que los microorganismos tiene sorción por eso avanzan mas lentos que los nutrientes, eso hace que algunos nutrientes logran pasar ya que se comportan como un soluto conservativo con respecto al medio físico y además no tienen sorción, con respecto al medio biológico (microorganismos) lo único que afecta a los nutrientes es que se los coman los microorganismos.

En estas gráficas figuras (9.7), podemos ver el comportamiento figura (a) de los microorganismos al ser inyectados inicialmente junto con los nutrientes para que se alimenten crezcan y se reproduzcan, podemos apreciar que al principio los microorganismos crecen alcanzan su máximo crecimiento, y se están consumiendo a los nutrientes por eso los nutrientes van decayendo aunque algunos se escapan como se puede ver en algunas de las líneas de la gráfica, para esto también podemos ver que la porosidad y la permeabilidad figura (b) están alcanzando su punto máximo y luego de un tiempo los microorganismos mueren y eso se aprecia justo donde la porosidad alcanza su valor máximo y lo mismo pasa con la permeabilidad, claramente se ve como los el comportamiento de los microorganismos se va volviendo estacionario así como la permeabilidad y la porosidad.

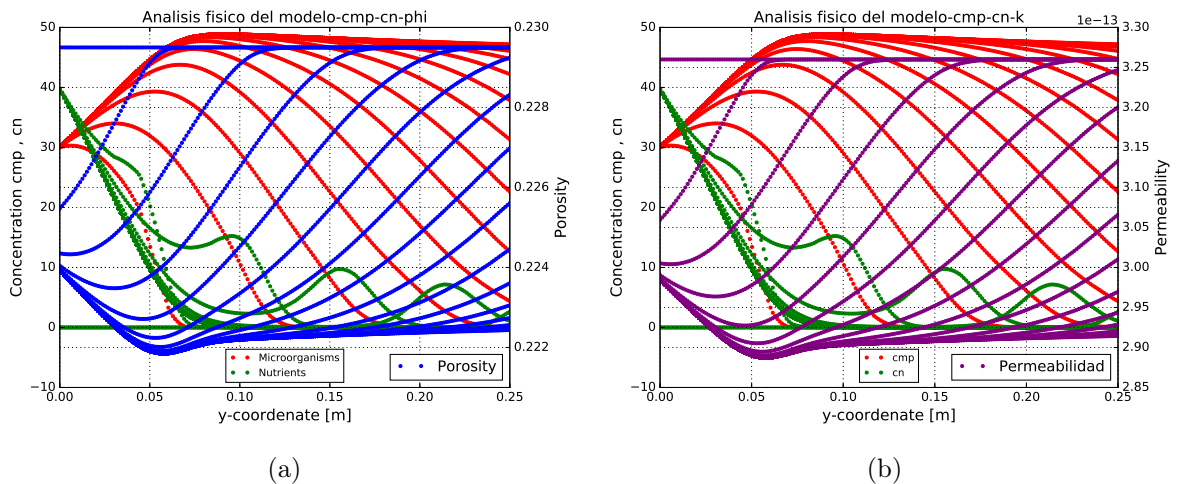


Figura 9.8: Comportamiento de las concentraciones las 24 horas de los microorganismos, nutrientes y porosidad a lo largo del eje vertical en secuencias de una hora.

Finalmente se gráfica para el tiempo total del experimento que es de 24 horas, con secuencias de una hora para cada concentración como se ve en las figuras (9.8), en la figura (a) se ve el comportamiento total del caso de estudio donde se muestran las concentraciones de los microorganismos junto a los nutrientes y porosidad, mismo situación que repetimos en la figura (b) para el caso de la permeabilidad, se aprecia que el comportamiento es el mismo, solo que después de una horas el proceso se vuelve estacionario en ambos casos.

## 9.7. Discusión de resultados

Este modelo desarrollado anteriormente se aplicó con éxito a un caso de estudio de referencia de taponamiento y destaponamiento por la inyección de agua con microorganismos y nutrientes, utilizando datos de la literatura publicada y luego se evaluó su rendimiento comparando variaciones de la referencia del caso de estudio. El error relativo que se obtuvo en la realización y la comparación del problema fue de 0.00198, podemos ver que las gráficas obtenidas en las simulaciones con el proyecto FEniCS muestran una distribución cualitativamente iguales a las realizadas con el software COMSOL *Multiphysics*<sup>®</sup>. Se investigó la modificación de porosidad y permeabilidad debido a la variación en la distribución de biomasa a lo largo del núcleo. La semejanza de la porosidad ( $\phi$ ) y la permeabilidad ( $k$ ) de las distribuciones espacio-temporales de la forma de la concentración de microorganismos sésiles ( $\sigma$ ), se puede atribuir a la simplicidad de las relaciones de porosidad y permeabilidad aplicadas, por lo tanto la aplicación de una dependencia más realista es un problema abierto. Después de ese análisis cualitativo que obtuvimos resultados congruentes se realiza una análisis mas cuantitativo realizando la comparación de las



concentraciones y distribuciones de microorganismos a lo largo del núcleo en tres puntos diferentes obteniendo así diferencias de errores muy razonables para poder mostrar que los resultados obtenidos son aceptables para este trabajo, cabe aclarar que los dos resultados mostrados tanto del proyecto FEniCS como los de software COMSOL tienen errores, lo que estamos haciendo es ver que hay consistencia que nos sirve para comparar dos soluciones realizadas de manera independiente, y que están usando el mismo método numérico para obtener las soluciones, y en los dos casos se obtienen los mismos resultados lo que nos sirve para mostrar que nuestro trabajo es congruente. En cuanto a la recuperación adicional marginal producida durante el proceso microbiano EOR, inferior al 1%. Es pertinente mencionar aquí, que el objetivo del estudio fue la investigación del fenómeno de obstrucción microbiana de los medios porosos. Un sistema para simular adecuadamente el taponamiento y destaponamiento. Para modelar otros efectos tales como: mojabilidad de rocas, viscosidad, permeabilidad relativa y modificación de las curvas de presión capilar debido a la acción de los bioproductos sobre fluidos, se requiere una investigación experimental adicional para cuantificar en términos de relaciones constitutivas la interacción entre propiedades petrofísicas y fluidas como una función de las concentraciones de bioproductos, además de un modelo de flujo bifásico agua-aceite.

## Conclusiones y Trabajo Futuro

A partir de la formulación axiomática de la modelación de medios continuos, para el desarrollo de modelos de transporte de microorganismos propuesto en este trabajo, se puede observar que se requiere tener control de varios factores para poder obtener resultados consistentes con el fenómeno en la naturaleza.

La propuesta del modelo conceptual es uno de los pasos más importantes en el desarrollo de un modelo, ya que a partir de esta decisión de supuestos se desarrolla todo el modelo.

Una vez obtenido el modelo conceptual y usando la derivación sistemática de medios continuos se pueden obtener el modelo matemático que es consistente con el modelo conceptual.

Después a partir del modelo matemático obtenido se obtiene el modelo numérico para su desarrollo se utilizó el método de elemento finito para la solución del problema variacional, se escogió la familia de elementos que mejor aproximo la solución, los elementos utilizados fueron los polinomios de Lagrange, para problemas con advección dominante se aplica los diferentes métodos para controlar ese perturbaciones en el problema como el números de Péclet globales y locales, aunque los números de Péclet globales son característicos del problema físico, se deben controlar lo más posible para facilitar la convergencia del método numérico y reducir las perturbaciones en la solución.

Los números de Péclet locales son relevantes para el control numérico de la solución, lo cual podemos controlar reduciendo la malla, si estos son descuidados se pueden obtener predicciones con un grado de distorsión importante y además de afectar a la solución, perjudicando a otros sistemas que dependen de este.

Es importante tomar en cuenta la diferencia en escala espacial y de tiempo en la que ocurren las pruebas de laboratorio y los proyectos de recuperación adicional en yacimientos y modificar los parámetros del modelo con el fin de tener una simulación acertada del proceso real.

Una vez obtenido el modelo numérico se implementa el modelo computacional del modelo desarrollado, para lo cual utilizamos la plataforma de software libre Proyecto FEniCS 2016.1 en Python, que consideramos que es un software que alcanzado cierto grado de madurez para desarrollar modelos más sofisticados. Aplicando refinamiento de la malla para poder obtener una malla óptima para la solución del problema, en este caso debido a que las matrices no son muy grandes se escogió un método directo para la solución de las matrices el cual fue LU con el paquete UMFPACK de FEniCS para matrices no simétricas y ralas.

El modelo ha sido validado con problemas controlados y se controló el error, se utilizó un caso de estudio tomado del artículo de Van Genuchten y W. J. Alves que es un problema de inyección de un escalón.

Se realizaron dos casos de estudios que fueron comparados con el software COMSOL *Multiphysics*<sup>®</sup>, el primer caso de estudio fue inyección de pulso trazador inerte y el segundo caso de estudio fue proceso de taponamiento y destaponamiento por microorganismos los cuales no dieron resultados muy consistentes.

El trabajo realizado puede servir de base para desarrollar modelos más sofisticados de flujo y transporte en medios porosos que pudieran ser aplicados a pruebas de recuperación mejorada a escala de laboratorio.

Como trabajo futuro se debe acoplar el modelo de transporte desarrollado con un modelo de flujo, así como extenderlo a tres dimensiones y un modelo de flujo bifásico agua-aceite. Además, se pudieran explorar otras variantes del método de elemento finito como son los métodos mixtos y de Galerkin discontinuo.

## Glosario de definiciones relevantes

A continuación se describen los conceptos de algunos parámetros mas relevantes utilizados en el trabajo de tesis.

El término **adsorción**, se refiere al proceso de acumulación de moléculas en la interfase, al proceso inverso se le conoce como *desorción*. Cuando el componente es adsorbido se le define como adsorbato, este mismo en su fase original es llamado adsortivo. Un error común es confundir este fenómeno con la *absorción*, la diferencia fundamental reside en que si las moléculas de los adsorbatos se mantienen exclusivamente en la interfase con el adsorbente, el proceso se denomina **adsorción**, por otro lado, si estas penetran al interior de adsorbente, al proceso se le llama **absorción**.

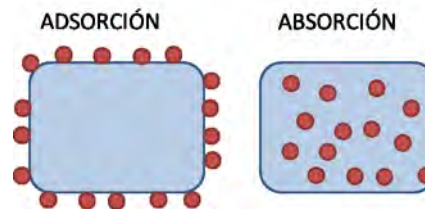


Figura A.1: Diferencia entre absorción y adsorción.

Frecuentemente estos dos fenómenos ocurren al mismo tiempo o son indistinguibles el uno del otro, a este proceso se le denomina sorción.

### A.0.1. Núcleos de Berea

Los núcleos de Berea han sido ampliamente reconocidos por la industria petrolera como la mejor muestra para probar la eficacia y eficiencia de los nuevos tratamientos químicos y físicos asociados a un yacimiento.

La Arenisca de Berea es una roca sedimentaria cuyos granos se componen de cuarzo unidos por sílice. La relativamente alta porosidad y permeabilidad de los núcleos de Berea (figura A.2) hace que sea una buena roca del yacimiento y almacenadora de hidrocarburos.



Figura A.2: Núcleo de Berea

## A.0.2. Inóculo microbiano

El inóculo puede ser un conjunto de diversas especies microbianas, como las que se encuentran en los lodos aerobios de las plantas de tratamiento de aguas residuales, o puede ser un cultivo de microorganismos especializados con afinidad por un compuesto.

