

**UNIVERSIDAD MICHOACANA
DE SAN NICOLÁS DE HIDALGO**

FACULTAD DE CIENCIAS FÍSICO-MATEMÁTICAS
“Mat. Luis Manuel Rivera Gutiérrez”

**MAESTRÍA EN CIENCIAS EN INGENIERÍA FÍSICA
DIVISIÓN DE ESTUDIOS DE POSGRADO**



TESIS

**MODELADO DE PROBLEMAS DE CONVECCIÓN NATURAL DIFUSIVA
DOBLE USANDO DIFERENCIAS FINITAS GENERALIZADAS**

**PARA OBTENER EL GRADO DE MAESTRO EN
CIENCIAS EN INGENIERÍA FÍSICA**

PRESENTA

Ricardo Román Gutiérrez

ASESOR

**Doctor en Ciencias en Matemáticas
Dr. Francisco Javier Domínguez Mota**

PREFACIO (RESUMEN)

A lo largo de la historia el hombre ha ido creando diferentes formas de expresar con matemáticas los problemas físicos a los que se ha ido enfrentando. Poco a poco las matemáticas fueron evolucionando y mejorando para expresar de forma más clara y sencilla fenómenos físicos; hoy en día lo más usual es utilizar ecuaciones diferenciales para crear los modelos matemáticos que describen estos fenómenos físicos; o en su defecto, en fenómenos más complicados, usar sistemas de ecuaciones diferenciales, estos sistemas pueden ser no lineales, y además pueden estar acoplados. En este trabajo se usará un método llamado diferencias finitas generalizadas, el cual se usa para resolver ecuaciones diferenciales. Este método está basado en otro que es conocido como el método de diferencias finitas. La diferencia entre ambos es que el segundo necesita implementarse en mallas de nodos con cierta estructura, mientras que el primero puede trabajar con mallas sin estructura. El método de diferencias finitas generalizadas será usado en este trabajo para resolver un problema de reacción difusión, el cual es tiene el nombre de *convección natural con doble difusividad en un medio poroso saturado dentro de un paralelogramo*, este problema es en dos dimensiones debido a que se considera un dominio dentro de un paralelogramo. Este fenómeno puede ser descrito con un sistema de tres ecuaciones diferenciales parciales acopladas, y que debido a que se multiplican derivadas de las funciones que se desean encontrar, entonces el sistema también es no lineal. Se utilizaron mallas triangulares no estructuradas para resolver los diferentes ejemplos presentados, ya que, si se usan mallas estructuradas donde los nodos se alineen verticalmente, entonces el método tiene problemas, debido a que para aproximar parciales en x se requiere la diferencia en la distancia horizontal entre nodos, y en este caso sería cero, que al multiplicar hace que se pierda información necesaria para aproximar una buena solución. En este trabajo se realizó un desacoplamiento de las ecuaciones diferenciales que describen nuestro problema, haciendo una primera aproximación lineal para las funciones buscadas, y posteriormente aplicando un algoritmo predictor-corrector para corregir los errores que estas aproximaciones pudieran tener. El problema que se resuelve aquí se puede implementar en la simulación de agua subterránea, donde puede incluirse o no la disolución de otro componente en el agua, tal componente puede ser algún contaminante. El método implementado da buenas soluciones para mallas triangulares no estructuradas, por otro lado si tienen estructura entonces se pierde algo de precisión. Para comparar el método se obtuvieron los datos de dos artículos de referencia [([Fan et al., 2013](#)),([Costa, 2004](#))], no dan el valor exacto de las dimensiones del paralelogramo, por lo que estas fueron propuestas. Al comparar las figuras se ve que los resultados son similares, por lo que se considera que las soluciones obtenidas son correctas.

Palabras clave: Ecuaciones Diferenciales, Métodos Numéricos, Solución Numérica, Dos Dimensiones, MatLAB.

PREFACE

Throughout history, man has created different ways of expressing mathematically the physical problems he has been facing. Little by little, mathematics evolved and improved to express physical phenomena more clearly and simply; Nowadays the most usual thing is to use differential equations to create the mathematical models that describe these physical phenomena; or failing that, in more complicated phenomena differential equation systems are used, these systems may be non-linear, and may also be coupled. In this work we will use a method called generalized finite differences, which is used to solve differential equations. This method is based on another that is known as finite differences. The difference between the two is that the second one needs to be implemented in node meshes with a certain structure, while the first one can work with unstructured meshes. The generalized finite difference method will be used to solve a diffusion reaction problem, which is called *natural convection with double diffusivity in a saturated porous medium within a parallelogram*, this problem is in two dimensions due to which is considered a domain within a parallelogram. This phenomenon can be described with a system of three coupled partial differential equations, and because they are multiplied by derivatives of the functions to be found, then the system is also nonlinear. Unstructured triangular meshes were used to solve the different examples presented, since, if structured meshes where the nodes are vertically aligned are used, then the method has problems, because to approximate partials in x requires the difference in the horizontal distance between nodes, and in this case it would be zero, which when multiplied causes the necessary information to approximate a good solution to be lost. In this work, a decoupling of the differential equations that describe our problem was carried out, making a first linear approximation for the functions sought, and then applying a predictor-corrector algorithm to correct the errors that these approximations may have. The problem that is solved here can be implemented in the simulation of groundwater, where the dissolution of another component in the water may or may not be included, such component may be some contaminant. The implemented method gives good solutions for unstructured triangular meshes, on the other hand if they have structure then some precision is lost. To compare the method, data were obtained from two reference articles [(Fan et al., 2013),(Costa, 2004)], they do not give the exact value of the parallelogram dimensions, so these were proposed. When comparing the figures, it is seen that the results are similar, so it is considered that the solutions obtained are correct.

NOTACIÓN

En la siguiente tabla se describen brevemente los símbolos utilizados en las ecuaciones presentadas a lo largo de este trabajo.

Tabla 1: Notación

Símbolo	Significado
x, y	Dimensiones espaciales.
t	Dimensión temporal (tiempo).
$\nabla = \frac{\partial^2}{\partial x^2} \hat{\mathbf{i}} + \frac{\partial^2}{\partial y^2} \hat{\mathbf{j}}$	Operador vectorial “Nabla” para dos dimensiones rectangulares.
$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$	Operador Laplaciano para dos dimensiones rectangulares.
u	Función que puede depender de una o dos variables espaciales (x, y) , y puede ser también dependiente del tiempo t .
Ω	Espacio computacional.
$\partial\Omega$	Frontera del espacio computacional.
H	Altura del paralelogramo.
L	Longitud del paralelogramo.
θ	Ángulo de inclinación del paralelogramo respecto a la horizontal.
\mathbb{R}^2	Espacio plano de dos dimensiones, donde en cada eje de coordenadas se encuentran todos los números reales.
u_{x_j}	Derivada parcial de u respecto de la coordenada j -ésima x_j .
\mathbb{L}	Operador diferencial lineal parcial de segundo orden con coeficientes constantes.
Ψ	Función de flujo.
T	Función de Temperatura.
C	Función de Concentración.
Le	Número de Lewis.
Ra	Número de Rayleigh.
N	Tasa de Flotabilidad.
$\frac{d^n u}{dx_j^n}$	Derivada n -ésima de u respecto de la coordenada j -ésima x_j .
$h_i = x_i - x_0$	Distancia horizontal del nodo central al nodo i -ésimo.
$k_i = y_i - y_0$	Distancia vertical del nodo central al nodo i -ésimo.
$p_i = (x_i, y_i)$	Coordenadas del nodo i -ésimo.

- M Matriz de coeficientes de distancias horizontales h_i y verticales k_i .
- Γ Vector de pesos Γ_i , con $i = 0, 1, \dots, q$, donde q es el número de nodos vecinos al nodo central.
- L Vector de coeficientes que relaciona cada componente con qué derivada se quiere encontrar.
- $\bar{\Psi}$ Función de flujo linealizada.
-

CONTENIDO

PREFACIO	1
PREFACE	2
NOTACIÓN	3
CONTENIDO	5
ÍNDICE DE FIGURAS	7
1. INTRODUCCIÓN	12
1.1. Estructura de la tesis	14
2. CONVECCIÓN NATURAL CON DOBLE DIFUSIVIDAD EN UN MEDIO POROSO SATURADO	16
2.1. Problema Físico	16
2.2. Conceptos Básicos	17
2.2.1. Ecuaciones Diferenciales Hiperbólicas, Parabólicas y Elípticas	18
2.2.2. Ley de Darcy	19
2.2.3. Stream Function (Función de Flujo)	20
2.2.4. Grupos Adimensionales	21
3. METODOLOGÍA	24
3.1. Expansión en Serie de Taylor	24
3.2. Diferencias Finitas	25
3.3. Diferencias Finitas Generalizadas (GFD)	27
3.4. Esquema GFD Generalmente Usado en la Literatura	27
3.5. Esquema GFD Propuesto	29
3.5.1. Aproximación a un operador diferencial lineal \mathbb{L}	30
3.6. Ecuaciones Gobernantes	32
3.7. Discretización del espacio	33
3.8. Linealización del problema	35
4. RESULTADOS	39
4.1. Ejemplos propuestos	39
4.2. Ejemplos con mallas no estructuradas	63

4.2.1. Ejemplos tomados de Fan	63
4.2.2. Ejemplos tomados de Costa	70
5. DISCUSIÓN Y CONCLUSIONES	85
ANEXOS	87
Anexo I: Código de la primera aproximación	87
Anexo II: Código <i>dirichgfd.m</i>	88
Anexo III: Código <i>matrizd.m</i>	89
Anexo IV: Código <i>robingfd_conv.m</i>	90
Anexo V: Segunda aproximación	91
Anexo VI: Código <i>robingfd_conv2.m</i>	93
Anexo VII: Código <i>matrizd_3.m</i>	94
Anexo VIII: Código para ejemplos con mallas generadas automáticamente	95
Anexo IX: Código <i>GenM3.m</i>	97
Anexo X: Código <i>matrizd_4.m</i>	99
Anexo XI: Código <i>robingfd_conv3.m</i>	100
Anexo XII: Código <i>VN4.m</i>	102
Anexo XIII: Código <i>trisubfigura.m</i>	102
Anexo XIV: Ejemplos para comparación	103
Anexo XV: Código <i>fronteras.m</i>	105
Anexo XVI: Código <i>robingfd_conv3b.m</i>	106
Anexo XVII: Código <i>dn.m</i>	107
Anexo XVIII: Código <i>VN4b.m</i>	107
Anexo XIX: Código <i>matrizd_4b.m</i>	108
BIBLIOGRAFÍA	109

ÍNDICE DE FIGURAS

1.1. Simplificación del problema de dispersión de un contaminante en agua subterránea a un problema en un espacio bidimensional en forma de paralelogramo.	13
2.1. Convección Natural.	16
2.2. Medio poroso saturado.	17
2.3. Esquema del experimento de Darcy.	20
2.4. Esquema de una lámina de flujo que transporta calor (conducción y/o convección), desde una capa de contorno caliente a una capa de contorno fría.	22
3.1. Discretización del dominio $[0, 1]$ en el eje x	25
3.2. Criterio de los cuatro cuadrantes con dos nodos en cada cuadrante.	28
3.3. Diagramas esquemáticos de (a) el dominio computacional y (b) las condiciones de contorno.	33
3.4. Malla utilizada en la primera aproximación	33
3.5. Diferentes estructuras en la malla	34
3.6. Malla utilizada en la segunda aproximación	34
3.7. Diferentes ejemplos de mallas creadas con el código de malla automática	35
3.8. Esquema del algoritmo de solución	38
4.1. Solución de la función de flujo Ψ	40
4.2. Campo vectorial de velocidades de flujo obtenidas a partir de la función de flujo Ψ .	40
4.3. Solución de la función de Temperatura T	41
4.4. Solución de la función de Concentración C	41
4.5. Solución de la función de flujo Ψ	42
4.6. Campo vectorial de velocidades de flujo obtenidas a partir de la función de flujo Ψ .	42
4.7. Solución de la función de Temperatura T	43
4.8. Solución de la función de Concentración C	43
4.9. Malla generada automáticamente usando los parámetros geométricos $H = 0.3$, $H/L = 0.25$, $\theta = 30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (°) representan la frontera con condiciones de Neumann.	44
4.10. Función de flujo.	44
4.11. Campo de velocidades obtenido de la función de flujo.	45
4.12. Función de temperatura.	45

4.13. Malla generada automáticamente usando los parámetros geométricos $H = 0.3$, $H/L = 0.25$, $\theta = -30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (°) representan la frontera con condiciones de Neumman.	46
4.14. Función de flujo.	46
4.15. Campo de velocidades obtenido de la función de flujo.	47
4.16. Función de temperatura.	47
4.17. Malla generada automáticamente usando los parámetros geométricos $H = 0.3$, $H/L = 0.5$, $\theta = 30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (°) representan la frontera con condiciones de Neumman.	48
4.18. Función de flujo.	48
4.19. Campo de velocidades obtenido de la función de flujo.	49
4.20. Función de temperatura.	49
4.21. Función de temperatura.	50
4.22. Malla generada automáticamente usando los parámetros geométricos $H = 0.46$, $H/L = 1$, $\theta = 30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (°) representan la frontera con condiciones de Neumman.	50
4.23. Función de flujo.	51
4.24. Campo de velocidades obtenido de la función de flujo.	51
4.25. Función de temperatura.	52
4.26. Malla generada automáticamente usando los parámetros geométricos $H = 0.31$, $H/L = 0.25$, $\theta = 30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (°) representan la frontera con condiciones de Neumman.	52
4.27. Función de flujo.	53
4.28. Campo de velocidades obtenido de la función de flujo.	53
4.29. Función de temperatura.	54
4.30. Malla generada automáticamente usando los parámetros geométricos $H = 0.31$, $H/L = 0.25$, $\theta = -30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (°) representan la frontera con condiciones de Neumman.	54
4.31. Función de flujo.	55
4.32. Campo de velocidades obtenido de la función de flujo.	55
4.33. Función de temperatura.	56
4.34. Malla generada automáticamente usando los parámetros geométricos $H = 0.3$, $H/L = 0.5$, $\theta = 30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (°) representan la frontera con condiciones de Neumman.	56
4.35. Función de flujo.	57
4.36. Campo de velocidades obtenido de la función de flujo.	57
4.37. Función de temperatura.	58
4.38. Función de concentración.	58

4.39. Malla generada automáticamente usando los parámetros geométricos $H = 0.3$, $H/L = 0.5$, $\theta = 30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (°) representan la frontera con condiciones de Neumman.	59
4.40. Función de flujo.	59
4.41. Campo de velocidades obtenido de la función de flujo.	60
4.42. Función de temperatura.	60
4.43. Función de concentración.	61
4.44. Malla generada automáticamente usando los parámetros geométricos $H = 0.3$, $H/L = 0.5$, $\theta = 30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (°) representan la frontera con condiciones de Neumman.	61
4.45. Función de flujo.	62
4.46. Campo de velocidades obtenido de la función de flujo.	62
4.47. Función de temperatura.	63
4.48. Función de concentración.	63
4.49. Malla generada usando los parámetros geométricos $H = 0.7$, $H/L = 0.25$, $\theta = 30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (°) representan la frontera con condiciones de Neumman.	64
4.50. Comparación de la función de flujo obtenida con la metodología propuesta y la obtenida por Fan.	65
4.51. Campo de velocidades obtenido de la función de flujo.	65
4.52. Comparación de la función de temperatura obtenida con la metodología propuesta y la obtenida por Fan.	66
4.53. Malla generada usando los parámetros geométricos $H = 0.6$, $H/L = 0.25$, $\theta = -30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (°) representan la frontera con condiciones de Neumman.	66
4.54. Comparación de la función de flujo obtenida con la metodología propuesta y la obtenida por Fan.	67
4.55. Campo de velocidades obtenido de la función de flujo.	67
4.56. Comparación de la función de temperatura obtenida con la metodología propuesta y la obtenida por Fan.	68
4.57. Malla generada usando los parámetros geométricos $H = 1$, $H/L = 0.5$, $\theta = 30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (°) representan la frontera con condiciones de Neumman.	68
4.58. Comparación de la función de flujo obtenida con la metodología propuesta y la obtenida por Fan.	69
4.59. Campo de velocidades obtenido de la función de flujo.	69
4.60. Comparación de la función de temperatura obtenida con la metodología propuesta y la obtenida por Fan.	70
4.61. Comparación de la función de concentración obtenida con la metodología propuesta y la obtenida por Fan.	70
4.62. Malla generada usando los parámetros geométricos $H = 0.9$, $H/L = 1$, $\theta = 30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (°) representan la frontera con condiciones de Neumman.	71

4.63. Comparación de la función de flujo obtenida con la metodología propuesta y la obtenida por Costa.	71
4.64. Campo de velocidades obtenido de la función de flujo.	72
4.65. Comparación de la función de temperatura obtenida con la metodología propuesta y la obtenida por Costa.	72
4.66. Malla generada usando los parámetros geométricos $H = 0.6$, $H/L = 0.25$, $\theta = 30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (°) representan la frontera con condiciones de Neumman.	73
4.67. Comparación de la función de flujo obtenida con la metodología propuesta y la obtenida por Costa.	73
4.68. Campo de velocidades obtenido de la función de flujo.	74
4.69. Comparación de la función de temperatura obtenida con la metodología propuesta y la obtenida por Costa.	74
4.70. Malla generada usando los parámetros geométricos $H = 0.5$, $H/L = 0.25$, $\theta = -30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (°) representan la frontera con condiciones de Neumman.	75
4.71. Comparación de la función de flujo obtenida con la metodología propuesta y la obtenida por Costa.	75
4.72. Campo de velocidades obtenido de la función de flujo.	76
4.73. Comparación de la función de temperatura obtenida con la metodología propuesta y la obtenida por Costa.	76
4.74. Malla generada usando los parámetros geométricos $H = 0.7$, $H/L = 0.5$, $\theta = 30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (°) representan la frontera con condiciones de Neumman.	77
4.75. Comparación de la función de flujo obtenida con la metodología propuesta y la obtenida por Costa.	77
4.76. Campo de velocidades obtenido de la función de flujo.	78
4.77. Comparación de la función de temperatura obtenida con la metodología propuesta y la obtenida por Costa.	78
4.78. Comparación de la función de concentración obtenida con la metodología propuesta y la obtenida por Costa.	79
4.79. Malla generada usando los parámetros geométricos $H = 0.8$, $H/L = 0.5$, $\theta = 30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (°) representan la frontera con condiciones de Neumman.	79
4.80. Comparación de la función de flujo obtenida con la metodología propuesta y la obtenida por Costa.	80
4.81. Campo de velocidades obtenido de la función de flujo.	80
4.82. Comparación de la función de temperatura obtenida con la metodología propuesta y la obtenida por Costa.	81
4.83. Comparación de la función de temperatura obtenida con la metodología propuesta y la obtenida por Costa.	81
4.84. Malla generada usando los parámetros geométricos $H = 0.7$, $H/L = 0.5$, $\theta = 30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (°) representan la frontera con condiciones de Neumman.	82

4.85. Comparación de la función de flujo obtenida con la metodología propuesta y la obtenida por Costa.	82
4.86. Campo de velocidades obtenido de la función de flujo.	83
4.87. Comparación de la función de temperatura obtenida con la metodología propuesta y la obtenida por Costa.	83
4.88. Comparación de la función de concentración obtenida con la metodología propuesta y la obtenida por Costa.	84

Capítulo 1

INTRODUCCIÓN

El descubrimiento por parte de Newton y de Leibniz, en el siglo XVII, sobre las ideas del cálculo integral y su relación con el cálculo diferencial fue crucial para el avance que sufrieron las matemáticas. Hay una gran variedad de problemas en los cuales se desea conocer un elemento variable a partir de su coeficiente de variación o, en otras palabras, como varía una variable respecto a una o varias variables. Conocemos a estas variaciones con el nombre de derivadas, y a las relaciones que hay entre variables y sus derivadas las denominamos como ecuaciones diferenciales.

Los sistemas de ecuaciones diferenciales parciales con valores iniciales y condiciones de frontera, son los que nos ayudan a crear modelos matemáticos de sistemas continuos. Algunos ejemplos de estos sistemas continuos macroscópicos son: los yacimientos petroleros, la atmósfera, los campos electromagnéticos, los océanos, el aparato circulatorio de los seres humanos, la corteza terrestre, entre muchos otros sistemas de interés en Ciencia y en Ingeniería ([Hutter, 2004](#)).

Las soluciones de estos sistemas de ecuaciones diferenciales, en general, no pueden ser obtenidas por métodos analíticos, esto es debido al gran número de grados de libertad, así como a que en muchas ocasiones son sistemas no lineales o las ecuaciones del sistema están acopladas, con lo cual no es posible aplicar los métodos de solución analítica convencionales. Por esta razón, a veces resulta más conveniente utilizar algún método numérico para obtener la solución de tales sistemas. Para lograr obtener buenas soluciones es necesario contar con un buen esquema de solución y un método numérico adecuado.

Con el paso del tiempo los problemas a los que nos tenemos que enfrentar han ido creciendo en cantidad y también en dificultad. Aunque a su vez la ciencia y la tecnología han avanzado con la misma rapidez; podemos de aquí destacar que nuevos métodos numéricos surgen, además de que continuamente se optimizan los que ya se conocen, sumado a todo esto tenemos la creciente potencia de cálculo de ordenadores cada vez mayores y mejores. Gracias a todo esto hemos podido hacerles frente a los diversos problemas que existen en la actualidad.

Durante años se ha trabajado en la solución analítica de muchos problemas descritos por ecuaciones diferenciales, sin embargo, para resolver gran parte de estos problemas de manera analítica se tienen que hacer diversas simplificaciones o aproximaciones, con lo cual se pierde precisión.

Muchos métodos para la resolución de estos problemas han surgido a lo largo de los años, de los cuales se pueden mencionar los siguientes: el método de la ecuación integral (IEM¹) ([Na, 1980](#)),

¹Por sus siglas en inglés, Integral Equation Method.

el método de los elementos finitos (FEM²)(Gockenbach, 2006) y el método de diferencias finitas (FDM³)(Na, 1980); de este último se deriva el método que se usará en este trabajo, el cual es el método de diferencias finitas generalizadas (GFDM⁴) [(Li et al., 2018),(Chávez-Negrete et al., 2018)], el cual ha probado ser un buen enfoque para la resolución de ecuaciones diferenciales.

En la actualidad la investigación dedicada al desarrollo de nuevas metodologías para la resolución de ecuaciones diferenciales, así como la aplicación de estas metodologías a diferentes problemas, es de especial interés para diversas aplicaciones ingenieriles. Tales aplicaciones requieren de un modelo matemático adecuado y preciso para la solución de problemas numéricos, además de que el modelo puede ser usado para visualizar de manera gráfica los resultados numéricos obtenidos.

En este trabajo se resolverá un problema de reacción difusión utilizando el método de diferencias finitas generalizadas (GFDM). La ecuación de reacción difusión $\frac{\partial u}{\partial t} = \nabla^2 u + f(u)$, donde para una dimensión tenemos $u = u(x, t)$ y $\nabla^2 u = \frac{\partial^2 u}{\partial x^2}$, se presenta en una gran variedad de modelos de problemas que se presentan en la naturaleza. Algunos ejemplos son: ecuación de calor, onda viajera, modelo de crecimiento logístico, genética poblacional, dispersión de mamíferos, etcétera.

Debido a lo amplio que es el campo de fenómenos descritos por la ecuación de reacción difusión, en este proyecto de investigación nos enfocaremos en obtener soluciones numéricas del problema denominado *convección natural con doble difusividad en estado estacionario, en un medio poroso saturado, dentro de un paralelogramo*. Como el problema es en estado estacionario entonces la función ya no depende del tiempo; sin embargo, al analizarla en un paralelogramo necesitamos dos dimensiones espaciales para describirlo, por lo que la función que queremos obtener al resolver esta ecuación diferencial es de la forma $u = u(x, y)$.

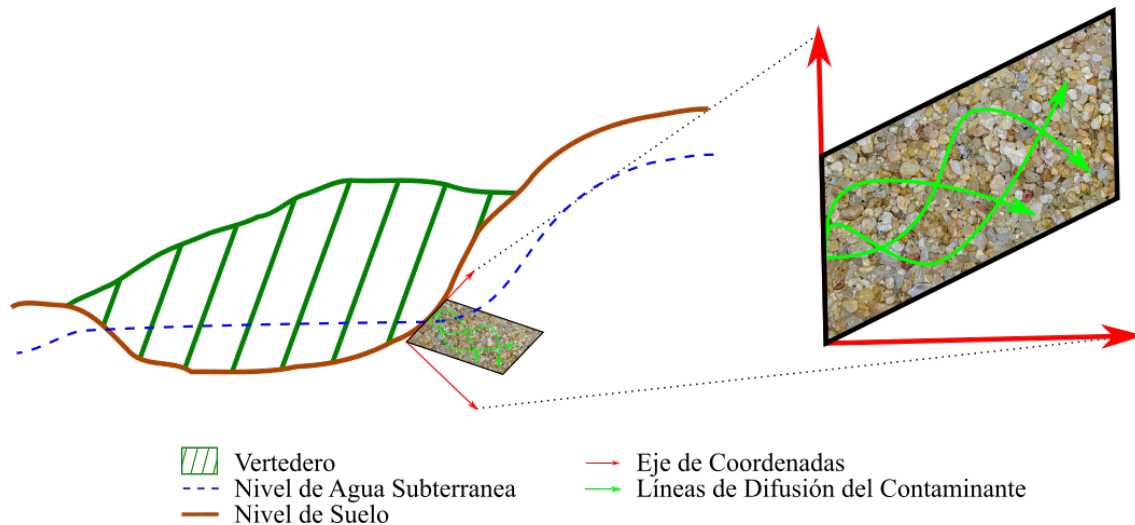


Figura 1.1: Simplificación del problema de dispersión de un contaminante en agua subterránea a un problema en un espacio bidimensional en forma de paralelogramo.

En varios trabajos ya se ha estudiado el problema de convección natural con doble difusividad

²Por sus siglas en inglés, Finite Element Method.

³Por sus siglas en inglés, Finite Difference Method.

⁴Por sus siglas en inglés, Generalized Finite Difference Method.

en un medio poroso saturado (CNDDMPS), en los cuales se han aplicado diferentes metodologías para solucionar esta problemática, como lo son: método implícito de dirección alterna (ADIM⁵)(Bourich et al., 2004), método integral de dominio de frontera (BDIM⁶)(Kramer et al., 2007), este método es una extensión del método de elementos de frontera (BEM⁷)(Gaul et al., 2013), también se ha implementado el método de colocación local de función de base radial (LRBFCM⁸)(Fan et al., 2013) que es una extensión del método de función de base radial (RBFM⁹), otro método usado es el método de elementos finitos basado en un volumen de control (CVBFEM¹⁰)(Costa, 2004). También se ha usado un esquema de diferencias finitas generalizadas (FDM) diferente al propuesto en este trabajo(Li et al., 2018). En este trabajo se utilizará el enfoque de las diferencias finitas generalizadas (GFDM) para resolverlo, pero usando un esquema diferente, ya que, este método ha demostrado ser mejor que otros para la solución de diversas ecuaciones diferenciales [(Chávez-Negrete et al., 2018),(Domínguez-Mota et al., 2014)]. Se tiene el sistema de ecuaciones diferenciales parciales que describe el problema, el cual es un sistema no lineal y acoplado, el cual se resolverá desacoplando las ecuaciones y utilizando un algoritmo predictor corrector, donde se resuelven las ecuaciones utilizando el esquema de GFDM propuesto; para ello se desarrollarán los códigos necesarios para la solución numérica utilizando el software MatLAB como lenguaje de programación. Una vez obtenidos los resultados se representarán de manera gráfica las funciones que describen el problema, y estas gráficas se compararán con aquellas obtenidas utilizando otros métodos.

El problema de CNDDMPS posteriormente se puede ampliar para observar la dispersión de un contaminante en el agua subterránea, pero por el momento aquí nos enfocaremos a resolver el problema solamente dentro de un paralelogramo, como se muestra en la figura 1.

1.1. Estructura de la tesis

En el *capítulo 2* se describe parte por parte el problema físico que se quiere resolver. Contiene además, una explicación breve de conceptos básicos que pueden ser útiles para tener una mejor comprensión del problema físico a resolver.

En el *capítulo 3* se desarrollan conceptos de métodos numéricos y de cálculo en diferencias finitas. También se describe un esquema de GFDM utilizado generalmente en la literatura, y posteriormente se describe el esquema de GFDM propuesto para usar en este trabajo. Posteriormente se presentan las ecuaciones gobernantes del problema de convección natural con doble difusividad, el cual se pretende resolver en este trabajo. Por último se presenta en forma gráfica como se hizo la discretización del dominio espacial (x, y) , en diferentes mallas de puntos donde se encontrarán las aproximaciones a las funciones a resolver de las ecuaciones gobernantes.

En el *capítulo 4* se muestran los resultados obtenidos para diferentes ejemplos. Los ejemplos fueron tomados de artículos que aplican un esquema distinto al usado en este trabajo, estos artículos son los de Fan (Li et al., 2018) y Costa (Costa, 2004). Se usaron parámetros físicos iguales que en los ejemplos encontrados en los artículos antes mencionados. Lo que cambia es el número de

⁵Por sus siglas en inglés, Alternating Direction Implicit Method.

⁶Por sus siglas en inglés, Boundary Domain Integral Method.

⁷Por sus siglas en inglés, Boundary Element Method.

⁸Por sus siglas en inglés, Local Radial Basis Function Collocation Method.

⁹Por sus siglas en inglés, Radial Basis Function Method.

¹⁰Por sus siglas en inglés, Control-Volume-Based Finite Element Method.

nodos vecinos para obtener la aproximación de un nodo central, la forma de la malla de nodos, y el esquema usado de diferencias finitas generalizadas.

En el *capítulo 5* se discuten los resultados obtenidos y la comparación de resultados de otros autores. Posteriormente se dan las conclusiones obtenidas en este trabajo.