### A.0.2.1. Porosidad

Se considera que la porosidad es una función de la porosidad inicial y del cambio en la ocupación de los poros, por los microorganismos, que a su vez depende de la posición y del tiempo [Chen \(2007\)](#).

$$\phi = \phi - \sigma \quad (\text{A.1})$$

### A.0.2.2. Permeabilidad

La modificación de la permeabilidad absoluta se expresa como una función de la porosidad a través de la ecuación de [Carman \(1956\)](#).

$$k = k_0 \frac{(1 - \phi_0)^2}{\phi_0^3} \frac{\phi^3}{(1 - \phi)^2} \quad (\text{A.2})$$

donde  $k$  y  $k_0$  son la permeabilidad actual e inicial, respectivamente.

## A.0.3. Propiedades del fluido

### A.0.3.1. Fase

La fase se refiere a una región del sistema en estudio, que posee una velocidad común, de forma que puede distinguirse de otra por una interfase, por ejemplo, oleica (aceite), acuosa (principalmente agua), o gaseosa (gases). Las fases serán descritas por superíndices  $w$  (acuosa),  $o$  (oleica), y  $g$  (gaseosa). La fase sólida (roca) se indica por  $s$  (sólida). La fase biopelícula (microorganismos sésiles) se denota por  $b$  (biopelícula).

### A.0.3.2. Componente

Una componente es una especie química simple que puede estar presente en una fase. Por ejemplo, la fase acuosa puede contener las componentes agua ( $H_2O$ ), cloruro de sodio ( $NaCl$ ), y oxígeno disuelto ( $O_2$ ), y la fase oleica puede contener cientos de componentes del hidrocarburo, por ejemplo,  $C_1, C_2, C_3$ , etc. Las componentes serán descritas por subíndices.

### A.0.3.3. Tipos de fluidos de yacimiento

En general, en un yacimiento de hidrocarburos pueden existir simultáneamente agua, aceite y gas. Estos fluidos se pueden clasificar como incompresible, ligeramente compresible, o compresible, dependiendo de cómo responden a la presión. Un fluido incompresible tiene compresibilidad cero; su densidad es independiente de la presión. El agua y el aceite libre de gas (muerto) pueden ser incompresibles. Un fluido ligeramente compresible, tiene una compresibilidad pequeña pero constante que típicamente varía de  $10^{-5}$  a  $10^{-6} \text{ psi}^{-1}$ . En condiciones de yacimiento, el agua, el aceite muerto y el bajo saturado se comportan como fluidos ligeramente compresibles. Un fluido compresible típicamente tiene una compresibilidad en el rango de  $10^{-3}$  a  $10^{-4} \text{ psi}^{-1}$ , su densidad incrementa conforme la presión incrementa pero tiende a estabilizarse a presiones altas.

### A.0.3.4. Métodos de recuperación de aceite

Los métodos de recuperación de aceite incluyen el agotamiento primario, la recuperación secundaria (usualmente inundación con agua), y recuperación terciaria (o recuperación mejorada de aceite). Un rango de métodos que están diseñados para recuperar aceite adicional que no puede producirse por métodos de recuperación primaria y secundaria incluye los métodos térmicos (inyección de vapor o combustión in situ), inundación con químicos (inyección de alcalinos, surfactantes, polímeros, y/o espuma), y métodos microbianos (usando microorganismos para recuperar aceite).

### A.0.3.5. Modelos de procesos de simulación

Las formulaciones de modelos de las ecuaciones de transporte para sistemas multicomponentes se usan para simular los diferentes procesos de recuperación.

## A.0.4. Propiedades de roca/fluido

Se describen las propiedades de la interacción roca/fluido.

### A.0.4.1. Saturación de fluido

La saturación de una fase (agua, aceite, o gas) es una fracción del espacio de poro que ésta ocupa. Comúnmente se indica por  $S$  (fracción). Para el flujo bifásico de agua y aceite, por ejemplo, las saturaciones  $S_w$  y  $S_o$  satisfacen

$$S_w + S_o = 1 \quad (\text{A.3})$$

Lo cual significa que los dos fluidos en conjunto llenan los huecos. Para el flujo trifásico de agua, aceite y gas, el hecho de que los tres fluidos en conjunto llenen el espacio de poro da

$$S_w + S_o + S_g = 1 \quad (\text{A.4})$$

Las funciones de flujo multifásico tal como la presión capilar y la permeabilidad relativa dependen fuertemente de las saturaciones.

### A.0.5. Trazador inerte

Se debe hacer un experimento de inyección de un trazador inerte para caracterizar el coeficiente de dispersión asociado con el medio poroso de interés. Dada la dependencia de éste coeficiente con la velocidad del fluido, se recomienda realizar varias pruebas a diferentes tasas de inyección.

### A.0.6. Biomasa

Es necesario un experimento en el que se inyecte una solución de biomasa sin nutrientes, (para evitar el crecimiento de los microorganismos) y a temperatura baja (para inhibir el decaimiento de los microorganismos), que permita caracterizar los coeficientes de adsorción/desorción y estimar la fracción de microorganismos adsorbidos de forma irreversible.

### A.0.7. Nutrientes

Se propone un experimento en el que se inyecte sólo el medio de cultivo de los microorganismos para verificar si el coeficiente de dispersión asociado con el medio poroso de interés es semejante al obtenido con el trazador. Dada la dependencia de éste coeficiente con la velocidad del fluido, se recomienda realizar varias pruebas a diferentes tasas de inyección. Además, este experimento permite observar si existe o no adsorción de los nutrientes en el medio poroso.

### A.0.8. Biomasa y nutrientes

Para estudiar el comportamiento conjunto de la biomasa con los nutrientes a través del medio poroso de interés, se propone un experimento de inyección simultánea de éstos. En principio, es posible obtener ajustes de los parámetros de la cinética de crecimiento en esta prueba dinámica, que podrían ser distintos de los obtenidos en condiciones estáticas.

### A.0.9. Decaimiento

La muerte de los microorganismos es una reacción irreversible de primer orden, por lo tanto puede ser expresada por

$$g_d^m = -d_m \sigma \rho_m SC \quad (\text{A.5})$$

cuando se encuentran libres en agua y

$$g^m d = -d_m \sigma \rho_m SC \quad (\text{A.6})$$

cuando se encuentran depositados en un sólido (roca) [Chang \*et al.\* \(1991\)](#).

donde  $d_m$  es la tasa de decaimiento específico,  $\sigma$  es la fracción de volumen de poro ocupado por los microorganismos depositados/volumen total y  $\rho_m SC$  es la densidad microbiana a condiciones estándar,  $[g/mL]$  o  $[lb/ft^3]$ .

### A.0.10. Crecimiento

El crecimiento de los microorganismos se supone que sigue la ecuación de Monod, la cual describe una relación entre la concentración limitante de un nutriente y la tasa de crecimiento de los microorganismos. Entonces una ecuación generalizada de Monod puede ser escrita como:

$$g_g^m = g_m \phi SC_w^m \quad (\text{A.7})$$

cuando se encuentran en estado libre en agua y

$$g_g^m = g_m \sigma \rho_m SC \quad (\text{A.8})$$

usando se encuentran depositados en un sólido (roca).

Donde  $g_m$  es la tasa de crecimiento específico,  $\sigma$  es la fracción de volumen de poro ocupado por los microorganismos depositados/volumen total y  $\rho_m SC$  es la densidad microbiana a condiciones estándar,  $[g/mL]$  o  $[lb/ft^3]$ .

### A.0.11. Taponamiento/Destaponamiento

Los fenómenos de taponamiento/destaponamiento pueden ser descritos mediante ecuaciones cinéticas [Chang et al. \(1991\)](#).

#### Taponamiento

$$g_c^m = -k_c(\phi - \sigma)c_w^m \quad (\text{A.9})$$

donde  $k_c$  es la tasa de taponamiento. Cuando el medio esta saturado entonces  $\phi = S\phi$  y por lo tanto se reescribe

$$g_c^m = -k_c S_w \phi c_w^m \quad (\text{A.10})$$

#### Destaponamiento

$$g_{dc}^m = k_d \rho_m \sigma_{irr} \quad (\text{A.11})$$

donde  $k_d$  es la tasa de destaponamiento,  $\sigma_{irr}$  es la fracción de volumen de poro reversible ocupada por los microorganismos depositados.

donde

$$\sigma = \sigma_{ir} + \sigma_i \quad (\text{A.12})$$

$\sigma$  es la fracción de volumen total de poro ocupado por los microorganismos depositados,  $\sigma_{irr}$  es la fracción de volumen de poro reversible ocupada por los microorganismos depositados,  $\sigma_i$  es la fracción de volumen de poro irreversible ocupada por los microorganismos depositados.





## Formulación Axiomática de la Modelación de Medios Continuos

En la Teoría de los Sistemas Continuos [Allen et al. \(1988\)](#), los cuerpos llenan todo el espacio que ocupan, y en cada punto del espacio físico hay una y solamente una partícula. Así, definimos como sistema continuo a un conjunto de partículas. Aún más, dicho conjunto es un subconjunto del espacio Euclidiano tridimensional, una hipótesis básica de la teoría de los sistemas continuos es que en cualquier tiempo,  $t \in (-\infty, +\infty)$ , y en cada punto,  $\underline{x} \in B(t)$ , de la región ocupada por el cuerpo, hay una y solo una partícula del cuerpo  $B$ . Sea  $\underline{X}$  una partícula y  $\underline{p}(\underline{X}, t)$  el vector de la posición que ocupa, en el espacio físico, dicha partícula en el tiempo  $t$ . Una forma, pero no la única, de identificar a la partícula,  $\underline{X}$ , es asociándole la posición que ocupa en un instante determinado. Tomaremos en particular el tiempo  $t = 0$ . En tal caso

$$\underline{p}(\underline{X}, 0) = \underline{X} \tag{B.1}$$

A las coordenadas del vector  $\underline{X} = (X_1, X_2, X_3)$ , se les llama las *coordenadas materiales* de la partícula. En este caso, las coordenadas materiales de una partícula son las coordenadas del punto del espacio físico que ocupaba la partícula en el tiempo inicial,  $t = 0$ . Desde luego, el tiempo inicial puede ser cualquier otro, si así se desea. Sea  $\mathcal{B}$  el dominio ocupado por un cuerpo en el tiempo inicial, entonces  $\underline{X} \in \mathcal{B}$  si y solamente si la partícula  $\underline{X}$  es del cuerpo. Es decir,  $\mathcal{B}$  caracteriza al cuerpo. Sin embargo, debido al movimiento, la región ocupada por el mismo cambia con el tiempo y será denotada por  $B(t)$ . Ver figura(B.1) formalmente, para cualquier  $t \in (-\infty, \infty)$ ,  $B(t)$  se define por: El vector de posición,  $\underline{p}(\underline{X}, t)$ , es función del vector (tridimensional)  $\underline{X}$  y del tiempo. Si fijamos el tiempo  $t$ ,  $\underline{p}(\underline{X}, t)$  define una transformación del espacio euclidiano  $\mathbb{R}^3$  en sí mismo. Una notación utilizada para representar esta familia de mapeos es  $\underline{p}(\cdot, t)$ . De acuerdo a la hipótesis de los sistemas continuos: *En cualquier tiempo  $t \in (-\infty, +\infty)$  y en cada punto  $\underline{x} \in B(t)$  de la región ocupada por el cuerpo hay una, y sólo una, partícula del cuerpo  $B$  para cada  $t$  fijo.* Es decir,  $\underline{p}(\cdot, t)$  es un mapeo biunívoco, por lo que existe el mapeo inverso  $\underline{p}^{-1}(\cdot, t)$  Ver Figura (B.1).