Capítulo 2

CONVECCIÓN NATURAL CON DOBLE DIFUSIVIDAD EN UN MEDIO POROSO SATURADO

2.1. Problema Físico

Para comprender mejor los agentes físicos que propician la presencia del fenómeno en estudio, denominado como “*Convección Natural de Doble Difusividad*” (CNDD), lo dividiremos en dos partes, convección natural y doble difusividad.

La **convección natural** (ver figura 2.1(a)) es un fenómeno físico de transporte de calor, en el que el movimiento de un fluido no es generado por una fuente externa, como pueden ser una turbina, un ventilador, un dispositivo de succión, etcétera, sino que es generado por un gradiente de densidad en el propio fluido.

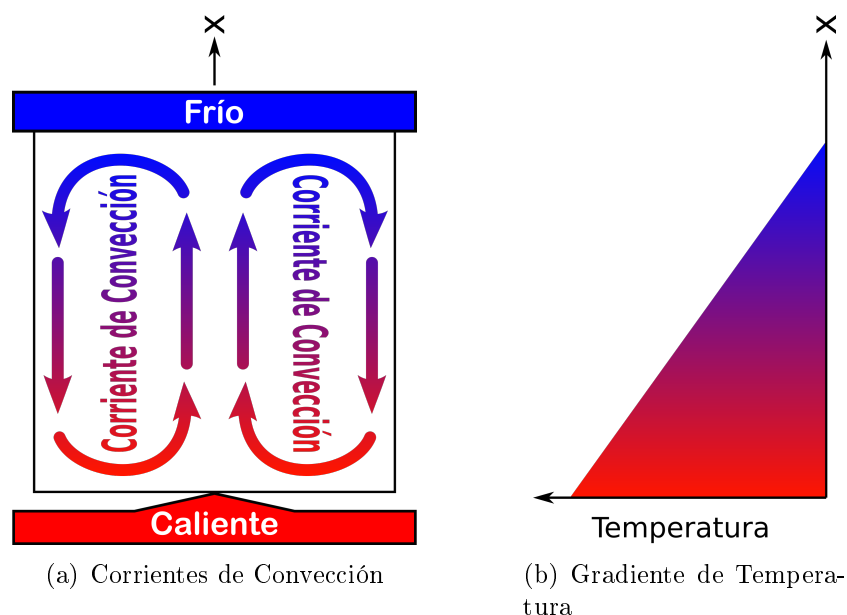


Figura 2.1: Convección Natural.

El gradiente de densidad del fluido puede ser provocado por dos factores, de aquí el término **doble difusividad**. Por un lado puede ser provocado por un *gradiente de temperatura* (ver figura 2.1(b)), ya que, el fluido al incrementar la temperatura incrementa su volumen, y por lo tanto disminuye su densidad. Debido a esta deficiencia de densidad el fluido caliente flota hacia arriba, y el fluido frío de los alrededores se hunde para ocupar el lugar del fluido desplazado. El gradiente de densidad también puede ser provocado por un *gradiente de concentración* de algún soluto disuelto en el fluido, que por el simple hecho de que se encuentre dentro del fluido crea una diferencia en la densidad, lo que provoca un efecto análogo al de la temperatura.

Cabe notar que para que ocurra convección se necesita la presencia de una fuerza debida a una aceleración, como puede ser la gravedad, la fuerza centrífuga o el efecto Coriolis, ya que es esta fuerza la que provoca la flotabilidad, donde el fluido menos denso flota y el más denso se hunde.

La convección natural se puede estudiar en dos y en tres dimensiones espaciales, pero en este estudio nos enfocaremos solamente en problemas en dos dimensiones. Donde el espacio de cómputo Ω es un paralelogramo, donde H es su altura, L es la longitud de su base y θ es el ángulo de inclinación de la base del paralelogramo respecto del eje de coordenadas horizontal x .

Dentro de este estudio nos delimitaremos al estudio del fenómeno de convección natural en un medio poroso saturado con un fluido incompresible(ver figura 2.2), como ejemplo se puede mencionar el suelo saturado con agua. Además de que nuestro estudio se limitará al espacio contenido dentro de un paralelogramo, como se muestra en la sección 3.7.

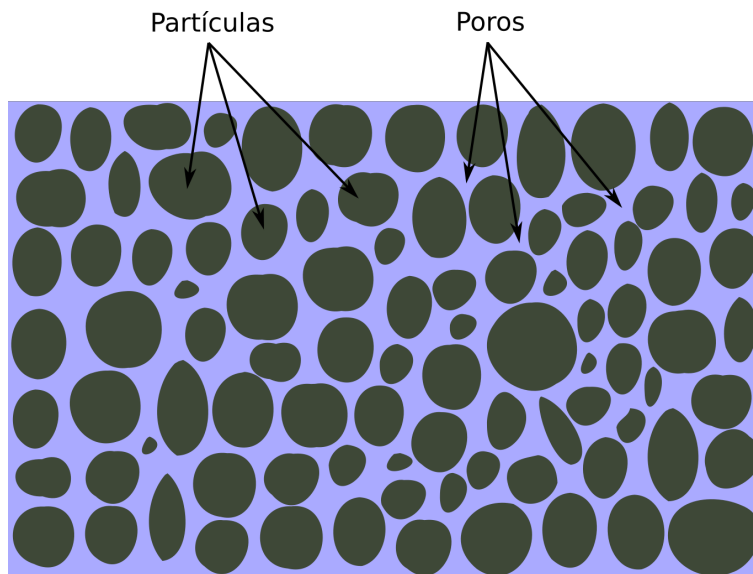


Figura 2.2: Medio poroso saturado.

2.2. Conceptos Básicos

Para entender mejor las ecuaciones gobernantes que describen el fenómeno de convección natural con doble difusividad en un medio poroso saturado, las cuales se muestran en la sección 3.6, a continuación se verán algunos conceptos básicos importantes.

2.2.1. Ecuaciones Diferenciales Hiperbólicas, Parabólicas y Elípticas

Definimos una Ecuación diferencial como una ecuación que relaciona una función y su variable (o sus variables) con sus derivadas.

Si una ecuación diferencial contiene derivadas respecto a una sola variable independiente se llama ecuación diferencial ordinaria, por otro lado, si contiene derivadas parciales respecto a dos o más variables independientes, se llama ecuación en derivadas parciales.

El orden de una ecuación diferencial está dado por el grado mayor de las derivadas que aparecen en la ecuación, por ejemplo si la ecuación presenta hasta una segunda derivada entonces es una ecuación de segundo orden.

El grado de una ecuación diferencial está dado por el mayor exponente al que se eleva una derivada, notando que en ecuaciones en derivadas parciales, al exponente se le suma un grado si se está multiplicando una derivada parcial por otra. Cabe destacar que una ecuación diferencial de grado uno es también llamada ecuación diferencial lineal ([Universidad Abierta y a Distancia de México et al., 2020](#)).

En general las ecuaciones lineales de segundo orden en dos variables tienen la forma:

$$Au_{xx} + Bu_{xy} + Cu_{yy} + Du_x + Eu_y + Fu + G = 0, \quad (2.1)$$

donde los coeficientes A, B, C, D, E, F, G son funciones reales definidas en una región $\Omega \subset \mathbb{R}^2$, y cumplen la condición $A^2 + B^2 + C^2 > 0$; $u_{xx} = \frac{\partial^2 u}{\partial x^2}$, $u_{yy} = \frac{\partial^2 u}{\partial y^2}$, $u_{xy} = \frac{\partial^2 u}{\partial x \partial y}$, $u_x = \frac{\partial u}{\partial x}$, $u_y = \frac{\partial u}{\partial y}$, $u = u(x, y)$. Es necesario hacer notar el parecido de la ecuación (2.1) con la ecuación general de las cónicas en el espacio \mathbb{R}^2 :

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0. \quad (2.2)$$

Mediante un cambio de coordenadas apropiado el discriminante $B^2 - 4AC$ permanece invariante respecto al signo, y la ecuación (2.2) puede simplificarse, es decir, tiene una propiedad intrínseca de la ecuación, la cual permite una clasificación de las cónicas de acuerdo al signo de $B^2 - 4AC$. Para la ecuación (2.1) ocurre algo similar a la ecuación (2.2).

Se define el discriminante I de la ecuación en derivadas parciales (2.1) como

$$I = B^2 - 4AC. \quad (2.3)$$

Se dice que la ecuación (2.1) es: Parabólica si $I = 0$, Elíptica si $I < 0$, e Hiperbólica si $I > 0$ ([López-Garza and Martínez-Ortiz, 2020](#)).

Las **ecuaciones parabólicas** generalmente tienen la forma

$$\frac{\partial u(x, t)}{\partial t} = \mathbb{L}[u(x, t)] \quad , \quad t > 0 \quad , \quad x \in \Omega \quad (\Omega \subset \mathbb{R}^1, \mathbb{R}^2, \mathbb{R}^3),$$

con la condición inicial

$$u(x, t) = f(x),$$

y la condición de contorno

$$\alpha \frac{\partial u}{\partial n} + \beta u = g(t) \quad \text{en} \quad \partial\Omega,$$

donde $f(x)$, $g(t)$ son dos funciones conocidas, α , β son constantes y $\partial\Omega$ es el contorno de Ω .

Las **ecuaciones hiperbólicas** generalmente tienen la forma

$$\frac{\partial^2 u(x, t)}{\partial t^2} = c^2 \mathbb{L}[u(x, t)] \quad , \quad t > 0 \quad , \quad x \in \Omega \quad (\Omega \subset \mathbb{R}^1, \mathbb{R}^2, \mathbb{R}^3),$$

con las condiciones iniciales

$$\begin{aligned} u(x, 0) &= f(x), \\ \frac{\partial u(x, 0)}{\partial t} &= h(x), \end{aligned}$$

y la condición de contorno

$$\alpha \frac{\partial u}{\partial n} + \beta u = g(t) \quad \text{en} \quad \partial\Omega,$$

donde c es constante, $f(x)$, $g(t)$, $h(x)$ son funciones conocidas, α , β son constantes y $\partial\Omega$ es el contorno de Ω .

En estas ecuaciones x se refiere a un vector de una, dos o tres dimensiones espaciales, y \mathbb{L} es un operador diferencial lineal parcial de segundo orden con coeficientes constantes en el espacio de variables (Benito et al., 2007).

2.2.2. Ley de Darcy

La Ley de Darcy describe, con base en experimentos de laboratorio, las características del movimiento del agua a través de un medio poroso.

La expresión matemática de la Ley de Darcy es la siguiente:

$$Q = k \frac{\Delta h}{\Delta l} A,$$

donde Q es el gasto o caudal, Δl es la longitud de la muestra, k es conocido como coeficiente de permeabilidad, A es el área de la sección transversal de la muestra, $\Delta h / \Delta l = (h_2 - h_1) / (x_2 - x_1)$ es el gradiente hidráulico (ver figura 2.3).

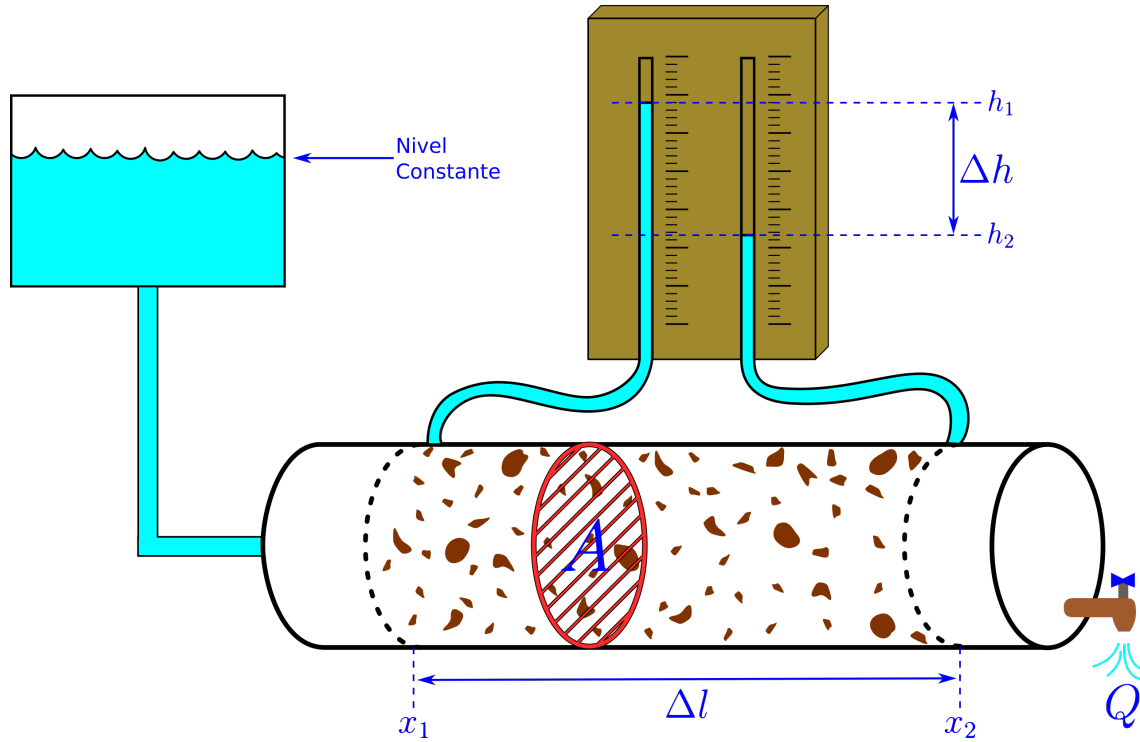


Figura 2.3: Esquema del experimento de Darcy.

El agua, por relaciones de energía, circula de mayor a menor altura piezométrica. Por lo cual se observa que el gradiente hidráulico adopta un valor negativo, por lo cual

$$\frac{\Delta h}{\Delta l} \leq 0,$$

con ello se puede expresar

$$q = \frac{Q}{A} = -k \frac{\partial h}{\partial x},$$

donde h es la altura piezométrica y x la longitud recorrida.

Generalizando a 3 dimensiones se obtiene

$$q = -\vec{K} \cdot \nabla (h(x, y, z)).$$

La ley de Darcy es válida en un medio poroso, saturado, continuo, homogéneo e isótropo y cuando las fuerzas inerciales son despreciables ($Re < 1$). Mantiene su validez para la mayoría de los tipos de flujo de fluidos en los suelos ([Universidad de Salamanca \(España\) and Sánchez-San-Román, 2020](#)).

2.2.3. Stream Function (Función de Flujo)

En un flujo plano de 2 dimensiones se tiene el vector de velocidad siguiente:

$$\vec{V} = v_x(x, y, t)\hat{\mathbf{i}} + v_y(x, y, t)\hat{\mathbf{j}}.$$

Cuando el fluido es incompresible ($\nabla \cdot \vec{V} = 0$) existe una función $\Psi = \Psi(x, y, t)$, denominada función de corriente (o también función de flujo) definida por

$$\begin{aligned} v_x &= \frac{\partial \Psi}{\partial y}, \\ v_y &= -\frac{\partial \Psi}{\partial x}, \end{aligned}$$

de forma que

$$d\Psi = -v_y dx + v_x dy.$$

Se puede observar que el campo de velocidades queda completamente definido una vez conocida la función $\Psi(x, y, t)$. Esta se puede determinar en cualquier punto mediante

$$\Psi = \int (v_x dy - v_y dx),$$

donde las constantes de integración se obtienen mediante las condiciones de frontera ([Universidad Nacional de Educación a Distancia, Madrid et al., 2020](#)).

2.2.4. Grupos Adimensionales

El *Análisis Dimensional* es una herramienta que nos ayuda a simplificar problemas complejos a unos más simples (o en otras palabras más económicos). El objetivo del análisis dimensional es relacionar los valores de las distintas variables implicadas en un sistema físico, de manera que estas relaciones queden con unidades adimensionales, a estas relaciones se les conoce como **Grupos adimensionales** (o también nombrados como números adimensionales). Dentro de las variables que se analizan están aquellas que describen la geometría del problema, la propiedades termodinámicas, y las propiedades de los materiales (sólidos o fluidos) involucrados.

Si los grupos adimensionales se conservan constantes entonces el problema físico puede ser escalado (también llamado escalamiento¹), ya que al mantener los grupos adimensionales también se mantienen las relaciones entre todas las variables que describen un problema.

A continuación se describen brevemente los grupos adimensionales presentes en el fenómeno de convección natural con doble difusividad en un medio poroso saturado.

Número de Lewis

El *Número de Lewis* es un número adimensional, llamado así por Warren K. Lewis (1882-1975). El número de Lewis se define como el cociente entre la difusividad térmica y la difusividad másica. Se utiliza para caracterizar flujos de fluidos en donde hay procesos simultáneos de transferencia de calor y masa por convección. La ecuación que describe este número es la siguiente:

$$Le = \frac{\alpha}{D}.$$

¹El escalamiento es muy útil cuando se requiere probar proyectos de ingeniería de gran magnitud, pero se desea verificar el comportamiento en una mesa de laboratorio.

Número de Prandtl

La transferencia de calor puede surgir por dos procesos llamados conducción y convección. En los fluidos ambos procesos pueden estar presentes simultáneamente. Como ambos reducen la diferencia de temperatura (ΔT) se puede considerar que compiten entre ellos en el transporte de calor. Las tasas de conducción y convección varían en diferentes fluidos (aire, agua, aceite, mercurio, etc.), en algunas ocasiones la conducción domina, mientras que en otras es la convección. El *Número de Prandtl* es un parámetro que puede ser usado para determinar de manera aproximada cual proceso es el de mayor importancia. La ecuación que describe este número es la siguiente:

$$Pr = \frac{\nu}{\alpha},$$

donde ν es la viscosidad cinemática, y α es la difusividad térmica.

También se puede considerar al número de Prandtl como un cociente entre la velocidad de difusión de la cantidad de momento (ν) y la velocidad de difusión de la energía térmica (α). Si un fluido es mas viscoso, el número de Prandtl incrementa, por lo que el transporte de calor será menos por convección y más por conducción (The Concord Consortium and Xie, 2020).

Número de Rayleigh

El *Número de Rayleigh* surge de comparar el flujo de calor convectivo ($\beta g \Delta T^2 h^2 / \nu$) con el flujo de calor conductivo ($\alpha \Delta T / h$), de tal forma se obtiene la relación:

$$Ra = \frac{\beta g \Delta T^2 h^2 / \nu}{\alpha \Delta T / h} = \frac{\beta g \Delta T h^3}{\nu \alpha} = Gr \cdot Pr,$$

donde Gr es el número de Grashov, Pr es el número de Prandtl, g es la aceleración de la gravedad, β es el coeficiente de expansión térmica, ν es la viscosidad cinemática, α es la difusividad térmica, ΔT es la diferencia de temperatura en la lámina de fluido, h es el espesor de la lámina (ver figura 2.4).

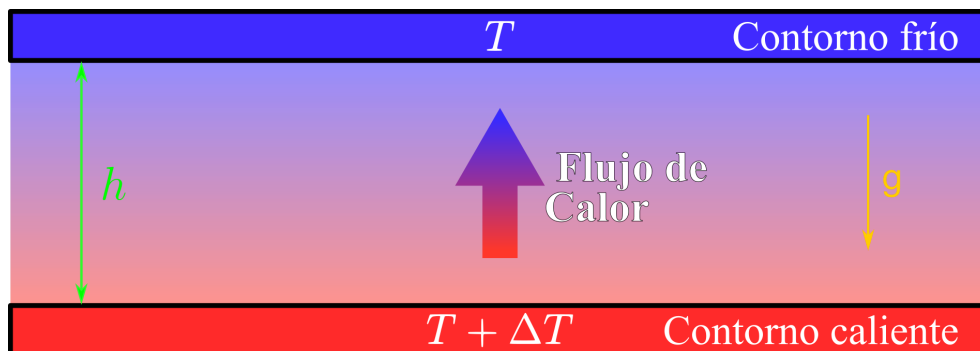


Figura 2.4: Esquema de una lámina de flujo que transporta calor (conducción y/o convección), desde una capa de contorno caliente a una capa de contorno fría.

La conveccion ocurre cuando este número excede un valor crítico, el cual depende de la naturaleza de las condiciones de contorno. Para un fluido confinado entre dos placas rígidas el valor

crítico del número de Rayleigh es $Ra = 1708$. A valores un poco por arriba del límite, la convección se organiza a si misma en rollos bidimensionales paralelos o en celdas hexagonales empaquetadas. A valores mucho más altos del número de Rayleigh se desarrollan movimientos erráticos que dependen del tiempo, y la convección parece estar mucho menos organizada ([Cushman-Roisin and Beckers, 2011](#)).

Número de Grashof

El *Número de Grashof* es una cantidad adimensional usada para analizar la distribución de velocidad en sistemas con convección libre (convección natural). En la convección libre la fuerza impulsora (driving force) es la fuerza de flotación causada por un gradiente de temperatura, ya que el fluido estará en reposo en ausencia de variaciones de temperatura.

Esencialmente el número de Grashof es un cociente entre las fuerzas de flotación y las fuerzas viscosas,

$$Gr = \frac{\text{fuerza de flotacion}}{\text{fuerza viscosa}} = \frac{g\Delta\rho V}{\rho\nu^2},$$

donde ρ es la densidad, $\Delta\rho$ es el cambio de densidad entre la región de temperatura alta y la región de temperatura baja, $V = h^3$ es el volumen que contiene al flujo.

El coeficiente de expansión térmica β se puede aproximar de la siguiente manera:

$$\beta = \frac{d\ln\rho}{dT} \approx \frac{1}{\rho} \frac{\Delta\rho}{\Delta T}.$$

Entonces tenemos

$$Gr = \frac{g\beta\Delta Th^3}{\nu^2}.$$

Cuando $Gr \gg 1$ la fuerza viscosa es despreciable en comparación con la flotabilidad y las fuerzas inerciales. Cuando las fuerzas de flotación superan a las fuerzas viscosas el flujo comienza a tener una transición de flujo a régimen turbulento ([Rice University - Houston - Texas et al., 2020](#)).

Tasa de Flotabilidad (Buoyancy Ratio)

La tasa de flotabilidad N es un cociente entre el Número de Rayleigh Térmico Ra_T y el Número de Rayleigh de Concentración Ra_C :

$$N = \frac{Ra_T}{Ra_C},$$

donde

$$Ra_T = \frac{g\beta_T\Delta Th^3}{\nu\alpha},$$

$$Ra_C = \frac{g\beta_C\Delta Ch^3}{\nu\alpha},$$

donde β_T es el coeficiente de expansión térmica, y β_C es el coeficiente de expansión debido a la concentración.

Capítulo 3

METODOLOGÍA

A continuación veremos la definición de la expansión en serie de Taylor, y cómo se deriva a partir de ella el método de diferencias finitas. También se usará esta serie de manera general para obtener el método de diferencias finitas generalizadas en una y dos dimensiones, el cual es un método muy usado en la literatura. Posteriormente se explicará el esquema GFDM propuesto para este trabajo, el cual tiene algunas diferencias con el método usado generalmente en la literatura. Después se verá el sistema de ecuaciones diferenciales que describen el problema de convección natural con doble difusividad en estado estacionario en un medio poroso saturado dentro de un paralelogramo. A continuación se describirá como se hará la discretización del espacio bidimensional correspondiente al paralelogramo, el cual corresponde al domino donde se resolverá el sistema de ecuaciones. Por último se verá como se hizo la linealización del problema (desacoplamiento de las ecuaciones diferenciales del sistema) para poder resolver con el algoritmo predictor corrector.

3.1. Expansión en Serie de Taylor

Si $u(x)$ es una función de buen comportamiento² alrededor de un punto de expansión elegido x_0 , entonces la función se puede expandir en una serie llamada *serie de Taylor* definida por:

$$\begin{aligned} u(x) &= \sum_{n=0}^{\infty} \frac{1}{n!} \left. \frac{d^n u(x)}{dx^n} \right|_{x=x_0} (x - x_0)^n \\ &= u(x_0) + \left. \frac{du(x)}{dx} \right|_{x=x_0} (x - x_0) + \frac{1}{2!} \left. \frac{d^2 u(x)}{dx^2} \right|_{x=x_0} (x - x_0)^2 + \dots \end{aligned} \quad (3.1)$$

La serie de Taylor de una función es a menudo útil en situaciones físicas donde se desee obtener el valor aproximado de la función alrededor del punto de expansión x_0 . Se evalúan todos los términos en el punto de expansión, los cuales dependen de las derivadas de la función. Es típico que los términos sucesivos en una aproximación en serie de una función sean cada vez más y más pequeños, y según el problema que se esté trabajando se puede considerar que a partir de cierto término pueden despreciarse todos los términos subsecuentes, y así uno puede juzgar que se ha logrado una aproximación suficientemente exacta de la función (Olmo and Nave, 2020).

²Buen comportamiento significa que tanto la función como sus derivadas son continuas y están definidas en el intervalo de expansión alrededor de la coordenada x_0 .

Es frecuente el caso donde un conveniente punto de expansión es $x_0 = 0$, y la serie alrededor de este punto de expansión especial se le llama también serie de Maclaurin. Hay muchas aplicaciones para expansiones de funciones comunes alrededor de $x = 0$; algunos ejemplos son:

$$\begin{aligned} e^x &= 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots, \\ \sin(x) &= x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots, \\ \cos(x) &= 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots. \end{aligned}$$

3.2. Diferencias Finitas

El método de diferencias finitas (FDM¹.) es uno de los métodos usados para resolver ecuaciones diferenciales que son, en general, difíciles o imposibles de resolver analíticamente. La ecuación en la que se basa este método es aquella que define la derivada:

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}. \quad (3.2)$$

Este límite nos lleva al cálculo infinitesimal. Si dejamos que h sea de un tamaño finito, entonces tenemos una aproximación a la derivada como

$$f'(x) \cong \frac{f(x+h) - f(x)}{h}. \quad (3.3)$$

Si hacemos una discretización del dominio $[0, 1]$ desde $x_0 = 0$ hasta $x_N = 1$, donde N es el número de nodos, entonces existen en el dominio $N - 1$ intervalos de tamaño finito, como se muestra en la figura 3.1,

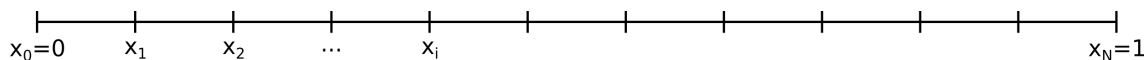


Figura 3.1: Discretización del dominio $[0, 1]$ en el eje x .

donde podemos expresar el valor x_i como $x_i = ih$ con $i = 0, 1, 2, \dots, N$, y donde $h = \frac{1}{N}$.

Entonces podemos escribir una aproximación a la derivada de una función $u(x)$ como

$$\begin{aligned} u'(x) &= \frac{u(x_{i+1}) - u(x_i)}{h} + O(h) \\ &\approx \frac{u_{i+1} - u_i}{h}, \end{aligned} \quad (3.4)$$

donde $O(h)$ representa el error de esta aproximación, el cual es de primer orden, o lineal, debido a que el error decrece linealmente al decrecer el paso h . Cabe mencionar que si el error dependiera de h^2 entonces sería de orden dos. También es necesario mencionar que $u(x_{i+1})$ representa el valor

¹Por sus siglas en inglés, Finite Differences Method

real de la función, mientras que u_{i+1} representa la aproximación de la función debido a que se ha despreciado el error. Es importante destacar que la fórmula de la diferencia finita anterior es también conocida como *derivada en diferencias hacia adelante*.

También podemos llegar a la misma aproximación utilizando la expansión en serie de Taylor, ecuación (3.1), sobre el nodo x_{i+1} :

$$u(x_{i+1}) = u(x_i) + h \frac{\partial u}{\partial x} + O(h^2).$$

Despejando entonces tenemos

$$\begin{aligned} \frac{u(x_{i+1}) - u(x_i)}{h} &= u'(x_i) + O(h), \\ u'(x_i) &\approx \frac{u_{i+1} - u_i}{h}. \end{aligned} \quad (3.5)$$

Se puede ver que la ecuación (3.4) y la ecuación (3.5) son la misma. Otra aproximación es la llamada *derivada en diferencias hacia atrás*, la cual se obtiene de hacer la expansión en serie de Taylor ahora en el nodo x_{i-1} :

$$u(x_{i-1}) = u(x_i) - hu'(x_i) + O(h^2).$$

Despejando entonces tenemos

$$\begin{aligned} \frac{u(x_{i-1}) - u(x_i)}{-h} &= u'(x_i) + O(h), \\ u'(x_i) &\approx \frac{u_i - u_{i-1}}{h}. \end{aligned} \quad (3.6)$$

También podemos obtener aproximaciones de orden superior utilizando la expansión en serie de Taylor, ecuación (3.1), para lo cual es necesario primero aplicar la expansión a la función u en los puntos x_{i+1} y x_{i-1} , lo cual resulta como:

$$\begin{aligned} u(x_{i+1}) &= u(x_i + h) = u(x_i) + u'(x_i)h + u''(x_i)\frac{h^2}{2!} + u'''(x_i)\frac{h^3}{3!} + u^{IV}(\xi_i)\frac{h^4}{4!}, \\ u(x_{i-1}) &= u(x_i - h) = u(x_i) - u'(x_i)h + u''(x_i)\frac{h^2}{2!} - u'''(x_i)\frac{h^3}{3!} + u^{IV}(\eta_i)\frac{h^4}{4!}. \end{aligned}$$

Si se suman las dos ecuaciones anteriores entonces obtenemos:

$$u(x_{i+1}) + u(x_{i-1}) = 2u(x_i) + 2u''(x_i)\frac{h^2}{2} + \frac{1}{4!}[u^{IV}(\xi_i) + u^{IV}(\eta_i)]h^4,$$

despejando

$$\begin{aligned} \frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1}))}{h^2} &= u''(x_i) + \frac{1}{4!}[u^{IV}(\xi_i) + u^{IV}(\eta_i)]h^2 \\ u''(x_i) &= \frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1}))}{h^2} + O(h^2) \\ u''(x_i) &\approx \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}, \end{aligned} \quad (3.7)$$

donde se ha considerado el error como $O(h^2) = \frac{1}{4!}[u^{IV}(\xi_i) + u^{IV}(\eta_i)]h^2$, y también que existe una constante M tal que $O(h^2) \leq Mh^2$, o en otras palabras, que el error $O(h^2)$ tiene un tamaño finito.