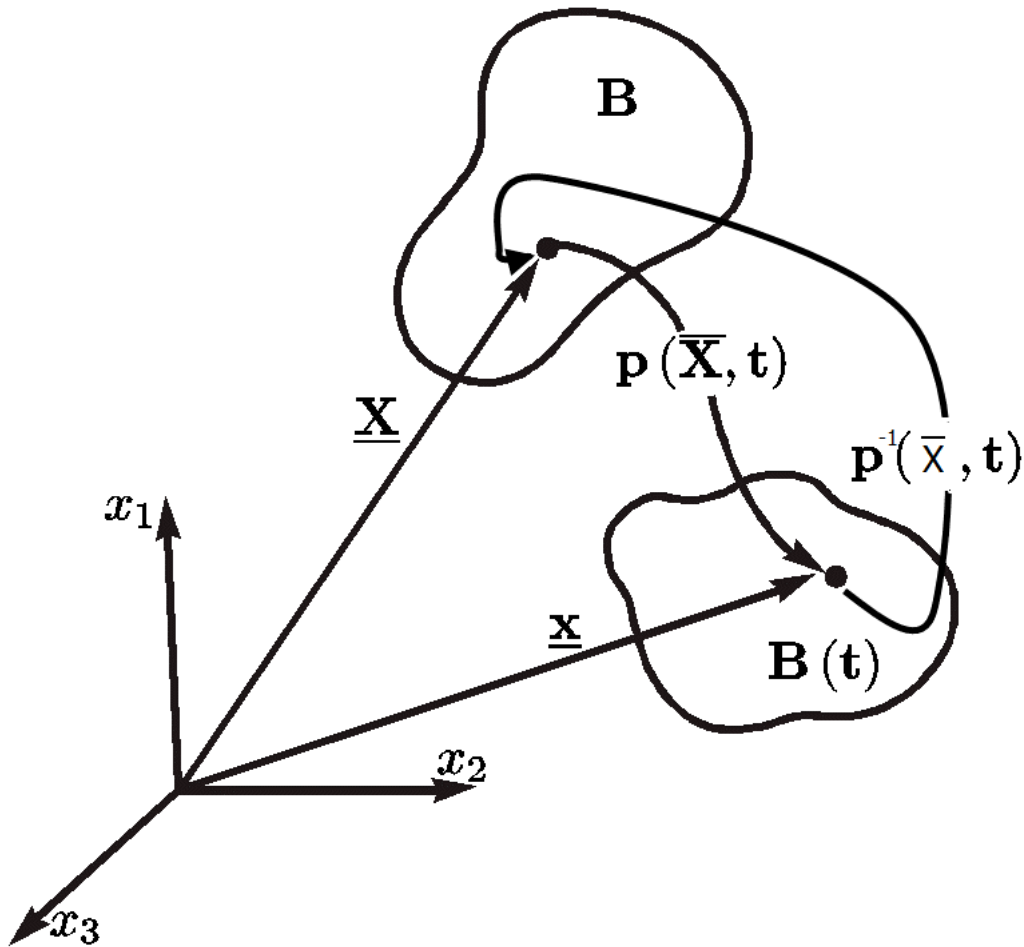


Figura B.1: Movimiento de partículas.

Si se fija la partícula,  $\underline{X}$ , en la función  $\underline{p}(\underline{X}, t)$  y se varía el tiempo  $t$  se obtiene su trayectoria. Esto permite obtener la velocidad de cualquier partícula, la cual es un concepto central en la descripción del movimiento. Ella se define como la derivada con respecto al tiempo de la posición cuando la partícula se mantiene fija. Es decir, es la derivada parcial con respecto al tiempo de la función de posición,  $\underline{p}(\underline{X}, t)$ . Por lo mismo, la velocidad como función de las coordenadas materiales de las partículas, está dada por

$$\underline{V}(\underline{X}, t) = \frac{\partial \underline{p}}{\partial t}(\underline{X}, t) \quad (\text{B.2})$$

## B.1. Propiedades intensivas

Las propiedades intensivas pueden ser funciones escalares o funciones vectoriales. Por ejemplo, la velocidad, definida por la ecuación (B.2), es una función vectorial que depende de la partícula  $\underline{X}$  y del tiempo  $t$ . Hay dos formas de representar a las propiedades intensivas: la representación Euleriana y la representación Lagrangiana. Frecuentemente, el punto de vista Lagrangiano es utilizado en el estudio de los sólidos, mientras que el Euleriano se usa más en el estudio de los fluidos. Considere una propiedad intensiva escalar, la cual en el tiempo,  $t$ , toma

el valor,  $\phi(\underline{X}, t)$ , en la partícula  $\underline{X}$ . Entonces, de esta manera se define una función  $\phi : \mathcal{B} \rightarrow \mathbb{R}^1$ , para cada  $t \in (-\infty, +\infty)$ , a la que se denomina representación Lagrangiana de la propiedad intensiva considerada. Ahora, sea  $\psi(\underline{x}, t)$  el valor que toma esa propiedad en la partícula que ocupa la posición  $\underline{x}$ , en el tiempo  $t$ . En este caso, para cada  $t \in (-\infty, +\infty)$  se define una función  $\psi : B(t) \rightarrow \mathbb{R}^1$ , a la cual se denomina representación Euleriana de la función considerada. Estas dos representaciones de una misma propiedad están relacionadas por la siguiente identidad:

$$\phi(\underline{X}, t) = \psi(\underline{p}(\underline{X}, t), t) \quad (\text{B.3})$$

Note también que aunque ambas representaciones satisfacen la ecuación (B.3), las funciones  $\phi(\underline{X}, t)$  y  $\psi(\underline{x}, t)$  no son idénticas. Sus argumentos  $\underline{X}$  y  $\underline{x}$  son vectores tridimensionales (es decir, puntos de  $\mathbb{R}^3$ ; sin embargo, si tomamos  $\underline{x} = \underline{X}$ , en general

$$\phi(\underline{X}, t) \neq \psi(\underline{X}, t) \quad (\text{B.4})$$

La expresión de la velocidad de una partícula dada por la ecuación (B.2), define a su representación Lagrangiana, por lo que utilizando la ecuación (B.3) es claro que

$$\frac{\partial p}{\partial t}(\underline{X}, t) = \underline{V}(\underline{X}, t) = \underline{v}(\underline{p}(\underline{X}, t), t) \quad (\text{B.5})$$

donde  $\underline{v}(\underline{x}, t)$  es la representación Euleriana de la velocidad. Por lo mismo

$$\underline{v}(\underline{x}, t) = \underline{V}(\underline{p}^{-1}(\underline{x}, t), t) \quad (\text{B.6})$$

esta ecuación tiene la interpretación de que la velocidad en el punto  $x$  del espacio físico, es igual a la velocidad de la partícula que pasa por dicho punto en el instante  $t$ , la ecuación (B.6), es un caso particular de la relación:

$$\psi(\underline{x}, t) = \phi(\underline{p}^{-1}(\underline{x}, t), t) \quad (\text{B.7})$$

es de validez general, la cual es otra forma de expresar la relación de la ecuación (B.3) que existe entre las dos representaciones de una misma propiedad intensiva. La derivada parcial con respecto al tiempo de la representación Lagrangiana,  $\phi(\underline{X}, t)$ , de una propiedad intensiva, de acuerdo a la definición de la derivada parcial de una función, es la tasa de cambio con respecto al tiempo que ocurre en una partícula fija. Es decir, si nos montamos en una partícula y medimos a la propiedad intensiva y luego los valores así obtenidos los derivamos con respecto al tiempo, el resultado final es  $\frac{\partial \phi(\underline{X}, t)}{\partial t}$ . En cambio, si  $\psi(\underline{X}, t)$  es la representación Euleriana de esa misma propiedad, entonces  $\frac{\partial \phi(\underline{X}, t)}{\partial t}$  es simplemente tasa de cambio con respecto al tiempo que ocurre en un punto fijo en el espacio. Tiene interés evaluar la tasa de cambio con respecto al tiempo que ocurre en una partícula fija, cuando se usa la representación Euleriana.

## B.2. Propiedades extensivas

Se consideran funciones definidas en las partículas de un cuerpo; más precisamente, funciones que hacen corresponder a cada partícula y cada tiempo un número real, o un vector del espacio Euclidiano tridimensional,  $\mathbb{R}^3$ . En ésta, en cambio, consideraremos funciones que a cada cuerpo,  $\mathcal{B}$  de un sistema continuo, y a cada tiempo,  $t$ , le asocia un número real o un vector de  $\mathbb{R}^3$ . una función de este tipo,  $E(\mathcal{B}, t)$  se le llama *propiedad extensiva* cuando está dado por una integral

$$E(\mathcal{B}, t) \equiv \int_{B(t)} \psi(\underline{x}, t) d\underline{x}. \quad (\text{B.8})$$

Observe que, en tal caso, el integrando define una función  $\psi(\underline{x}, t)$  y, por lo mismo, una *propiedad intensiva*. En particular, la función  $\psi(\underline{x}, t)$  es la representación Euleriana de esa propiedad intensiva. Además, la ecuación (B.8)

establece una correspondencia biunívoca entre las propiedades extensivas y las intensivas, porque dada la representación Euleriana,  $\psi(\underline{x}, t)$  de cualquier propiedad intensiva, su integral sobre el dominio ocupado por cualquier cuerpo, define una propiedad extensiva.

### B.3. Balance de propiedades extensivas e intensivas

Los modelos matemáticos de los sistemas continuos están constituidos por balances de propiedades extensivas. Por ejemplo, los modelos de transporte de solutos (los contaminantes transportados por corrientes superficiales o subterráneas, son un caso particular de estos procesos de transporte) se construyen haciendo el balance de la masa de soluto que hay en cualquier dominio del espacio físico.

### B.4. Ecuación de balance global

Para realizar tales balances es necesario, en primer lugar, identificar las causas por las que las propiedades extensivas pueden cambiar:

- Por producción en el interior del cuerpo; y
- Por importación (es decir, transporte ) a través de la frontera.

Esto conduce a la siguiente ecuación de balance global, de gran generalidad, para las propiedades extensivas:

$$\frac{dE(t)}{dt} = \int_{B(t)} g(\underline{x}, t) d\underline{x} + \int_{\partial B(t)} q(\underline{x}, t) d\underline{x} + \int_{\Sigma(t)} g_{\Sigma}(\underline{x}, t) d\underline{x} \quad (\text{B.9})$$

Donde  $g(\underline{x}, t)$  es la generación en el interior del cuerpo, con signo, de la propiedad extensiva correspondiente, por unidad de volumen, por unidad de tiempo. Además, en la Ecuación (B.9) se ha tomado en cuenta la posibilidad de que haya producción concentrada en la superficie  $\Sigma(t)$ , la cual está dada en esa ecuación por la última integral, donde  $g_{\Sigma}(\underline{x}, t)$  es la producción por unidad de área. Por otra parte,  $q(\underline{x}, t)$  es lo que se importa, o transporta, hacia e interior del cuerpo a través de la frontera del cuerpo  $\partial B(t)$ ; en otras palabras, es el flujo de la propiedad extensiva a través de la frontera del cuerpo, por unidad de área, por unidad de tiempo. Puede demostrarse, con base en hipótesis válidas en condiciones muy generales, que para cada tiempo,  $t$ , existe un campo vectorial  $\underline{\tau}(\underline{x}, t)$  tal que

$$q(\underline{x}, t) \equiv \underline{\tau}(\underline{x}, t) \cdot \underline{n}(\underline{x}, t) \quad (\text{B.10})$$

donde  $\underline{n}(\underline{x}, t)$  es normal exterior a  $\partial B(t)$ . En vista de esta relación, la ecuación de balance se puede escribir como

$$\frac{dE(t)}{dt} = \int_{B(t)} g(\underline{x}, t) d\underline{x} + \int_{\partial B(t)} \underline{\tau}(\underline{x}, t) \cdot \underline{n}(\underline{x}, t) d\underline{x} + \int_{\Sigma(t)} g_{\Sigma}(\underline{x}, t) d\underline{x} \quad (\text{B.11})$$

La relación ecuación (B.11) se conoce con el nombre de **Ecuación de Balance Global**. A la función  $g(\underline{x}, t)$  se le llama *generación interna* y al campo vectorial  $\underline{\tau}(\underline{x}, t)$ , el *campo de flujo*. A la ecuación (B.11) se le denomina **Ecuación General de Balance Global** y es la ecuación básica de los balances de los *sistemas continuos*.

### B.5. Condiciones de balance local

Los modelos de los sistemas continuos [Herrera y Pinder \(2012\)](#) están constituidos por las ecuaciones de balance correspondientes a una colección de propiedades extensivas. Esto es posible porque las Ecuaciones de Balance Global son equivalentes a las llamadas Condiciones de Balance Local, las cuales se expresan en términos de las propiedades intensivas correspondientes. Las Condiciones de Balance Local son de dos clases: Las Ecuaciones Diferenciales de Balance Local. Las ecuaciones diferenciales parciales, que se deben satisfacer en cada punto del espacio ocupado por el sistema continuo.

## B.6. Ecuaciones de balance local

**Teorema 1:** *Considere un sistema continuo. Entonces, la Ecuación de Balance Global, ecuación (B.11), se satisface para todo cuerpo del sistema continuo si y solamente si, se cumplen las siguientes condiciones:*

- La ecuación diferencial

$$\frac{\partial \psi(\underline{x}, t)}{\partial t} + \nabla \cdot (\underline{v}\psi(\underline{x}, t)) = \nabla \cdot \underline{\tau}(\underline{x}, t) + g(\underline{x}, t) \quad (\text{B.12})$$

vale en todo punto  $\underline{x} \in \mathbb{R}^3$ , de la región ocupada por el sistema. Desde luego, el caso más general que se estudiará se refiere a situaciones dinámicas; es decir, aquéllas en que las propiedades intensivas cambian con el tiempo. Sin embargo, los estados estacionarios de los sistemas continuos son de sumo interés. Por estado estacionario se entiende uno en que las propiedades intensivas son independientes del tiempo. En los estados estacionarios, En este caso,  $\frac{\partial \psi}{\partial t} \equiv 0$ . Por lo mismo, para los estados estacionarios, la Ecuación de balance Local se reduce a

$$\nabla \cdot (\underline{v}\psi) = \nabla \cdot \underline{\tau} + g \quad (\text{B.13})$$

que vale en todo punto  $\underline{x} \in \mathbb{R}^3$ .

### B.6.1. Ejemplos de condiciones de balance local

Una de las aplicaciones más sencillas de las Condiciones de Balance Local, es para formular restricciones en el movimiento. Aquí ilustramos este tipo de aplicaciones formulando condiciones que se deben cumplir localmente cuando un fluido es incompresible. La afirmación de que un fluido es incompresible significa que todo cuerpo conserva el volumen de fluido en su movimiento. Entonces se considerarán dos casos: el de un **fluido libre** y el de un "fluido en medio poroso". El primer caso, el fluido llena completamente el espacio físico que ocupa el cuerpo, por lo que el volumen del fluido es igual al volumen del dominio que ocupa el cuerpo. Así

$$V_f(t) = \int_{B(t)} d\underline{x} \quad (\text{B.14})$$

aquí  $V_f(t)$  es el volumen del fluido y  $B(t)$  es el dominio del espacio físico (es decir, de  $\mathbb{R}^3$ ) ocupado. Observe que una forma más explícita de la ecuación anterior es

$$V_f(t) = \int_{B(t)} 1 d\underline{x} \quad (\text{B.15})$$

porque la integral que aparece en la ecuación (B.17) el integrando es la función idénticamente 1. Comparando esta ecuación con la ecuación (B.8), vemos que el volumen del fluido es una *propiedad extensiva* y que la *propiedad intensiva* que le corresponde es  $\psi = 1$ .

Además, la hipótesis de incompresibilidad implica

$$\frac{dV_f}{dt}(t) = 0 \quad (\text{B.16})$$

esta es el Balance Global de la ecuación (B.11), con  $g \equiv 0$  y  $\tau \equiv 0$ , el cual a su vez es equivalente a las ecuaciones (B.13) y (B.14). Tomando en cuenta además que  $\psi = 1$ , la ecuación (B.13) se reduce a

$$\nabla \cdot \underline{v} = 0 \quad (\text{B.17})$$

Esta es la bien conocida condición de incompresibilidad para un *fluido libre*. El caso en que el fluido se encuentra en un *medio poroso*, es bastante diferente. Un medio poroso es un material sólido que tiene huecos distribuidos en

toda su extensión ver figura (B.2). Cuando los poros están llenos de un fluido, se dice que el medio poroso está *saturado*, situación que tiene mucho interés y también la más estudiada. En muchos de los casos que ocurren en las aplicaciones, el fluido es agua o petróleo. A la fracción del volumen del sistema, constituido por la matriz sólida y los huecos, se le llama *porosidad* y se le representa por  $\varepsilon$ . Así

$$\varepsilon = (\underline{x}, t) = \lim_{Vol \rightarrow 0} \frac{\text{Volumen de huecos}}{\text{Volumen total}} \quad (\text{B.18})$$

Aquí hemos escrito  $\varepsilon(\underline{x}, t)$  para enfatizar que la porosidad generalmente es función tanto de la posición como del tiempo. Las variaciones con la posición pueden ser debidas, por ejemplo, a heterogeneidad del medio y los cambios

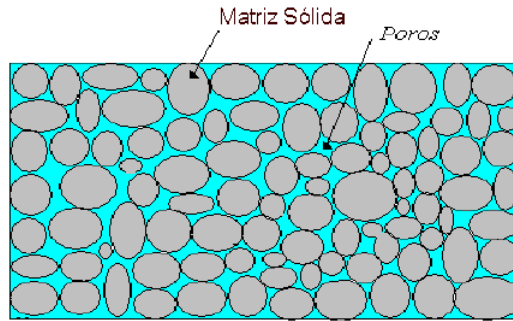


Figura B.2: Medio poroso.

con el tiempo a su elasticidad; es decir, los cambios de presión del fluido originan esfuerzos en los poros que los dilatan o los encogen. Cuando el medio está *saturado*, el volumen del fluido ( $V_f$ ) es igual al volumen de los huecos del dominio del espacio físico que ocupa. Así,

$$V_f(t) = \int_{B(t)} \varepsilon(\underline{x}, t) d\underline{x} \quad (\text{B.19})$$

en vista de esta ecuación, la propiedad intensiva asociada al volumen de fluido es la porosidad,  $\varepsilon(\underline{x}, t)$ , or lo que la condición de incompresibilidad del fluido contenido en un medio poroso está dado por la ecuación diferencial

$$\frac{\partial \varepsilon}{\partial t} + \nabla \cdot (\underline{v} \varepsilon) = 0 \quad (\text{B.20})$$

Que la divergencia de la velocidad sea igual a cero, ecuación (B.20), como condición para que un fluido en su movimiento conserve su volumen, es ampliamente conocida, como condición para la conservación del volumen de los cuerpos de fluidos contenidos en un *medio poroso*. Finalmente, debe observarse que cualquier fluido incompresible satisface la ecuación (B.20) cuando se mueve en el espacio libre. Cuando un fluido efectúa un movimiento en el que conserva su volumen, al movimiento se le llama *isocórico*. Es oportuno mencionar que si es bien cierto que cuando un fluido tiene la propiedad de ser incompresible, todos sus movimientos son isocóricos, lo inverso no es cierto: *un fluido compresible en ocasiones puede efectuar movimientos isocóricos*.

## B.7. Condiciones iniciales y de frontera

Las soluciones al sistema de ecuaciones descritas anteriormente constituyen un modelo completo cuando se prescriben ciertas condiciones iniciales y de frontera. Las condiciones iniciales comprenden la especificación de las

variables dependientes en un momento determinado, por lo general referido como el tiempo inicial,  $t=0$ . En cuanto a las condiciones de frontera, generalmente cuando un fluido viscoso está en contacto con un sólido, las partículas de fluido se adhieren a la superficie del cuerpo. Debido a esta adhesión, la velocidad del fluido en relación a la superficie del sólido es cero. En particular, si el cuerpo está en reposo, entonces la velocidad del fluido es cero; puesto que la velocidad es un vector, este hecho implica que tanto sus componentes normal y tangencial se desvanecen [Herrera y Pinder \(2012\)](#).

- **Condiciones iniciales:** El valor inicial de la concentración se prescribe:

$$c(\underline{x}, 0) = c_0(\underline{x}); \quad \forall \underline{x} \in \Omega. \quad (\text{B.21})$$

aquí la función  $c_0(\underline{x})$  es la concentración inicial, la cual se asume que es dato del problema. Observe que aunque el tiempo inicial que se ha tomado aquí es  $t=0$ , cualquier otro momento podría ser elegido.

- **Condiciones de frontera tipo Robin:** La forma más general de condiciones de frontera que se consideran son conocidas como condiciones de frontera tipo Robin. Sea  $\alpha$  y  $\beta$  números tales que  $\alpha^2 + \beta^2 = 1$ . Entonces

$$\alpha \frac{\partial c}{\partial n}(\underline{x}, t) + \beta c(\underline{x}, t) = \gamma(\underline{x}, t); \quad \forall \underline{x} \in \partial\Omega \quad y \quad t > 0 \quad (\text{B.22})$$

donde la función  $\gamma(\underline{x}, t)$  es un dato del problema. La condición de frontera tipo Robin, en su forma más general admite la posibilidad que  $\alpha$  y  $\beta$  cambiar de punto a punto de vez en cuando, esto es que  $\alpha$  y  $\beta$  son funciones de posición y del tiempo. De este modo también se observa que la exigencia de  $\alpha^2 + \beta^2 = 1$  es una simple manera de garantizar que  $\alpha$  y  $\beta$  no son ambas simultáneamente igual a cero. Dos importantes casos particulares de condiciones de frontera tipo Robin deben mencionarse: Condiciones de frontera tipo Dirichlet y Neumann. Corresponden a las opciones  $\beta = 1$  (es decir,  $\alpha = 0$ ) y  $\alpha = 1$  (es decir  $\beta = 0$ ), respectivamente.