Si ahora se restan las dos ecuaciones, entonces se tiene

$$\begin{aligned} u(x_{i+1}) - u(x_{i-1}) &= 2u'(x_i)h + O(h^3), \\ u'(x_i) &= \frac{u(x_{i+1}) - u(x_{i-1})}{2h} + O(h^2), \\ u'(x_i) &\approx \frac{u_{i+1} - u_{i-1}}{2h}, \end{aligned} \tag{3.8}$$

donde la ecuación (3.8) es conocida como *derivada en diferencias centrada*, la cual tiene error de orden h^2 .

3.3. Diferencias Finitas Generalizadas (GFD)

El Método de Diferencias Finitas Generalizadas (GFDM²) es un método usado para la resolución de sistemas de ecuaciones diferenciales de manera numérica. Dos ejemplos notables de ecuaciones diferenciales, que son usadas en muchos ámbitos para modelar diferentes fenómenos, son las *ecuaciones parabólicas* y las *ecuaciones hiperbólicas* (mencionadas anteriormente en la sección 2.2), cuyo comportamiento aparece en nuestro problema de interés.

Este método se deriva del método de diferencias finitas, con la diferencia de que los nodos vecinos no están siempre a la misma distancia (en el caso anterior esta distancia es llamada h), sino que para cada nodo se calcula numéricamente esta distancia. Esto se puede generalizar en dos dimensiones, y se desarrollará la teoría de este método en las secciones 3.4 y 3.5.

3.4. Esquema GFD Generalmente Usado en la Literatura

Se considera la siguiente ecuación para resolver en el dominio espacial, utilizando GFD

$$\mathbb{L}[U] = 0 \quad \text{in } \Omega,$$

con la condición de contorno

$$\alpha \frac{\partial U}{\partial n} + \beta U = g \quad \text{in } \partial\Omega.$$

Para la solución del problema se requiere una nube de puntos sobre los cuales se hará la aproximación numérica, para cada nodo se hará una aproximación tomando en cuenta varios de sus nodos vecinos, estos nodos son denominados *nodos de soporte de estrella*. La elección de los nodos de soporte en la literatura es resuelto, generalmente, utilizando el *criterio cruzado*. Por ejemplo, en dos dimensiones el área alrededor del nodo central es dividido en cuatro sectores correspondientes a los cuadrantes del sistema cartesiano de coordenadas con origen en el nodo central como se muestra en la figura 3.2. En cada sector dos o más nodos son seleccionados, se suelen usar los más cercanos al origen.

²Por sus siglas en inglés, Generalized Finite Differences Method.

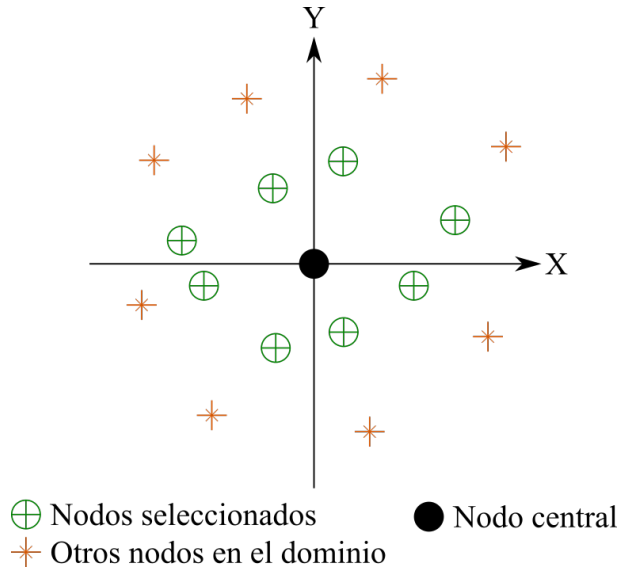


Figura 3.2: Criterio de los cuatro cuadrantes con dos nodos en cada cuadrante.

Si U_0 es el valor de la función en el nodo central de la estrella, y U_i son los valores de la función en el resto de los nodos, con $i = 1, \dots, N$, entonces, de acuerdo a la expansión en serie de Taylor en 1D y 2D respectivamente tenemos:

$$U_i = U_0 + h_i \left. \frac{\partial U}{\partial x} \right|_{x_0} + \frac{1}{2} \left(h_i^2 \left. \frac{\partial^2 U}{\partial x^2} \right|_{x_0} \right) + \dots,$$

$$U_i = U_0 + h_i \left. \frac{\partial U}{\partial x} \right|_{x_0} + k_i \left. \frac{\partial U}{\partial y} \right|_{x_0} + \frac{1}{2} \left(h_i^2 \left. \frac{\partial^2 U}{\partial x^2} \right|_{x_0} + k_i^2 \left. \frac{\partial^2 U}{\partial y^2} \right|_{x_0} + 2h_i k_i \left. \frac{\partial^2 U}{\partial x \partial y} \right|_{x_0} \right) + \dots,$$

donde x_0 o (x_0, y_0) son las coordenadas del nodo central, x_i o (x_i, y_i) son las coordenadas del i -ésimo nodo en la estrella, y $h_i = x_i - x_0$, $k_i = y_i - y_0$ son las distancias (horizontal y vertical, respectivamente) del nodo central al nodo i -ésimo.

Si los términos de tercer orden son ignorados entonces se obtiene una aproximación de segundo orden para la función U_i denotada como u_i . Es posible entonces definir las funciones $B_2(u)$ en una dimensión y $B_5(u)$ en dos dimensiones, las cuales definen el residual de nuestra aproximación:

$$B_2(u) = \sum_{i=1}^N \left(\left(u_0 - u_i + h_i \left. \frac{\partial u}{\partial x} \right|_{x_0} + \frac{h_i^2}{2} \left. \frac{\partial^2 u}{\partial x^2} \right|_{x_0} \right) \omega(h_i) \right)^2,$$

$$B_5(u) = \sum_{i=1}^N \left(\left(u_0 - u_i + h_i \left. \frac{\partial u}{\partial x} \right|_{x_0} + k_i \left. \frac{\partial u}{\partial y} \right|_{x_0} + \frac{h_i^2}{2} \left. \frac{\partial^2 u}{\partial x^2} \right|_{x_0} \right. \right. \\ \left. \left. + \frac{k_i^2}{2} \left. \frac{\partial^2 u}{\partial y^2} \right|_{x_0} + h_i k_i \left. \frac{\partial^2 u}{\partial x \partial y} \right|_{x_0} \right) \omega(h_i, k_i) \right)^2,$$

donde $\omega(h_i)$ y $\omega(h_i, k_i)$ son denominados funciones peso en 1D y en 2D respectivamente.

Si la norma de la ecuación que define a $B(u)$ se minimiza respecto a las derivadas parciales, se

obtienen los siguientes sistemas de ecuaciones para 1D y 2D:

$$\begin{aligned} A_2 Du_2 &= b_2, \\ A_5 Du_5 &= b_5, \end{aligned}$$

donde los vectores Du_2 y Du_5 están dados respectivamente por:

$$Du_2 = \begin{bmatrix} \frac{\partial u}{\partial x} \Big|_{x_0} \\ \frac{\partial^2 u}{\partial x^2} \Big|_{x_0} \end{bmatrix}, \quad Du_5 = \begin{bmatrix} \frac{\partial u}{\partial x} \Big|_{x_0} \\ \frac{\partial u}{\partial y} \Big|_{x_0} \\ \frac{\partial^2 u}{\partial x^2} \Big|_{x_0} \\ \frac{\partial^2 u}{\partial y^2} \Big|_{x_0} \\ \frac{\partial^2 u}{\partial x \partial y} \Big|_{x_0} \end{bmatrix},$$

las matrices A_2 y A_5 están dadas por:

$$A_2 = \begin{bmatrix} \sum_{i=1}^N h_i^2 w_i^2 & \sum_{i=1}^N \frac{h_i^3}{2} w_i^2 \\ SYM & \sum_{i=1}^N \frac{h_i^4}{4} w_i^2 \end{bmatrix},$$

$$A_5 = \begin{bmatrix} \sum_{i=1}^N h_i^2 w_i^2 & \sum_{i=1}^N h_i k_i w_i^2 & \sum_{i=1}^N \frac{h_i^3}{2} w_i^2 & \sum_{i=1}^N \frac{h_i k_i^2}{2} w_i^2 & \sum_{i=1}^N h_i^2 k_i w_i^2 \\ \sum_{i=1}^N h_i^2 w_i^2 & \sum_{i=1}^N k_i^2 w_i^2 & \sum_{i=1}^N \frac{h_i^2 k_i}{2} w_i^2 & \sum_{i=1}^N \frac{k_i^3}{2} w_i^2 & \sum_{i=1}^N h_i k_i^2 w_i^2 \\ & SYM & \sum_{i=1}^N \frac{h_i^4}{4} w_i^2 & \sum_{i=1}^N \frac{h_i^2 k_i^2}{4} w_i^2 & \sum_{i=1}^N \frac{h_i^3 k_i}{2} w_i^2 \\ & & & \sum_{i=1}^N \frac{k_i^4}{4} w_i^2 & \sum_{i=1}^N \frac{h_i k_i^3}{2} w_i^2 \\ & & & & \sum_{i=1}^N h_i^2 k_i^2 w_i^2 \end{bmatrix},$$

y por último los vectores b^2 y b^5 están dados por [(Benito et al., 2007),(Li et al., 2018)]:

$$b_2 = \begin{bmatrix} \sum_{i=1}^N (u_0 - u_i) h_i w_i^2 \\ \sum_{i=1}^N (u_0 - u_i) \frac{h_i^2}{2} w_i^2 \end{bmatrix}, \quad b_5 = \begin{bmatrix} \sum_{i=1}^N (u_0 - u_i) h_i w_i^2 \\ \sum_{i=1}^N (u_0 - u_i) k_i w_i^2 \\ \sum_{i=1}^N (u_0 - u_i) \frac{h_i^2}{2} w_i^2 \\ \sum_{i=1}^N (u_0 - u_i) \frac{k_i^2}{2} w_i^2 \\ \sum_{i=1}^N (u_0 - u_i) h_i k_i w_i^2 \end{bmatrix}.$$

Nota: Es importante mencionar que el planteamiento de este esquema de diferencias finitas generalizadas (GFDM) está mal condicionado, ya que, pequeñas variaciones en las funciones de peso ω puede causar una variación muy grande en las soluciones obtenidas.

3.5. Esquema GFD Propuesto

Antes de la existencia del Método de Diferencias Finitas Generalizadas, la manera de resolver un problema era transformar el dominio físico en un dominio computacional sencillo de resolver, con el cual se obtenía una solución usando diferencias clásicas. Una vez obtenida la solución en el dominio computacional se volvían a transformar tanto el dominio como la solución al dominio físico.

Lo que hace el método de GFDM es obtener una solución directamente sobre el dominio físico, sin necesidad de transformarlo a un dominio computacional. Otra ventaja importante de destacar es que este método ha demostrado ser muy robusto para resolver diversos problemas.

A continuación se describe el método de GFDM implementado en este trabajo. Tomemos las coordenadas de un nodo en una malla como:

$$p_i = (x_i, y_i).$$

Usando expansión en serie de Taylor para una función u en la dirección de un nodo con coordenadas p_0 hacia un nodo con coordenadas p_i , y multiplicando todos los términos por la función de peso en el nodo i -ésimo Γ_i , se tiene lo siguiente:

$$\Gamma_i u(p_i) = \Gamma_i \left(u(p_0) + h_i \frac{\partial u}{\partial x} \Big|_{p_0} + k_i \frac{\partial u}{\partial y} \Big|_{p_0} + \frac{h_i^2}{2} \frac{\partial^2 u}{\partial x^2} \Big|_{p_0} + h_i k_i \frac{\partial^2 u}{\partial x \partial y} \Big|_{p_0} + \frac{k_i^2}{2} \frac{\partial^2 u}{\partial y^2} \Big|_{p_0} \right) + \dots,$$

donde $h_i = x_i - x_0$ y $k_i = y_i - y_0$.

3.5.1. Aproximación a un operador diferencial lineal \mathbb{L}

El operador diferencial lineal \mathbb{L} (de manera general) aplicado sobre la función u , evaluado en el nodo de coordenadas p_0 es

$$\mathbb{L}[u] = Bu + C \frac{\partial u}{\partial x} \Big|_{p_0} + D \frac{\partial u}{\partial y} \Big|_{p_0} + E \frac{\partial^2 u}{\partial x^2} \Big|_{p_0} + F \frac{\partial^2 u}{\partial x \partial y} \Big|_{p_0} + G \frac{\partial^2 u}{\partial y^2} \Big|_{p_0}.$$

Se puede aproximar este operador como

$$\sum_{i=0}^q \Gamma_i u(p_i) \cong \mathbb{L}[u],$$

donde q es el número de nodos vecinos, o nodos de soporte.

Restando la aproximación y el valor real obtenemos el error de truncamiento τ . Tenemos entonces

$$\tau = \sum_{i=0}^q \Gamma_i u(p_i) - \mathbb{L}[u].$$

Si sustituimos el valor de la aproximación de la serie de Taylor y los términos del operador \mathbb{L} , entonces obtenemos

$$\begin{aligned} \tau = & \left(\sum_i \Gamma_i - B \right) u(p_0) + \left(\sum_i \Gamma_i h_i - C \right) \frac{\partial u}{\partial x} \Big|_{p_0} + \left(\sum_i \Gamma_i k_i - D \right) \frac{\partial u}{\partial y} \Big|_{p_0} \\ & + \left(\frac{1}{2} \sum_i \Gamma_i h_i^2 - E \right) \frac{\partial^2 u}{\partial x^2} \Big|_{p_0} + \left(\sum_i \Gamma_i h_i k_i - F \right) \frac{\partial^2 u}{\partial x \partial y} \Big|_{p_0} + \left(\frac{1}{2} \sum_i \Gamma_i k_i^2 - G \right) \frac{\partial^2 u}{\partial y^2} \Big|_{p_0} + O(h_i^3, k_i^3). \end{aligned}$$

Queremos que el error sea cero, por lo que hacemos cada término de la expresión igual a cero, de lo cual se obtienen las siguientes ecuaciones:

$$\begin{aligned}\sum_i \Gamma_i &= B, \\ \sum_i \Gamma_i h_i &= C, \\ \sum_i \Gamma_i k_i &= D, \\ \sum_i \Gamma_i h_i^2 &= 2E, \\ \sum_i \Gamma_i h_i k_i &= F, \\ \sum_i \Gamma_i k_i^2 &= 2G,\end{aligned}$$

o expresado en forma matricial, y considerando que el nodo central cuenta con q nodos vecinos, se puede escribir como

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ 0 & h_1 & \cdots & h_q \\ 0 & k_1 & \cdots & k_q \\ 0 & h_1^2 & \cdots & h_q^2 \\ 0 & h_1 k_1 & \cdots & h_q k_q \\ 0 & k_1^2 & \cdots & k_q^2 \end{pmatrix} \begin{pmatrix} \Gamma_0 \\ \Gamma_1 \\ \cdot \\ \cdot \\ \cdot \\ \Gamma_q \end{pmatrix} = \begin{pmatrix} B \\ C \\ D \\ 2E \\ F \\ 2G \end{pmatrix}.$$

Definiendo las matrices M , Γ y L como

$$M = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 0 & h_1 & \cdots & h_q \\ 0 & k_1 & \cdots & k_q \\ 0 & h_1^2 & \cdots & h_q^2 \\ 0 & h_1 k_1 & \cdots & h_q k_q \\ 0 & k_1^2 & \cdots & k_q^2 \end{pmatrix}, \quad \Gamma = \begin{pmatrix} \Gamma_0 \\ \Gamma_1 \\ \cdot \\ \cdot \\ \cdot \\ \Gamma_q \end{pmatrix}, \quad L = \begin{pmatrix} B \\ C \\ D \\ 2E \\ F \\ 2G \end{pmatrix},$$

podemos escribir el sistema como [(Chávez-Negrete et al., 2018), (Domínguez-Mota et al., 2014)]:

$$M\Gamma = L.$$

Es importante mencionar que el sistema tiene q incógnitas en el vector Γ debido a que en cada nodo vecino se requiere una función de peso Γ_i , mientras que se cuenta con 6 ecuaciones independientes, representadas por los renglones de la matriz M , debido a que consideramos 6 términos dentro de la expansión en serie de Taylor; si el número de nodos vecinos es mayor que 6 entonces habrá más incógnitas Γ_i que ecuaciones independientes, por lo que el sistema de ecuaciones estará subdeterminado.