- **Condiciones de frontera tipo Dirichlet.** Estas son:

$$c(\underline{x}, t) = \gamma(\underline{x}, t); \quad \forall \underline{x} \in \partial\Omega \quad y \quad t > 0. \quad (\text{B.23})$$

- **Condiciones de frontera tipo Neumann.** Estas son:

$$\frac{\partial c}{\partial n}(\underline{x}, t) = \gamma(\underline{x}, t); \quad \forall \underline{x} \in \partial\Omega \quad y \quad t > 0. \quad (\text{B.24})$$

Por lo tanto, los problemas más generales bien planteados que serán considerados cuando se modela transporte difusivo problemas-límite-valor inicial con las condiciones de frontera Robin.





## Elementos de análisis funcional

Para presentar el método de elementos finitos en una formulación general, deberemos utilizar los espacios de funciones. Esta sección está dedicada al desarrollo de los espacios de la función que son ligeramente más general que los espacios de funciones continuas suave por tramos continuos derivados. Establecemos la pequeña fracción de estos espacios que es suficiente para desarrollar la base del método de elemento finitos.

### C.1. Teoría de integración de espacios de Lebesgue

Asumimos que  $\Omega$  es un subconjunto abierto de  $\mathbb{R}^d$ ,  $1 \leq d \leq 3$  suaves a tramos en la frontera. Para una función de valor real  $v$  sobre  $\Omega$ , usamos la notación

$$\int_{\Omega} v(x) dx \tag{C.1}$$

para denotar la integral de  $f$  en el sentido de Lebesgue (Rudin, 1987). Para  $1 \leq q \leq \infty$ , define

$$\|v\|_{L^q(\Omega)} = \left( \int_{\Omega} |v(x)|^q dx \right)^{\frac{1}{q}}. \tag{C.2}$$

Para  $q = \infty$ , el conjunto

$$\|v\|_{L^\infty(\Omega)} = \text{ess sup } \{|v(x)| : x \in \Omega\} \tag{C.3}$$

donde *ess sup* denota el *supremo esencial*. Ahora, para  $1 \leq q \leq \infty$ , definimos los espacios de Lebesgue

$$L^q(\Omega) = \left\{ v : v \text{ es definido sobre } \Omega \text{ y } \|v\|_{L^q(\Omega)} < \infty \right\} \tag{C.4}$$

Para  $q = 2$ , por ejemplo  $L^2(\Omega)$ , que consiste en todas las funciones cuadrado integrable (en el sentido de Lebesgue). Para evitar diferencias triviales, identificamos dos funciones,  $u$  y  $v$  siempre que  $\|u - v\|_{L^q(\Omega)} = 0$ , es decir,  $u(x) = v(x)$  para  $x \in \Omega$ , excepto un conjunto de medida cero.

Dado un espacio lineal (vector)  $V$ , una norma en  $V$ ,  $\|\cdot\|$ , es una función de  $V$ , en  $\mathbb{R}$ , tal que

- $\|v\| \geq 0 \quad \forall v \in V; \quad \|v\| = 0 \text{ si y solo si } v = 0.$
- $\|cv\| = |c| \|v\| \quad \forall c \in \mathbb{R}, \quad u \in V.$
- $\|u + v\| \leq \|u\| + \|v\| \quad \forall u, v \in V. \quad (\text{desigualdad del triangulo})$

Un espacio normal  $V$  con norma  $\|\cdot\|$ , es llamado espacio lineal normado.

$V$  es considerado completo si cada sucesión de Cauchy  $\{v_i\}$  en  $V$  tiene un límite  $v$  que es un elemento de  $V$ . La sucesión de Cauchy  $\{v_i\}$  nos dice que  $\|v_i - v_j\| \rightarrow 0$  como  $i, j \rightarrow \infty$ , y la integral dice que  $\|v_i - v\| \rightarrow 0$  así  $i \rightarrow \infty$ .

Un espacio lineal normado  $(V, \|\cdot\|)$  es llamado espacio de Banach si este es completo con respecto a la norma  $\|\cdot\|$ . Para  $1 \leq q \leq \infty$ , el espacio  $L^q(\Omega)$  es un espacio de Banach.

### C.1.1. Espacios de Sobolev

Ahora usamos derivadas débiles generalizar los espacios de Lebesgue.

Para  $r = 1, 2, \dots$  y  $v \in L^1_{loc}(\Omega)$  asumimos que la derivada débil  $D^\alpha v$  existe para toda  $|\alpha| \leq r$ . Definimos la norma de Sobolev

$$\|v\|_{W^{r,q}(\Omega)} = \left( \sum_{|\alpha| \leq r} \|D^\alpha v\|_{L^q(\Omega)}^q \right)^{\frac{1}{q}} \quad (\text{C.5})$$

Si  $1 \leq q < \infty$ .

Para  $q = \infty$ , define

$$\|v\|_{W^{r,\infty}(\Omega)} = \max_{|\alpha| \leq r} \|D^\alpha v\|_{L^\infty(\Omega)} \quad (\text{C.6})$$

Los *Espacios de Sobolev* son definidos por

$$W^{r,q}(\Omega) = \left\{ v \in L^1_{loc}(\Omega) : \|v\|_{W^{r,q}(\Omega)} < \infty \right\}, \quad 1 \leq q \leq \infty \quad (\text{C.7})$$

podemos ver que  $\|v\|_{W^{r,q}(\Omega)}$  es una norma; por otra parte, los espacios de Sobolev  $W^{r,q}(\Omega)$  es un espacio de Banach.

Denotamos por  $W_0^{r,q}(\Omega)$  la completación  $D(\Omega)$  con respecto a la norma  $\|\cdot\|_{W_0^{r,q}(\Omega)}$ . Para  $\Omega \subset \mathbb{R}^d$  con bordes suaves en la frontera y  $v \in W^{1,q}(\Omega)$ , la restricción de la frontera  $\Gamma$ ,  $v|_\Gamma$ , puede interpretarse como función en  $L^q(\Gamma)$ ,  $1 \leq q \leq \infty$ . Esto no afirma que los valores suaves de  $v$  en  $\Gamma$  tienen sentido. Para  $q = 2$ , por ejemplo  $v|_\Gamma$  es solamente cuadrado integrable en  $\Gamma$ . Usando esta propiedad, el espacio  $W_0^{r,q}(\Omega)$  puede ser caracterizado como

$$W_0^{r,q}(\Omega) = \left\{ v \in W_0^{r,q}(\Omega) : D^\alpha v|_\Gamma = 0 \text{ en } L^2(\Gamma), \quad |\alpha| < r \right\} \quad (\text{C.8})$$

Para las aplicaciones posteriores, se utilizarán las seminormas:

$$|v|_{W^{r,q}(\Omega)} = \left( \sum_{|\alpha|=r} \|D^\alpha v\|_{L^q(\Omega)}^q \right)^{\frac{1}{q}}, \quad 1 \leq q \leq \infty, \quad |v|_{W^{r,\infty}(\Omega)} = \max_{|\alpha|=r} \|D^\alpha v\|_{L^\infty(\Omega)} \quad (\text{C.9})$$

Por lo tanto para  $q = 2$  vamos a utilizar los símbolos

$$W^r(\Omega) = W^{r,2}(\Omega), \quad W_0^r(\Omega) = W_0^{r,2}(\Omega) \quad r = 1, 2, \dots \quad (\text{C.10})$$

Es decir, las funciones en  $H^r(\Omega)$ , junto con sus derivadas  $D^\alpha v$  de orden  $|\alpha| \leq r$ , son cuadrado integrable en  $\Omega$ . Note que  $H^0(\Omega) = L^2(\Omega)$ .

Los espacios de Sobolev  $W^{r,q}(\Omega)$ , una serie de propiedades importantes.

Teniendo en cuenta los índices que definen estos espacios, es natural que hay relaciones de inclusión para proporcionar algún tipo de ordenación entre ellos. Tenemos una lista de un par de las relaciones de inclusión. Para enteros no negativos  $r$  y  $k$  tal que  $r \leq k$ , se cumple que

$$W^{k,q}(\Omega) \subset W^{r,q}(\Omega), \quad 1 \leq q \leq \infty \quad (\text{C.11})$$

Ademas, cuando  $\Omega$  esta en la frontera

$$W^{r,q'}(\Omega) \subset W^{r,q}(\Omega), \quad 1 \leq q \leq q' \leq \infty \quad (\text{C.12})$$

Para  $r = 1, 2, \dots$  [Chen \(2005\)](#)

## C.2. Espacios de Hilbert

**Proposición:**  $\|v\| := \sqrt{(v, v)}$  define una norma en el espacio de producto interno  $(V, (\cdot, \cdot))$ .

Dado un espacio de producto interno  $(V, (\cdot, \cdot))$ , hay una norma asociada definida en  $V$ , es decir,  $\|v\| := \sqrt{(v, v)}$ .

De este modo, un espacio de producto interior puede convertirse en un espacio lineal normado.

**Definición.** Sea  $(V, (\cdot, \cdot))$ , sea un espacio de producto interno. Si al espacio lineal normado le asociamos  $(V, (\|\cdot\|))$  está completo, entonces  $(V, (\cdot, \cdot))$  es llamado un **Espacio de Hilbert**.

## C.3. El Teorema de Lax-Milgram

Queremos demostrar la existencia y unicidad de la solución del problema variacional (no simétrico): Hallar  $u \in V$  tal que, [Brenner y Scott \(2008\)](#).

$$a(u, v) = L(v) \quad \forall v \in V \quad (\text{C.13})$$

donde  $V$  es un espacio de Hilbert,  $L \in V'$  y  $a(\cdot, \cdot)$  es continuo, coercitiva bilineal que no es necesariamente simétrica.

El teorema de Lax-Milgram garantiza la existencia y unicidad de la solución a [C.13](#).

## C.4. Teorema: (Lax-Milgram)

Dado un espacio de Hilbert  $(V, (\cdot, \cdot))$ , una forma bilineal continua y coercitiva  $a(\cdot, \cdot)$ , Y una función lineal continua  $L \in V'$ , existe una única  $u \in V$ , tal que

$$a(u, v) = L(v) \quad \forall v \in V. \quad (\text{C.14})$$

*Prueba.* para cualquier  $u \in V$ , define una función  $Au$  por  $Au(v) = a(\cdot, v) \forall v \in V$ .  $Au$  es lineal desde

$$\begin{aligned} Au(\alpha v_1 + \beta v_2) &= a(u, \alpha v_1 + \beta v_2) \\ &= \alpha a(u, v_1) + \beta a(u, v_2) \\ &= \alpha Au(v_1) + \beta Au(v_2) \quad v_1, v_2 \in V, \alpha, \beta \in \mathbb{R}. \end{aligned} \quad (\text{C.15})$$

$Au$  es también continua ya que, para todo  $v \in V$ ,

$$|Au(v)| = |a(u, v)| \leq C \|u\| \|v\|, \quad (\text{C.16})$$

Aquí  $C$  es la constante de la definición de continuidad para,  $a(\cdot, \cdot)$ . Por lo tanto,

$$\|Au\|_{V'} = \sup_{v \neq 0} \frac{|Au(v)|}{\|v\|} \leq C \|u\| < \infty \quad (\text{C.17})$$

Así,  $Au \in V'$ .

## C.5. Teorema de la representación de Riesz

Dado  $u \in H$ , recordar que un funcional lineal continuo  $L_u$  puede ser definido en  $H$  por

$$L_u(v) = (u, v) \quad (\text{C.18})$$

el siguiente teorema demuestra que lo contrario también es cierto.

**Teorema. (Teorema de la Representación de Riesz).** Cualquier funcional  $L$  lineal continuo en un espacio de Hilbert  $H$  puede representarse únicamente como

$$L_u(v) = (u, v) \quad (\text{C.19})$$

para algún  $u \in H$ . Además, tenemos

$$\|L\|_{H'} = \|u\|_H. \quad (\text{C.20})$$

**Prueba.** La unicidad se deriva de la no degeneración del producto interno. Pues si  $u_1$  y  $u_2$  son dos de tales soluciones, tendríamos

$$\begin{aligned} 0 &= L(u_1 - u_2) - L(u_1 - u_2) \\ &= (u_1, u_1 - u_2) - (u_2, u_1 - u_2) \\ &= (u_1 - u_2, u_1 - u_2) \end{aligned} \quad (\text{C.21})$$

lo que implica  $u_1 = u_2$ , [Brenner y Scott \(2008\)](#).

## Generalización del Tensor de dispersión Hidrodinámica

Se supone que el flujo sigue la Ley de Fick, es decir [Bear \(1988\)](#),

$$\underline{\tau}^\eta = \phi S \underline{D}^\eta \cdot \nabla c^\eta \quad \eta = w \quad (\text{D.1})$$

En este caso  $D^\eta$  es el tensor de dispersión hidrodinámica para la componente  $\eta$  en la fase agua  $w$  y se expresa en términos de la velocidad de Darcy como:

$$\phi S_w (D_w^\eta)_{ij} = \alpha_T^\eta |\underline{u}_w| \delta_{ij} + (\alpha_L^\eta - \alpha_T^\eta) \frac{(u_w)_i (u_w)_j}{|\underline{u}_w|} + \frac{\phi S_w D_w^{*\eta}}{\tau} \delta_{ij} \quad (\text{D.2})$$

donde  $\tau$  es la tortuosidad del medio poroso ( $\tau > 1$ ),  $\alpha_T$  es el coeficiente de dispersión mecánica transversal,  $\alpha_L$  es el coeficiente de dispersión mecánica longitudinal y  $D^{*\eta}$  es el coeficiente de difusión molecular de la componente  $\eta$  en fase agua  $w$ .

Para el caso que estamos trabajando vamos a definir el tensor de la siguiente manera:

$$\underline{D} = \begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{pmatrix}$$

Ahora basados en la definición anterior podemos escribir el desarrollo de cada componente del tensor de la forma siguiente:

$$\begin{aligned}
D_{xx} &= \phi S_w(D_w^\eta)_{xx} = \alpha_T^\eta |\underline{u}_w| + (\alpha_L^\eta - \alpha_T^\eta) \frac{(u_w)_x (u_w)_x}{|\underline{u}_w|} + \frac{\phi S_w D_w^{*\eta}}{\tau} \\
D_{xy} &= \phi S_w(D_w^\eta)_{xy} = (\alpha_L^\eta - \alpha_T^\eta) \frac{(u_w)_x (u_w)_y}{|\underline{u}_w|} \\
D_{xz} &= \phi S_w(D_w^\eta)_{xz} = (\alpha_L^\eta - \alpha_T^\eta) \frac{(u_w)_x (u_w)_z}{|\underline{u}_w|} \\
D_{yx} &= \phi S_w(D_w^\eta)_{yx} = (\alpha_L^\eta - \alpha_T^\eta) \frac{(u_w)_y (u_w)_x}{|\underline{u}_w|} \\
D_{yy} &= \phi S_w(D_w^\eta)_{yy} = \alpha_T^\eta |\underline{u}_w| + (\alpha_L^\eta - \alpha_T^\eta) \frac{(u_w)_y (u_w)_y}{|\underline{u}_w|} + \frac{\phi S_w D_w^{*\eta}}{\tau} \\
D_{yz} &= \phi S_w(D_w^\eta)_{yz} = (\alpha_L^\eta - \alpha_T^\eta) \frac{(u_w)_y (u_w)_z}{|\underline{u}_w|} \\
D_{zx} &= \phi S_w(D_w^\eta)_{zx} = (\alpha_L^\eta - \alpha_T^\eta) \frac{(u_w)_z (u_w)_x}{|\underline{u}_w|} \\
D_{zy} &= \phi S_w(D_w^\eta)_{zy} = (\alpha_L^\eta - \alpha_T^\eta) \frac{(u_w)_z (u_w)_y}{|\underline{u}_w|} \\
D_{zz} &= \phi S_w(D_w^\eta)_{zz} = \alpha_T^\eta |\underline{u}_w| + (\alpha_L^\eta - \alpha_T^\eta) \frac{(u_w)_z (u_w)_z}{|\underline{u}_w|} + \frac{\phi S_w D_w^{*\eta}}{\tau}
\end{aligned}$$

O de forma mas compacta:

$$\begin{aligned}
D_{xx} &= \phi S_w(D_w^\eta)_{xx} = \alpha_T^\eta |\underline{u}_w| + (\alpha_L^\eta - \alpha_T^\eta) \frac{(u_w)_x (u_w)_x}{|\underline{u}_w|} + \frac{\phi S_w D_w^{*\eta}}{\tau} \\
D_{xy} &= D_{yx} = \phi S_w(D_w^\eta)_{xy} = (\alpha_L^\eta - \alpha_T^\eta) \frac{(u_w)_x (u_w)_y}{|\underline{u}_w|} \\
D_{xz} &= D_{zx} = \phi S_w(D_w^\eta)_{zx} = (\alpha_L^\eta - \alpha_T^\eta) \frac{(u_w)_z (u_w)_x}{|\underline{u}_w|} \\
D_{yy} &= \phi S_w(D_w^\eta)_{yy} = \alpha_T^\eta |\underline{u}_w| + (\alpha_L^\eta - \alpha_T^\eta) \frac{(u_w)_y (u_w)_y}{|\underline{u}_w|} + \frac{\phi S_w D_w^{*\eta}}{\tau} \\
D_{yz} &= D_{zy} = \phi S_w(D_w^\eta)_{zy} = (\alpha_L^\eta - \alpha_T^\eta) \frac{(u_w)_z (u_w)_y}{|\underline{u}_w|} \\
D_{zz} &= \phi S_w(D_w^\eta)_{zz} = \alpha_T^\eta |\underline{u}_w| + (\alpha_L^\eta - \alpha_T^\eta) \frac{(u_w)_z (u_w)_z}{|\underline{u}_w|} + \frac{\phi S_w D_w^{*\eta}}{\tau}
\end{aligned}$$

Se aplicó la definición de la delta de Kronecker para el desarrollo de tensor de dispersión hidrodinámica. Es una función de dos variables  $\delta : \mathbb{Z} \times \mathbb{Z} \rightarrow \{0, 1\}$  se define mediante la regla:

$$\delta_{i,j} = \begin{cases} 1, & \text{si } i = j, \\ 0, & \text{si } i \neq j, \end{cases}$$

y se llama el *símbolo de Kronecker* o la *delta de Kronecker*.





## Esquema de Diferencias Finitas en el Tiempo

Para la solución del problema compuesto por las ecuaciones (5.2) y (5.3) se pueden emplear diferentes esquemas en diferencias finitas para la discretización en el tiempo. Estos esquemas se conocen en general como esquemas Theta.

### E.0.1. Esquema Theta ( $\theta$ ) general

Si integramos la ecuación (5.2) expresando el operador de la forma (5.7) con respecto al tiempo en el intervalo  $[t_n, t_{n+1}]$  se tiene que:

$$c^{n+1} - c^n + \int_{t_n}^{t_{n+1}} [\mathcal{L}_{\underline{x},t}c - f_{\Omega}] dt = 0 \quad (\text{E.1})$$

Al desarrollar el segundo término bajo la integral como una serie de Taylor respecto a un punto intermedio  $t^*$  en el intervalo  $[t_n, t_{n+1}]$  se obtiene:

$$[\mathcal{L}_{\underline{x},t}c - f_{\Omega}] = [\mathcal{L}_{\underline{x},t}c - f_{\Omega}]_{t^*} + \frac{\partial}{\partial t} [\mathcal{L}_{\underline{x},t}c - f_{\Omega}]_{t^*} (t - t^*) + 0(t - t^*)^2 \quad (\text{E.2})$$

entonces la integral resulta

$$\int_{t_n}^{t_{n+1}} [\mathcal{L}_{\underline{x},t}c - f_{\Omega}] dt = [\mathcal{L}_{\underline{x},t}c - f_{\Omega}]_{t^*} k + 0(k^2) \quad (\text{E.3})$$

donde  $k = t_{n+1} - t_n$  es el paso del tiempo. al sustituir la ecuación (E.3) en la ecuación (E.1) se obtiene:

$$\frac{c^{n+1} - c^n}{k} = [t_{n+1} - t_n]_{t^*} + 0(k) = 0 \quad (\text{E.4})$$

El punto  $t^*$  se puede expresar como  $t_n + \theta k$ , donde  $0 \leq \theta \leq 1$ , entonces el segundo miembro de la ecuación (E.4) se puede aproximar mediante una interpolación lineal de la siguiente manera:

$$[\mathcal{L}_{\underline{x},t}c - f_{\Omega}] \approx \{(1 - \theta)[\mathcal{L}_{\underline{x},t}c - f_{\Omega}]_{t_n} + \theta[\mathcal{L}_{\underline{x},t}c - f_{\Omega}]_{t_{n+1}}\} \quad (\text{E.5})$$

Finalmente se obtiene la siguiente discretización con respecto al tiempo de la ecuación (5.2), la cual se conoce como esquema Theta:

$$\frac{c^{n+1} - c^n}{k} + \{(1 - \theta)[\mathcal{L}_{\underline{x},t}c - f_{\Omega}]^n + \theta[\mathcal{L}_{\underline{x},t}c - f_{\Omega}]^{n+1}\} + 0(k) = 0 \quad (\text{E.6})$$

donde los superíndices se refieren a pasos en el tiempo, es decir

$$[\mathcal{L}_{\underline{x},t}c - f_{\Omega}]^n = [\mathcal{L}_{\underline{x},t}c - f_{\Omega}]_{t_n} \quad \text{y} \quad c^n = c(t_n).$$

### E.0.2. Consideremos tres casos para Theta

1. Para  $\theta = 0$ , es decir con respecto al punto  $t_n$ , se obtiene un esquema en diferencias finitas hacia adelante en el tiempo, también conocido como esquema completamente explícito ya que el paso  $n + 1$  se obtiene de una expresión algebraica de los términos del paso anterior  $n$ .

$$\frac{c^{n+1} - c^n}{k} + [\mathcal{L}_{\underline{x},t}c - f_\Omega]^n + 0(k) = 0 \quad (\text{E.7})$$

$$\frac{c^{n+1}}{k} = \frac{c^n}{k} - [\mathcal{L}_{\underline{x},t}c - f_\Omega] \quad (\text{E.8})$$

Es un esquema simple, pero se ha demostrado que es condicionalmente estable cuando  $k \leq \frac{1}{2}(\Delta x)^2$ .