Para poder continuar con el cálculo de las soluciones de nuestro problema, se ha optado por obtener la pseudoinversa de la matriz M y así poder resolver el sistema para Γ . Sin embargo la

pseudoinversa puede no ser adecuada para cualquier malla, sino que es necesario que dicha malla tenga una estructuración correcta. Por otro lado en una malla con simetría, se espera que los pesos Γ_i de los nodos simétricos sean iguales entre sí, por lo que el número de incógnitas a encontrar se ve reducido, y por ende el sistema podría llegar a dejar de estar subdeterminado, y entonces el cálculo de la pseudoinversa sería adecuado.

Otra manera de tener más ecuaciones independientes dentro del sistema $M\Gamma = L$, es usar más términos de la serie de Taylor. Se decidió conservar solamente los términos hasta segundo orden, para que el error de la aproximación fuera de orden h^2 , de tal manera que si reducimos el valor de h , por ejemplo por la mitad ($h/2$), entonces el error se reduzca como el cuadrado, en este caso como un cuarto ($h^2/4$).

Cabe destacar que este método puede no ser adecuado para cualquier tipo de malla, ya que si los nodos se alinean verticalmente, entonces la diferencia de distancia en x entre el nodo i -ésimo y el nodo central, sería cero, y si esto ocurre entonces es imposible calcular una aproximación para las derivadas en x : u_x, u_{xx}, u_{xy} . De igual manera si los nodos de soporte se alinean horizontalmente con el nodo central, entonces no es posible aproximar ningún tipo de derivada respecto de y : u_y, u_{yy}, u_{xy} ; esto es debido a que la diferencia de distancia entre nodos en este eje, sería cero.

3.6. Ecuaciones Gobernantes

Las ecuaciones gobernantes que describen el fenómeno que estamos estudiando (Convección Natural con Doble Difusividad en un Medio Poroso Saturado en un Dominio en Forma de Paralelogramo) se obtienen considerando la relación entre presión y velocidad del flujo dado por la ley de Darcy y definiendo la función de corriente $\Psi(x, y)$, descritas en la sección 2.2, con lo cual se obtiene:

$$\frac{\partial^2 \Psi}{\partial x^2} + \frac{\partial^2 \Psi}{\partial y^2} + Ra \frac{\partial}{\partial x} (T + NC) = 0, \quad (x, y) \in \Omega, \quad (3.9)$$

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} - \frac{\partial \Psi}{\partial y} \frac{\partial T}{\partial x} + \frac{\partial \Psi}{\partial x} \frac{\partial T}{\partial y} = 0, \quad (x, y) \in \Omega, \quad (3.10)$$

$$\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} - Le \frac{\partial \Psi}{\partial y} \frac{\partial C}{\partial x} + Le \frac{\partial \Psi}{\partial x} \frac{\partial C}{\partial y} = 0, \quad (x, y) \in \Omega, \quad (3.11)$$

donde $T(x, y)$ es la temperatura adimensional, $C(x, y)$ es la concentración adimensional, Ra es el número de Rayleigh, Le es el número de Lewis, N es la tasa de flotabilidad, (x, y) son las coordenadas espaciales, y Ω es el dominio computacional. Discusión detallada de las ecuaciones anteriores puede ser encontrada en estudios previos [(Costa, 2004), (Nield et al., 2006)].

El diagrama esquemático para el problema en dos dimensiones de convección natural con doble difusividad en un paralelogramo lleno con un medio poroso saturado es ilustrado en la figura 3.3. La altura H y la longitud L del paralelogramo son mostrados en la figura 3.3(a) mientras que las condiciones de contorno para la función de corriente, temperatura y concentración, son descritas en la figura 3.3(b).

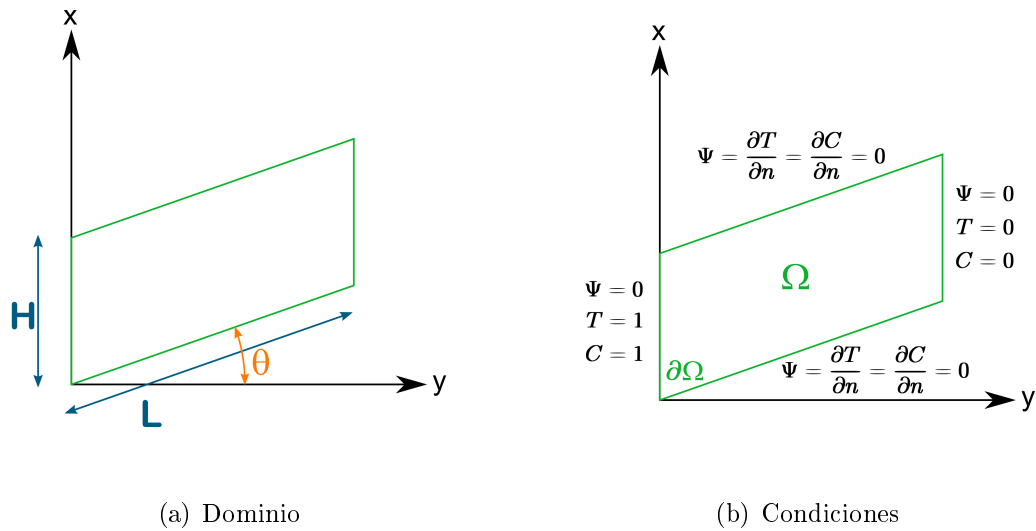


Figura 3.3: Diagramas esquemáticos de (a) el dominio computacional y (b) las condiciones de contorno.

3.7. Discretización del espacio

Para una primera aproximación se generó una malla de 1111 nodos usando la herramienta de MATLAB “pdetool”. Las coordenadas (x, y) de los nodos que describen los vértices del paralelogramo son $(0, 0)$, $(1.5, 0.4)$, $(1.5, 1)$, $(0, 0.6)$. Se puede ver dicha malla en la figura 3.4.

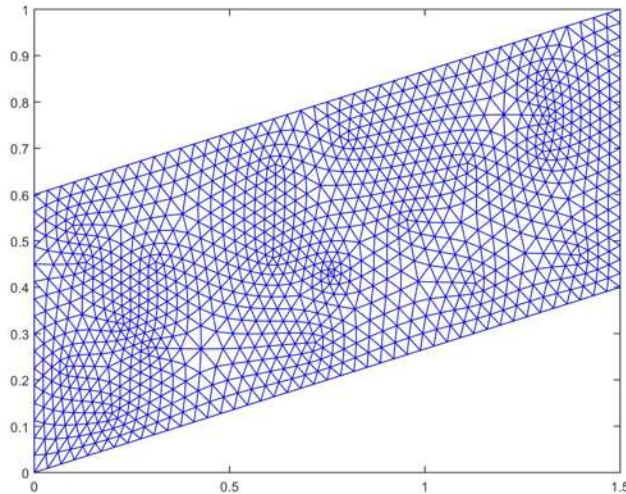


Figura 3.4: Malla utilizada en la primera aproximación

Esta es una malla triangular con una estructura irregular, de la cual se tomarán los seis nodos vecinos más cercanos, como nodos de soporte, para aproximar la solución en el nodo central. En

el caso donde el número de nodos vecinos es menor que seis, además de seleccionar los nodos vecinos también se seleccionan los vecinos de los nodos anteriormente seleccionados, esto se ilustra mejor en la figura 3.5.

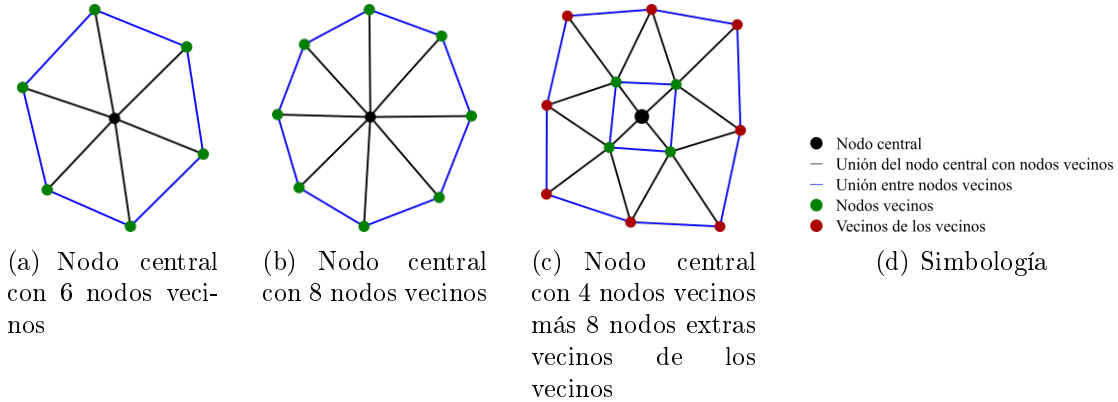


Figura 3.5: Diferentes estructuras en la malla

Para la siguiente aproximación se utilizó el mismo paralelogramo, pero con la diferencia de que se generó una malla de pequeños paralelogramos regulares, los cuales se crearon dividiendo las longitudes del paralelogramo en x y en y en tramos iguales, esto se hizo con el comando *linspace*, una vez conseguidos los nodos internos de los paralelogramos regulares se obtuvo una triangulación con estos nodos utilizando el comando *delaunay*, el cual crea una triangulación de nodos utilizando el algoritmo de triangulación de Delaunay; se puede ver esta malla en la figura 3.6.

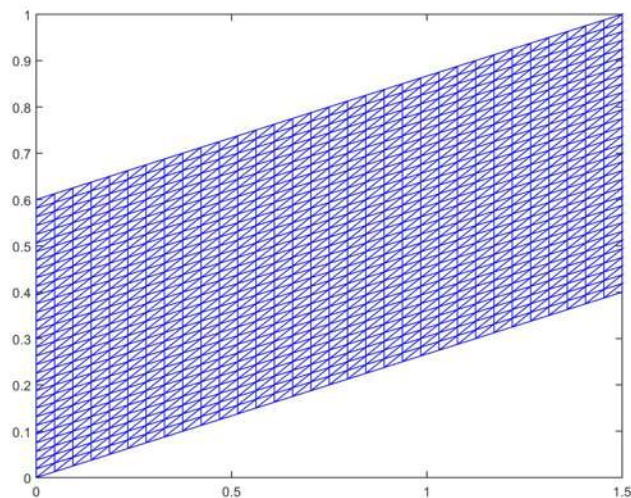


Figura 3.6: Malla utilizada en la segunda aproximación

Para las siguientes simulaciones se creó un código de MatLAB que genera mallas dentro de un paralelogramo que se divide en paralelogramos más pequeños y regulares; además estas mallas tienen una triangulación de Delaunay. Las mallas generadas son creadas de manera automática

para valores dados de altura H y longitud L del paralelogramo y también un valor dado para el ángulo de inclinación θ . Algunos ejemplos de mallas creadas utilizando este código se muestran en la figura 3.7.

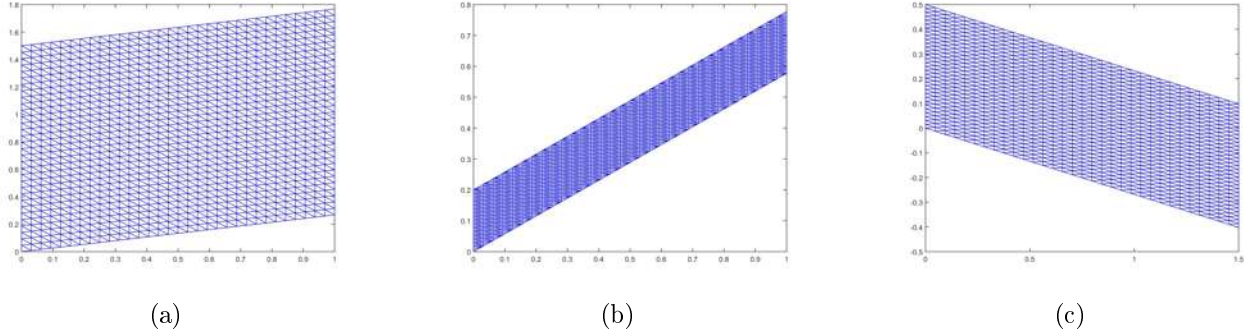


Figura 3.7: Diferentes ejemplos de mallas creadas con el código de malla automática

Se puede observar que estas mallas son buenas, ya que tienen una estructura suficientemente regular, o en otras palabras, cuentan con cierta simetría; gracias a esto puede considerarse que el uso de la pseudoinversa de M para resolver el sistema $M\Gamma = L$ generará resultados aceptables.

3.8. Linealización del problema

Como ya vimos el problema de convección natural con doble difusividad se define con tres ecuaciones diferenciales acopladas (3.9)-(3.11); y además este es un problema no lineal. Para poder resolverlo se propone hacer una aproximación lineal para la función de flujo Ψ a la cual denotaremos como $\bar{\Psi}$. Con ella se aproximan las funciones de temperatura T y de concentración C , y se harán varias iteraciones para corregir el valor de la función de flujo y las funciones de temperatura y de concentración. El algoritmo usado para resolver el sistema de ecuaciones gobernantes de nuestro problema es el siguiente:

1. Se propone una aproximación lineal para T y C , de tal manera que sus derivadas $\frac{\partial T}{\partial x}$ y $\frac{\partial C}{\partial x}$ son constantes. Esta aproximación lineal debe cumplir las condiciones de Dirichlet en las fronteras izquierda y derecha, y las condiciones de Neumann igual a cero en las fronteras de arriba y abajo, por lo que un plano inclinado que represente estas funciones puede cumplir estas condiciones.

Con la aproximación anterior el término $Ra \frac{\partial}{\partial x}(T + NC) = A$, donde A es una constante. Entonces la ecuación (3.9) se convierte en

$$\frac{\partial^2 \bar{\Psi}}{\partial x^2} + \frac{\partial^2 \bar{\Psi}}{\partial y^2} + A = 0, \quad (3.12)$$

la cual se puede resolver para $\bar{\Psi}$.

2. Una vez que se tiene la función $\bar{\Psi}$, esta se sustituye en las dos ecuaciones gobernantes faltantes, (3.13) y (3.14), que al resolver se obtienen T y C respectivamente.

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} - \frac{\partial \bar{\Psi}}{\partial y} \frac{\partial T}{\partial x} + \frac{\partial \bar{\Psi}}{\partial x} \frac{\partial T}{\partial y} = 0, \quad (3.13)$$

$$\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} - Le \frac{\partial \bar{\Psi}}{\partial y} \frac{\partial C}{\partial x} + Le \frac{\partial \bar{\Psi}}{\partial x} \frac{\partial C}{\partial y} = 0. \quad (3.14)$$

3. Aquí comienzan las iteraciones, donde i es el número de la iteración i -ésima. Se substituyen los valores obtenidos de T y C , en la ecuación (3.12) y se resuelve obteniendo ahora $\bar{\Psi}_i$.
4. Se substituye el valor de $\bar{\Psi}_i$ en las ecuaciones (3.13) y (3.14) para obtener ahora T_i y C_i respectivamente.
5. Se calcula la norma de la diferencia de las funciones anteriormente calculadas y las funciones actuales de la iteración i . La norma se calcula entonces como

$$Norma = \|\bar{\Psi}_i - \bar{\Psi}\| + \|T_i - T\| + \|C_i - C\|.$$

6. Se actualizan los valores de las funciones como

$$\bar{\Psi} = \bar{\Psi}_i, \quad T = T_i, \quad C = C_i.$$

7. Si la norma cumple la condición

$$Norma \leq 1 \times 10^{-2} + 1 \times 10^{-4} \|\bar{\Psi}\|, \quad (3.15)$$

entonces se actualizan los valores como en el paso 6 y termina el proceso de iteración, y por ende el algoritmo, con lo cual ya se han obtenido las soluciones. Si no se cumple la condición entonces se repiten los pasos 3 al 7 hasta que se cumpla la condición o hasta que se alcance el número máximo de iteraciones permitidas.

Es importante mencionar que la condición que debe cumplir la norma ,ecuación (3.15), es la tolerancia que queremos tener para las soluciones de las tres funciones (Ψ , T y C); la cual es una propuesta hecha para este trabajo.

Este algoritmo se ilustra de manera resumida en la figura 3.8.

Con este algoritmo se han desacoplado las ecuaciones diferenciales, de tal manera que tenemos tres ecuaciones, una para cada incógnita (Ψ , T , C). Entonces con la ecuación (2.3) podemos obtener el discriminante I para cada una de ellas. Para la primera ecuación gobernante (3.9), considerando que con ella se resuelve para Ψ , tenemos

$$A = 1, \quad B = 0, \quad C = 0,$$

por lo que su discriminante es

$$I = 0^2 - 4(1)(1) = -4 < 0,$$

por lo tanto esta primera ecuación es elíptica. Para la segunda ecuación gobernante (3.10), considerando que en ella se resuelve para T , tenemos los coeficientes

$$A = 1, \quad B = 0, \quad C = 1,$$

con lo cual obtenemos el discriminante

$$I = 0^2 - 4(1)(1) = -4 < 0,$$

por lo que esta ecuación también es elíptica. Por último tenemos la tercera ecuación gobernante (3.11), con la cual se resuelve para C , teniendo entonces los coeficientes

$$A = 1, \quad B = 0, \quad C = 1,$$

con lo que el discriminante queda como

$$I = 0^2 - 4(1)(1) = -4 < 0,$$

por lo que esta ecuación también es elíptica.

Entonces tenemos que al considerar el desacoplamiento de las ecuaciones diferenciales, entonces las tres ecuaciones son elípticas.

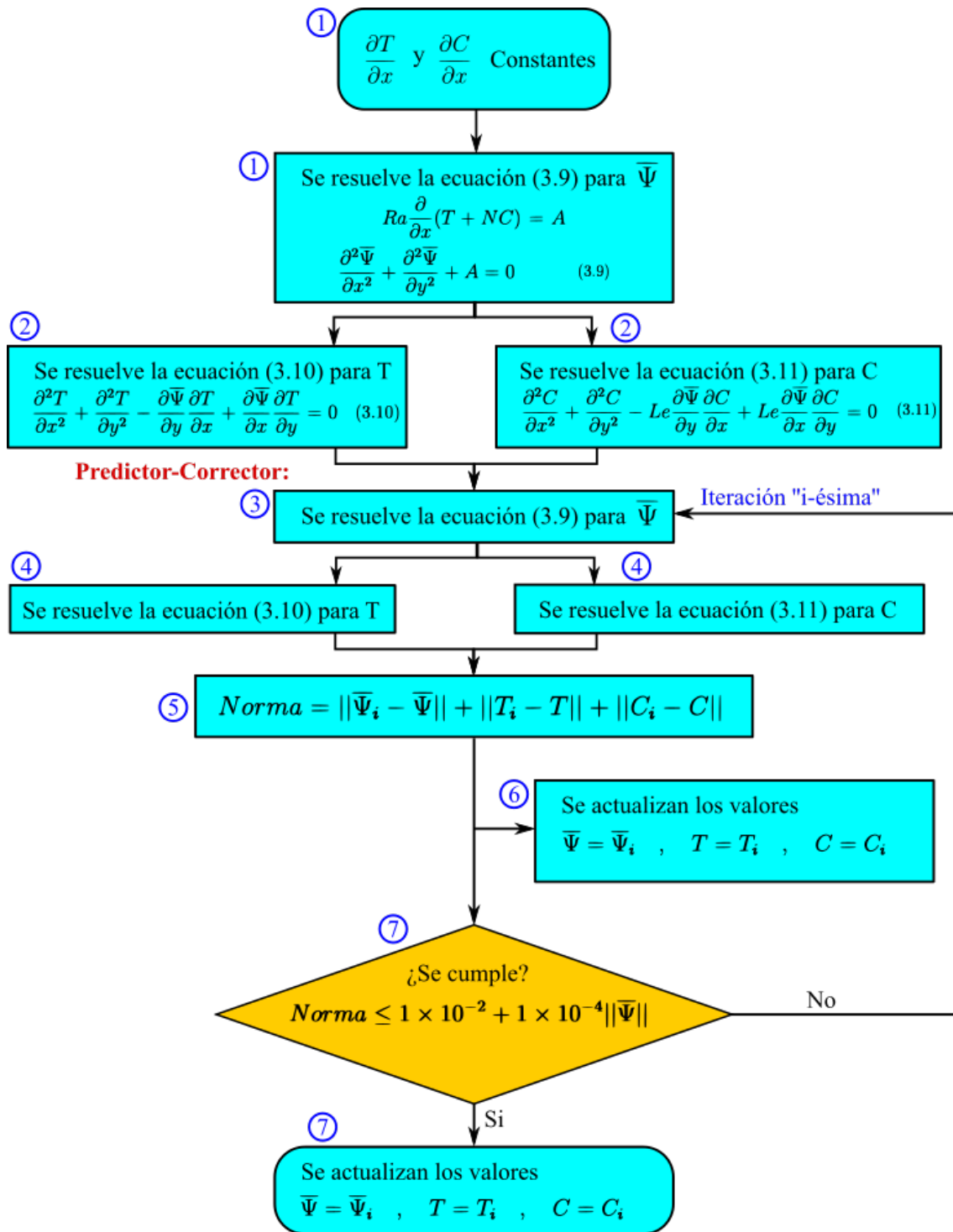


Figura 3.8: Esquema del algoritmo de solución

Capítulo 4

RESULTADOS

En varios de los ejemplos siguientes se tomaron como referencia ejemplos propuestos en otros artículos donde se usan esquemas diferentes de GFDM para resolver el problema que estamos tratando. Debido a la dificultad de que no se cuenta con resultados numéricos (como en tablas) dentro de estos artículos, entonces se comparan de manera gráfica dichos resultados con los obtenidos en este trabajo.

4.1. Ejemplos propuestos

En esta sección veremos algunas de las gráficas de resultados, de varios ejemplos con diferentes parámetros, los cuales fueron propuestos en este trabajo.

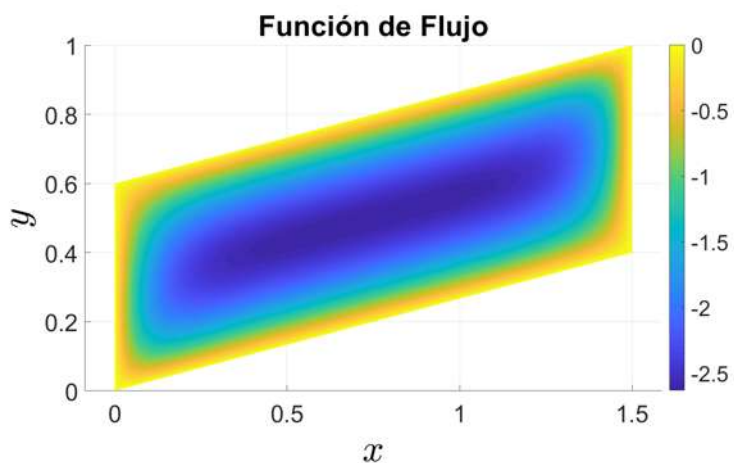
Ejemplo 1

Para esta aproximación se usó una malla triangular creada con la herramienta *pdetool* del software MatLAB, esta malla se puede ver en la figura (3.4). Para resolver se usó el algoritmo propuesto anteriormente, el cual se ilustra en la figura (3.8).

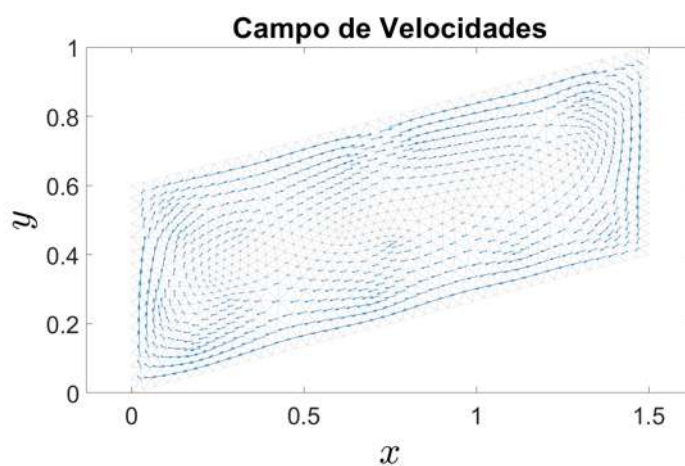
Los parámetros usados en esta aproximación son:

$$Ra = 100 \quad , \quad Le = 0.80 \quad , \quad N = 2.0 \quad .$$

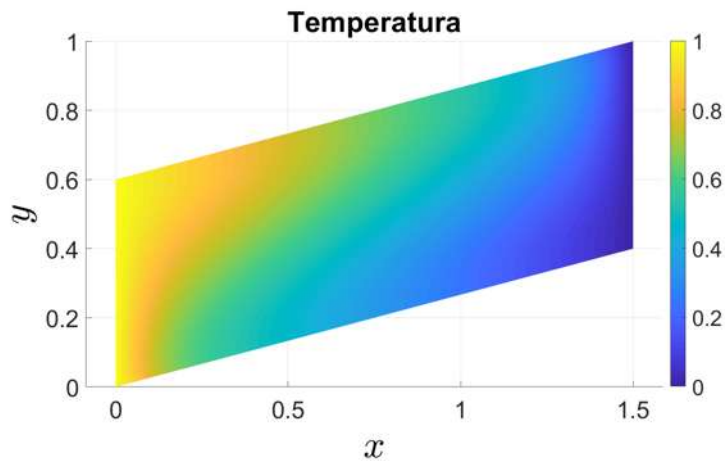
Los resultados de esta primera aproximación pueden verse en las figuras siguientes. En la figura siguiente se muestran los resultados obtenidos de la función de flujo.

Figura 4.1: Solución de la función de flujo Ψ

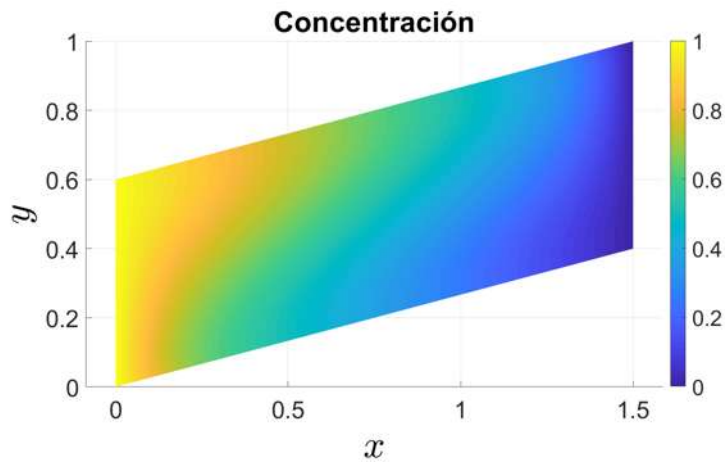
El campo de velocidades obtenido a partir de la función de flujo se muestra en la figura siguiente.

Figura 4.2: Campo vectorial de velocidades de flujo obtenidas a partir de la función de flujo Ψ

En la siguiente figura se muestran los datos de la función de temperatura.

Figura 4.3: Solución de la función de Temperatura T

Por último tenemos los resultados obtenidos para la función de concentración.

Figura 4.4: Solución de la función de Concentración C

Ejemplo 2

Para la segunda aproximación se hace uso también del algoritmo descrito en la figura (3.8), pero en lugar de usar una malla triangular irregular se utiliza una malla rectangular donde se ha triangulado usando el algoritmo de Delaunay. Se puede ver esta malla en la figura (3.6).

Los parámetros utilizados son los mismos que en la primera aproximación, los cuales son $Ra = 100$, $Le = 0.80$, $N = 2.0$. Los resultados obtenidos se muestran en las siguientes figuras.

En la figura siguiente se muestran los resultados obtenidos de la función de flujo.