2. Para  $\theta = 1$ , es decir con respecto al punto  $t_{n+1}$ , se obtiene un esquema en diferencias hacia atrás en el tiempo, también conocido como esquema completamente implícito:

$$\frac{c^{n+1} - c^n}{k} + [\mathcal{L}_{\underline{x},t}c - f_\Omega]^{n+1} + 0(k) = 0 \quad (\text{E.9})$$

$$\frac{c^{n+1}}{k} + [\mathcal{L}_{\underline{x},t}c - f_\Omega]^{n+1} \cong \frac{c^n}{k} \quad (\text{E.10})$$

3. Para  $\theta = \frac{1}{2}$ , es decir con respecto al punto al punto  $t_n + \frac{1}{2}$ , se obtiene un esquema en diferencias finitas centrales en el tiempo, también conocido como esquema de Crank-Nicolson:

$$\frac{c^{n+1} - c^n}{k} + \frac{1}{2} \{ [\mathcal{L}_{\underline{x},t}c - f_\Omega]^{n+1} + [\mathcal{L}_{\underline{x},t}c - f_\Omega]^n \} + 0(k^2) = 0 \quad (\text{E.11})$$

$$\frac{c^{n+1}}{k} + \frac{1}{2} [\mathcal{L}_{\underline{x},t}c - f_\Omega]^{n+1} \cong \frac{c^n}{k} - \frac{1}{2} [\mathcal{L}_{\underline{x},t}c - f_\Omega]^n \quad (\text{E.12})$$

Es un método incondicionalmente estable y es más preciso que los métodos anteriores.

### E.0.3. Ecuación parabólica semi-descretizada empleando el esquema de diferencias finitas hacia atrás

La formulación de las diferencias finitas hacia atrás en el tiempo esquema implícito con  $\theta = 1$  es  $\frac{c^{n+1}}{k} + [\mathcal{L}_{\underline{x},t}c - f_\Omega]^{n+1} \cong \frac{c^n}{k}$ , aplicando esta ecuación al problema (5.2) y con la ecuación (5.5) entonces tenemos:

$$\frac{c^{n+1}}{k} + [-\nabla \cdot (\underline{D}^{n+1} \cdot \nabla c^{n+1}) + \nabla \cdot (\underline{u}^{n+1} c^{n+1}) + r^{n+1} c^{n+1} - f^{n+1}] \cong \frac{c^n}{k} \quad (\text{E.13})$$

entonces

$$-\nabla \cdot (\underline{D}^{n+1} \cdot \nabla c^{n+1}) + \nabla \cdot (\underline{u}^{n+1} c^{n+1}) + (r^{n+1} \frac{1}{n} c^{n+1}) \cong \frac{c^n}{k} + f^{n+1} \quad (\text{E.14})$$

de aquí tenemos

$$-\nabla \cdot (\alpha \cdot \nabla v) + \nabla \cdot (\beta v) + \gamma v = \phi \quad (\text{E.15})$$

con condiciones de frontera:  $-\nabla \cdot (\alpha \cdot \nabla v) + \nabla \cdot (\beta v) + \gamma v = \phi$ ,  $x \in \partial\Omega$ , donde  $v = c^{n+1}$ ;  $\alpha = \underline{D}^{n+1}$ ;  $\beta = \underline{u}^{n+1}$ ;  $\gamma = \frac{1}{k} + c^{n+1}$ ;  $\phi = \frac{c^n}{k} + f^{n+1}$ . Así en la ecuación (E.14)  $c^{n+1}(x) = c(x, t_{n+1})$ , es decir, no depende del tiempo, entonces el problema parabólico se convierte en un problema elíptico en cada paso del tiempo.

## Código que se utilizó para la Ecuación de Transporte

Presentamos el código en el Proyecto FEniCS, tomando en cuenta el método de elementos finitos en una formulación general, los espacios de funciones, la malla, las condiciones generales, condiciones de frontera, los tipos de polinomios, el paso del tiempo, y mostramos los resultados.

```

__author__      = "Martin A. Diaz-Viera (mdiazv64@gmail.com), Mario Noyola Rodríguez"
__date__        = "2016-05-23"
__copyright__   = "Copyright (C) 2016 Martin A. Diaz-Viera"
__license__     = "GNU GPL Version 3.0"

from fenics import *
from mshr import *
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

def XYplot_profiles(nt, time, y_time, x_sec_min, x_sec_max, nx, xlab, ylab, fig_title, legend):

# These are the colors that will be used in the plot
    color_sequence = ['#1f77b4', '#aec7e8', '#ff7f0e', '#ffbb78', '#2ca02c',
                      '#98df8a', '#d62728', '#ff9896', '#9467bd', '#c5b0d5',
                      '#8c564b', '#c49c94', '#e377c2', '#f7b6d2', '#7f7f7f',
                      '#c7c7c7', '#bcbd22', '#dbdb8d', '#17becf', '#9edae5']

    fig = plt.figure()
    fig.suptitle(fig_title, fontsize=12)
    delta_x=(x_sec_max-x_sec_min)/nx
    x_sec=np.multiply(range(0,nx+1,1),delta_x)
#     plt.subplots(1, 1, figsize=(12, 14))

    plt.xlim(x_sec_min, x_sec_max)

```

```

plt.xlabel(xlab)
plt.ylabel(ylab)

plt.grid(True)
yt=np.zeros(nx+1)

for i in range(0,nt+1,1):
    for j in range(0,nx+1,1):
        yt[j]=y_time[j,i]
        subfig = fig.add_subplot(1, 1, 1)
        label = 't = ' + str(time[i])
        plt.plot(x_sec, yt, color=color_sequence[i % 20], label=label)
        if legend:
            subfig.legend()
#         subfig.grid(True)

#     plt.show(block=True)
    return fig

#Unit Conversion Factors
millidarcy=1e-15 # [m^2]
millilitr=1e-6 # [m^2]
psi=6894.757 # [Pa]
minute=60 # [minute]=60 s
hour=3600 # [h]=3600 s
day=24*hour # [day]=86400 s

# Optimise compilation of forms
parameters["form_compiler"]["cpp_optimize"] = True
parameters["form_compiler"]["optimize"] = True

# Domain and mesh generation
dolfin.set_log_level(dolfin.DEBUG)

# Define 2D geometry
l=0.08 # core length L=0.08 [m]
d=0.038 # core diameter d=0.038 [m]
r=d/2 # core radius
xmin=0.
xmax=d
ymin=0.
ymax=l
domain = Rectangle(dolfin.Point(xmin, ymin), dolfin.Point(xmax, ymax))
dolfin.info("\nVerbose output of 2D geometry:")
dolfin.info(domain, True)

# Creating a mesh generator object gives access to parameters of the
generator = CSGCGALMeshGenerator2D()
generator.parameters["cell_size"] = 0.25

```

```

generator.parameters["mesh_resolution"] = 30.0
generator.parameters["edge_truncate_tolerance"] = -1.0
generator.parameters["triangle_shape_bound"] = 0.125
#generator.parameters["pixel_size"] = 1e-16
generator.parameters.values()
generator.parameters.to_dict()

# Generate and plot mesh
mesh = dolfin.Mesh()
mesh = generate_mesh(domain, 64)

print "-----"
print("Mesh Statistics:")
print("-----")
print "hmin =", mesh.hmin()
print "hmax =", mesh.hmax()
print "num. cells =", mesh.num_cells()
print "num. edges =", mesh.num_edges()
print "num. faces =", mesh.num_faces()
print "num. facets =", mesh.num_facets()
print "num. vertices =", mesh.num_vertices()
print "-----"

plot(mesh, "2D mesh")
interactive()

n = FacetNormal(mesh)
#Define function space
fs_order = 1 # function space order
V = FunctionSpace(mesh, 'Lagrange', fs_order)

PecletGlobal=100

# Physical parameters, functional forms and boundary conditions
print "-----"
print("Data of the Problem:")
print("-----")
R=1 # Adsorption term
print "Adsorption term, R =", R
u_in=0.0223/day # Injection velocity [m/s]
print "Injection velocity, u_in =", u_in, '[m/s]'
u_x=0 # Darcy velocity in the x-direction [m/s]
print "Darcy velocity in the x-direction, u_x =", u_x, '[m/s]'
u_y=u_in # Darcy velocity in the y-direction [m/s]
print "Darcy velocity in the y-direction, u_y =", u_y, '[m/s]'
alfa_L=1/PecletGlobal #Longitudinal dispersion coefficient [m]
print "Longitudinal dispersion coefficient, alfa_L =", alfa_L, '[m]'
alfa_T=0 # Transversal dispersion coefficient [m]
print "Transversal dispersion coefficient, alfa_T =", alfa_T, '[m]'

```

```

D_xx= alfa_L*u_x # [m2/s]
print ", Dxx =", D_xx, '[m2/s]'
D_xy=0 # [m2/s]
print ", Dxy =", D_xy, '[m2/s]'
D_yx=D_xy # [m2/s]
print ", Dyx =", D_yx, '[m2/s]'
D_yy= alfa_L*u_y # [m2/s]
print ", Dyy =", D_yy, '[m2/s]'
q_c=0 # Source term
print "Source term, q_c =", q_c, '[ppm/s]'
c0= 1326.2 # Initial concentration [ppm]
print "Initial concentration, c0 =", c0, '[ppm]'
c_in=c0/100 # [ppm]
print "Injected concentration, c_in =", c_in, '[ppm]'

#Time
t_0=0 # [s] #Initial time
print "Initial time, t_0 =", t_0, '[s]'
t_N=400000 # [s] #End time
print "Final time, t_N =", t_N, '[s]'
dt = 2500 # Time step
print "Time step, dt =", dt, '[s]'
NT=int((t_N-t_0)/dt) #Number of time steps
print "Number of time steps, NT =", NT

# Hydrodynamic dispersion tensor
D = as_matrix(((D_xx, D_xy), (D_yx, D_yy)))
# Vector velocity
u = as_vector([u_x, u_y])
# Boundary Conditions

boundary_parts = FacetFunction("size_t", mesh)
#Boundary condition at the bottom
class BottomBoundary(SubDomain):
    def inside(self, x, on_boundary):
        tol = 1E-14 # tolerance for coordinate comparisons
        return on_boundary and abs(x[1] - ymin) < tol

Gamma_1 = BottomBoundary()
Gamma_1.mark(boundary_parts, 1)

#If Dirichlet condition
c_B = Constant(c_in)
bc1=DirichletBC(V, c_B, boundary_parts, 1)

#Boundary condition at the right
class RightBoundary(SubDomain):
    def inside(self, x, on_boundary):
        tol = 1E-14 # tolerance for coordinate comparisons

```

```

        return on_boundary and abs(x[0] - xmax) < tol

Gamma_2 = RightBoundary()
Gamma_2.mark(boundary_parts, 2)

#Boundary condition at the top
class TopBoundary(SubDomain):
    def inside(self, x, on_boundary):
        tol = 1E-14 # tolerance for coordinate comparisons
        return on_boundary and abs(x[1] - ymax) < tol

Gamma_3 = TopBoundary()
Gamma_3.mark(boundary_parts, 3)

#Boundary condition at the left
class LeftBoundary(SubDomain):
    def inside(self, x, on_boundary):
        tol = 1E-14 # tolerance for coordinate comparisons
        return on_boundary and abs(x[0] - xmin) < tol

Gamma_4 = LeftBoundary()
Gamma_4.mark(boundary_parts, 4)

bcs = [bc1]

dx=Measure('dx')[boundary_parts]
ds=Measure('ds')[boundary_parts]

#Initial condition
c_0 = Expression('c0 + t',c0=c0,t=t_0, degree=1)
c_1 = project(c_0, V)

# FIXME: Select a 'good' quadrature degree
ffc_parameters = {"quadrature_degree": fs_order + 1, "representation": "quadrature"}

# Define Trial and Test functions
c = TrialFunction(V)
w = TestFunction(V)

# Variational forms and problem
a = R*c*w*dx + dt*inner(nabla_grad(w), dot(D,nabla_grad(c))-u*c)*dx \
    + dt*u_y*c*w*ds(3)
L = (R*c_1 + dt*q_c)*w*dx #- dt*g_in*w*ds(1)

#A = assemble(a, exterior_facet_domains=boundary_parts)
A = assemble(a)
b = None

c_file = File("vtk/concentration.pvd")

```



```

c = Function(V) # the unknown at a new time level

t = t_0
i=0
time=np.zeros(NT+1)
c_t_in=np.zeros(NT+1)
c_t_out=np.zeros(NT+1)

NY=100
delta_y_section=(ymax-ymin)/(NY)

y_section=np.multiply(range(0,NY+1,1),delta_y_section)
cy_time=np.zeros((NY+1, NT+1))

for jj in range(0,NY+1,1):
    cy_time[jj,0] =c0

while t < t_N:
#   b = assemble(L, exterior_facet_domains=boundary_parts)
    b = assemble(L, tensor=b)
    c_0.t = t
    for bc in bcs:
        bc.apply(A, b) #if Dirichlet conditions
    solve(A, c.vector(), b)
    t += dt
    i=i+1
    c_1.assign(c)
    c_e = interpolate(c_0, V)
    maxdiff = np.abs(c_e.vector().array()-c.vector().array()).max()
    print 'Max error, t=%.2f: %-10.3f' % (t, maxdiff)
    ch = project(c)
    ch.rename("conc", "conc")
    c_file << ch, t
    time[i]=i*dt
    c_t_in[i]=c(r,0)
    c_t_out[i]=c(r,1)

    for jj in range(0,NY+1,1):
        cy_time[jj,i] =c(r, jj*delta_y_section)

#plt.show(block=True)
xlab='y-coordinate [m]'
ylab='C [ppm]'
nt=NT
nx=NY
x_section=y_section
y_time=cy_time
x_sec_min=ymin

```

```
x_sec_max=y_max
fig_title='conc vs time'
legend=False
XYplot_profiles(nt, time, y_time, x_sec_min, x_sec_max, nx, xlab, ylab, fig_title, legend)

xlab='Time [s]'
ylab='C [ppm]'
xmin=t_0
xmax=t_N
symbol_color='r-'
XYplot(time, c_t_in, xmin, xmax, xlab, ylab, symbol_color)
XYplot(time, c_t_out, xmin, xmax, xlab, ylab, symbol_color)
df=pd.DataFrame(c_t_out)

df.to_csv('c_t_out.csv')

print "Dyy =", D_yy
print "u_y =", u_y
print "Peclet Global =", PecletGlobal
print "hmin =", mesh.hmin()
print "Peclet Local minimo =", Pe_y_min
print "hmax =", mesh.hmax()
print "Peclet Local maximo =", Pe_y_max
```



# Bibliografía

- ALLEN, M.B., HERRERA, I., Y PINDER, G.F. *Numerical modeling in science and engineering*. John Wiley & Sons (1988) **83**
- BABUŠKA, I. Y RHEINBOLDT, W. *Error estimate for Adaptive finite element computations*. SIAM (1978) **5**
- BEAR, J. *Dynamics of Fluids in Porous Media*. Dover, Inc. New York (1988) **95**
- BRENNER, S.C. Y SCOTT, L.R. *The Mathematical Theory of Finite Element Methods*. Texts in Applied Mathematics, Springer (2008) **5, 8, 93, 94**
- BRYANT, R.S., BERTUS, K.M., CHANG, M.M., Y CHASE, K.L. Laboratory Studies of Parameters Involved in Modeling Microbial Oil Mobilization. *Society of Petroleum Engineers* (24205) (1992) **xvii**
- BRYANT, S.L. Y LOCKHART, T.P. Reservoir Engineering Analysis of Microbial Enhanced Oil Recovery. *Society of Petroleum Engineers* (79719) (2002) **xv, xv**
- CARMAN, P.C. *Flow of gases through porous media*. Butterworths Scientific Publications London (1956) **59, 78**
- CHANG, M.M., CHUNG, F.T.H., BRYANT, R.S., GAO, H.W., Y BURCHFIELD, T.E. Modeling and laboratory investigation of microbial transport phenomena in porous media. En *SPE Annual Technical Conference and Exhibition*, págs. 299–308 (1991) **xvi, 63, 80, 81**
- CHANG, M.M., BRYANT, R.S., STEPP, A.K., Y BERTUS, K.M. Modeling and laboratory investigations of microbial oil recovery mechanisms in porous media. Informe técnico, National Inst. for Petroleum and Energy Research, Bartlesville, OK (United States) (1992) **xvi, 59**
- CHEN, Z. *Finite Element Methods And Their Applications*. Scientific Computation. Springer (2005) **5, 12, 92**
- CHEN, Z. *Reservoir simulation: Mathematical techniques in oil recovery*, tomo 1. Siam (2007) **78**
- CIARLET, P. *Lectures on the Finite Element Method*. Bombay (1975) **5**
- CIARLET, P.G. *The Finite Element Method Elliptic Problems*. SIAM's Classics in Applied Mathematics (2002) **5, 12**
- COMSOL MULTIPHYSICS. User's Guide, version 3.5 a. *COMSOL AB* (2008) **xix, 64**

- CORAPCIOGLU YAVUZ, M. Y HARIDAS, A. Transport and fate of microorganisms in porous media: A theoretical investigation. *Journal of Hydrology* **72**(1):149–169 (1984) **58**
- CORAPCIOGLU YAVUZ, M. Y HARIDAS, A. Microbial transport in soils and groundwater: A numerical model. *Advances in Water Resources* **8**(4):188–200 (1985) **xvi**
- COURANT, R. *Variational method for the Solutions of equilibrium and vibrations*. Bulletin of the American Mathematical (1943) **5**, **7**
- DESAI, R.P., NIELSEN, L.K., Y PAPOUTSAKIS, E.T. Stoichiometric modeling of *Clostridium acetobutylicum* fermentations with non-linear constraints. *Journal of biotechnology* **71**(1):191–205 (1999) **xvii**
- DÍAZ-VIERA, M.A. Y MOCTEZUMA-BERTHIER. Dynamic porosity and permeability modification due to microbial growth using a coupled flow and transport model in porous media. En M.A. Díaz-Viera, P.N. Sahay, M. Coronado, y A. Ortiz-Tapia (editores), *Mathematical and Numerical Modeling in Porous Media: Applications in Geosciences, Multiphysics Modeling Series*, tomo 6, págs. 79–96. CRC Press, London (2012) **51**, **57**
- DÍAZ-VIERA, M.A., LÓPEZ-FALCÓN, D.A., Y ORTIZ-TAPIA, A. MODELACIÓN MATEMÁTICA Y SIMULACIÓN NUMÉRICA DE RECUPERACIÓN MEJORADA DE ACEITE MEDIANTE MICROORGANISMOS (MEOR). *Instituto Nacional del Derecho de Autor No.*(03-2008-071111030100-01):61 (2008) **1**
- DÍAZ-VIERA, M.A., ORTIZ-TAPIA, A., Y HERNÁNDEZ-PÉREZ, J. MODELO MATEMÁTICO, NUMÉRICO Y COMPUTACIONAL PARA LA SIMULACIÓN DE PRUEBAS DE LABORATORIO DEL PROCESO DE RECUPERACIÓN MEJORADA DE HIDROCARBUROS VIA MICROBIANA (MEOR). *Instituto Nacional del Derecho de Autor No.*(03-2015-022411202600-01):202 (2015) **xviii**, **29**
- DIETRICH, B. *Finite Element Theory, Fast solver, in applications in solid mechanics*. Cambridge university press (2007) **6**
- GALERKIN, B.G. *On some problems of elastic equilibration of rods and plates*. Vestnik Inzenerou petrograd (1915) **5**
- HERRERA, I. Y PINDER, G.F. *Mathematical Modeling in Science and Engineering, An Axiomatic Approach*. John Wiley & Sons (2012) **30**, **86**, **89**
- HUANG, C. Y MONTEAGUDO, J. Mathematical Modeling of Microbial Processes for Oil Recovery. En *ECMOR XIII 13<sup>th</sup> European Conference on the Mathematics of Oil Recovery, Biarritz, France* (2012) **xviii**
- ISLAM, M. Y GIANETTO, A. Mathematical modelling and scaling up of microbial enhanced oil recovery. *Journal of Canadian Petroleum Technology* **32**(4) (1993) **xvi**
- KIM, S.B. Numerical analysis of bacterial transport in saturated porous media. *Hydrological processes* **20**(5):1177–1186 (2006) **xvii**
- KNAPP, F.CIVAN, M.M. Modeling Growth and Transport of Microorganisms in Porous Formations. En *12th World Congress on Scientific Computation* (1988) **xvi**
- LAPPAN, E. Y FOGLER, H. Reduction of porous media permeability from in situ *Leuconostoc mesenteroides* growth and dextran production. *Biotechnology and Bioengineering* **50**(1):6–15 (1996) **58**
- LI, J., LIU, J., TREFRY, M.G., PARK, J., LIU, K., HAQ, B., JOHNSTON, C.D., Y VOLK, H. Interactions of microbial-enhanced oil recovery processes. *Transport in porous media* **87**(1):77–104 (2011) **xviii**

- LOGG, A., MARDAL, K.A., Y WELLS, G.N. *Automated Solution of Differential Equations by the Finite Element-Method*. Springer (2012) **5, 13, 17**
- MONOD, J. The growth of bacterial cultures. *Annual Reviews in Microbiology* **3**(1):371–394 (1949) **xvi**
- NIELSEN, S., SHAPIRO, A., MICHELSEN, M., Y STENBY, E. 1D Simulations for Microbial Enhanced Oil Recovery with Metabolite Partitioning. *Transport in Porous Media* **85**:785–802 (2010) **xvii**
- PETTER LANGTANGEN, H. Y LOGG, A. *Solving PDEs in python The FEniCS Tutorial I*. ISBN 978-3-319-52461-0. SpringerOpen (2016) **xix, 22**
- ROSAS-MEDINA, A.A. Y HERRERA, I. *El Número de Pécelet y su Significación en la Modelación de Transporte Difusivo de Contaminantes*. Tesis UNAM (2005) **49**
- SARKAR, A.K., GOURSAUD, J.C., SHARMA, M.M., Y GEORGIU, G. Critical evaluation of MEOR processes. *In Situ*, **13**(4), (1989) **xv**
- SARKAR, A., SHARMA, M., Y GEORGIU, G. Compositional numerical simulation of MEOR processes. *Microbial Enhancement of Oil Recovery-Recent Advances, Elsevier Science Publishing Company, Inc., New York City* págs. 331–343 (1991) **xvi**
- THULLNER, M., SCHROTH, M., ZEYER, J., Y KINZELBACH, W. Modeling of a microbial growth experiment with bioclogging in a two-dimensional saturated porous media flow field. *Journal of contaminant hydrology* **70**(1):37–62 (2004) **xvii**
- VAN GENUCHTEN, M. Y ALVES, W.J. Analytical Solutions of the one Dimensional Convective-dispersive Solute Transport Ecuation. *Agricultural Research Service* (1982) **xx, 43**
- ZHANG, X., KNAPP, R.M., Y MCINERNEY, M.J. Quantitation of microbial products and their effectiveness in enhanced oil recovery. Final report. Informe técnico, Oklahoma Univ., Norman, OK (United States) (1995) **xvii**