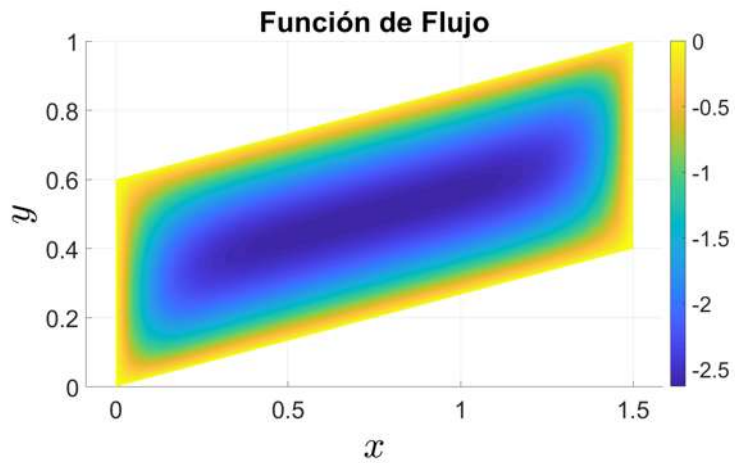


Figura 4.5: Solución de la función de flujo Ψ

El campo de velocidades obtenido a partir de la función de flujo se muestra en la figura siguiente.

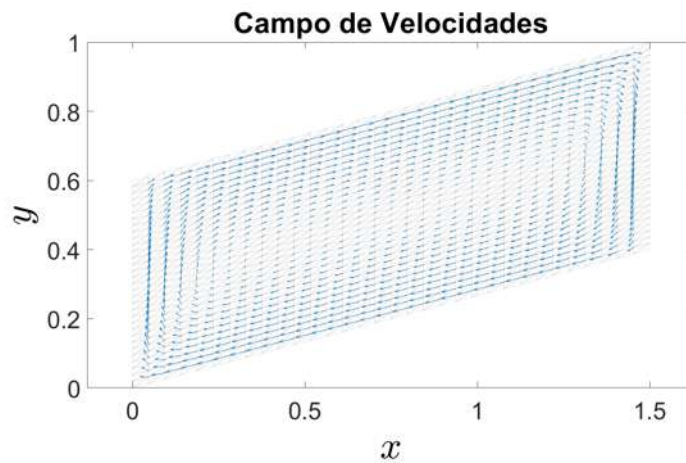
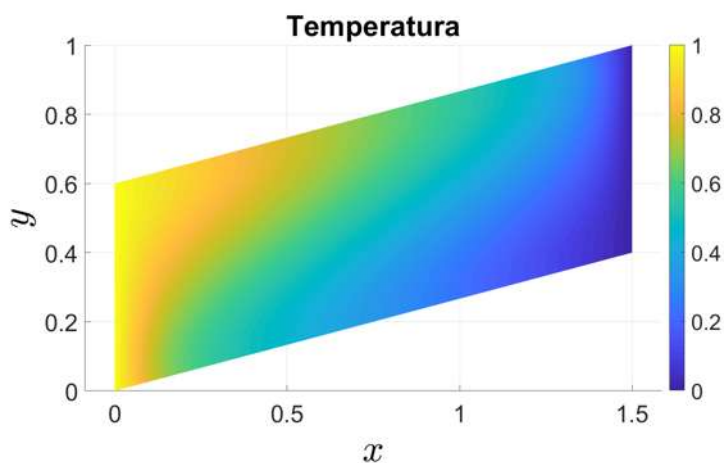
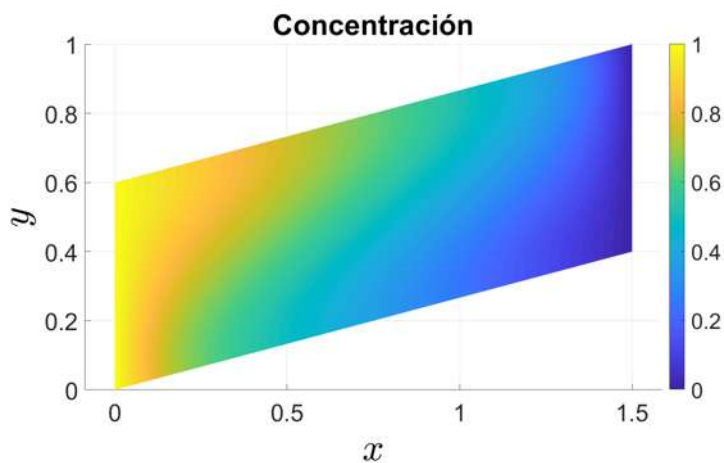


Figura 4.6: Campo vectorial de velocidades de flujo obtenidas a partir de la función de flujo Ψ

En la siguiente figura se muestran los datos de la función de temperatura.

Figura 4.7: Solución de la función de Temperatura T

Por último tenemos los resultados obtenidos para la función de concentración.

Figura 4.8: Solución de la función de Concentración C

Ejemplo 3

En este ejemplo se considera que no hay transporte de concentración, por lo cual $C = 0$. Los parámetros empleados son los siguientes: $Ra = 100$, $N = 2$, $Le = 0.8$, $H = 0.3$, $H/L = 0.25$, $\theta = 30^\circ$. Con los parámetros geométricos se obtiene la siguiente malla:

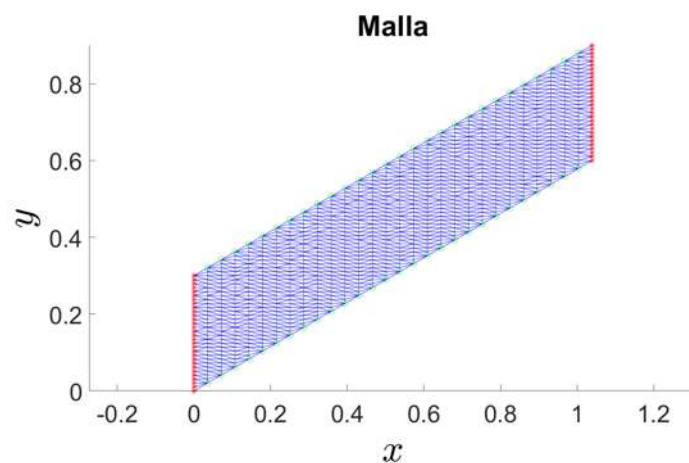


Figura 4.9: Malla generada automáticamente usando los parámetros geométricos $H = 0.3$, $H/L = 0.25$, $\theta = 30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (o) representan la frontera con condiciones de Neumann.

En la figura siguiente se muestran los resultados obtenidos de la función de flujo.

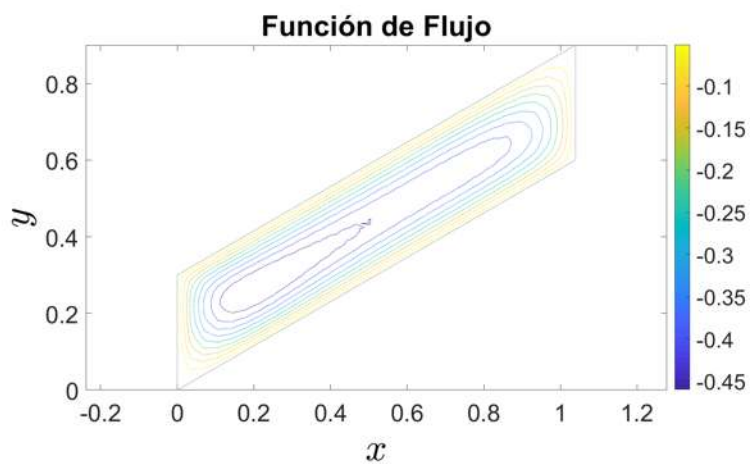


Figura 4.10: Función de flujo.

El campo de velocidades obtenido a partir de la función de flujo se muestra en la figura siguiente.

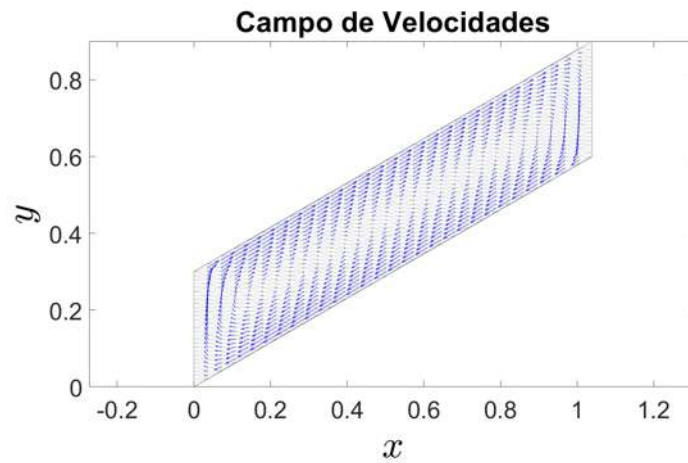


Figura 4.11: Campo de velocidades obtenido de la función de flujo.

En la siguiente figura se muestran los datos de la función de temperatura.

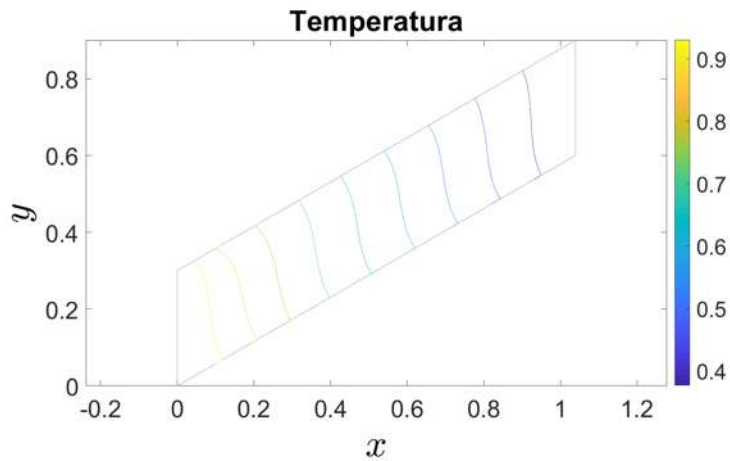


Figura 4.12: Función de temperatura.

Ejemplo 4

Este ejemplo es muy parecido al anterior, el único parámetro que se modifica respecto al ejemplo 3 es el ángulo de inclinación del paralelogramo. En este caso este ángulo es $\theta = -30^\circ$. Con estos datos geométricos se obtiene la siguiente malla.

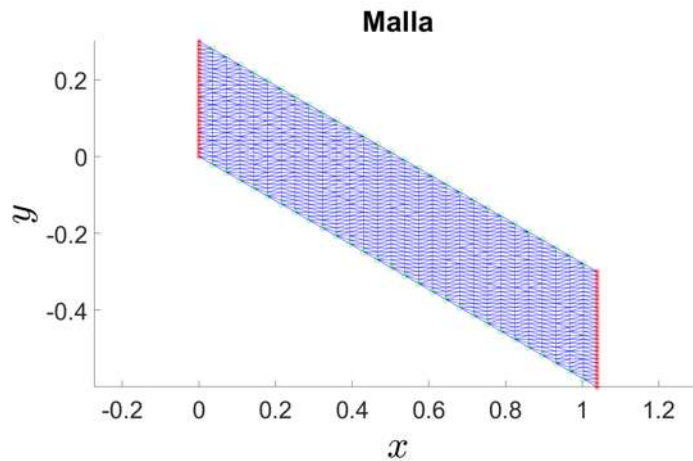


Figura 4.13: Malla generada automáticamente usando los parámetros geométricos $H = 0.3$, $H/L = 0.25$, $\theta = -30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (o) representan la frontera con condiciones de Neumann.

En la figura siguiente se muestran los resultados obtenidos de la función de flujo.

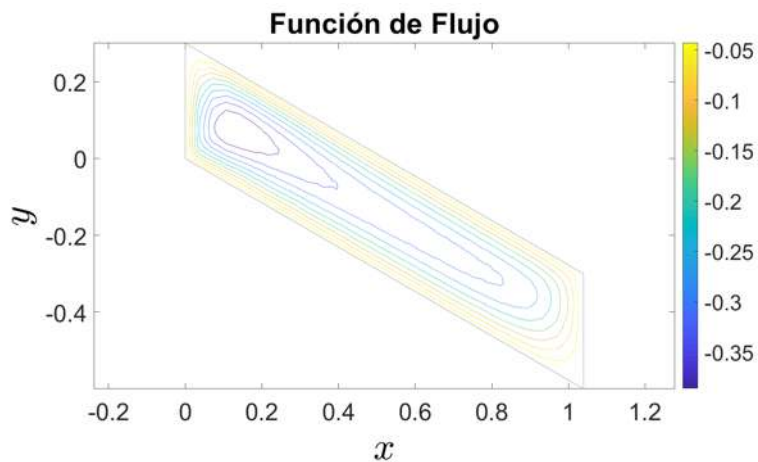


Figura 4.14: Función de flujo.

El campo de velocidades obtenido a partir de la función de flujo se muestra en la figura siguiente.

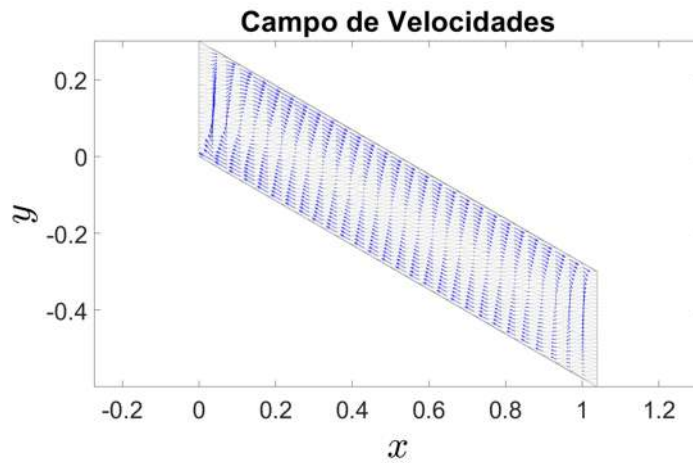


Figura 4.15: Campo de velocidades obtenido de la función de flujo.

En la siguiente figura se muestran los datos de la función de temperatura.

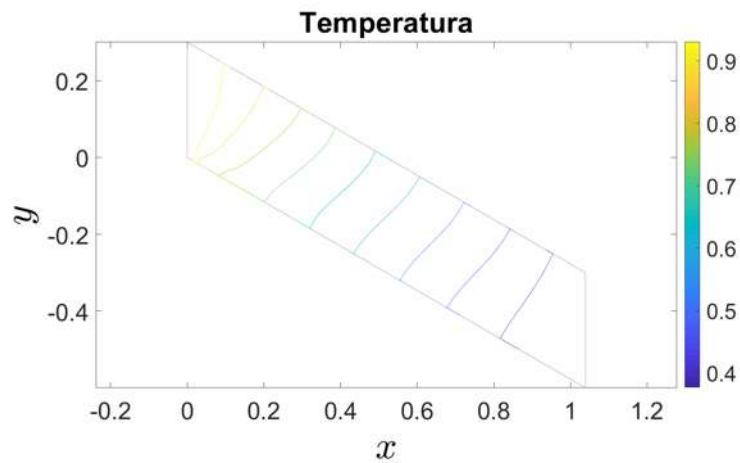


Figura 4.16: Función de temperatura.

Ejemplo 5

En este ejemplo se considera ahora que existe transporte de concentración, por lo cual $C \neq 0$. Los parámetros empleados son los siguientes: $Ra = 100$, $N = 2$, $Le = 0.8$, $H = 0.3$, $H/L = 0.5$, $\theta = 30^\circ$. De los parámetros geométricos se obtiene la siguiente malla:

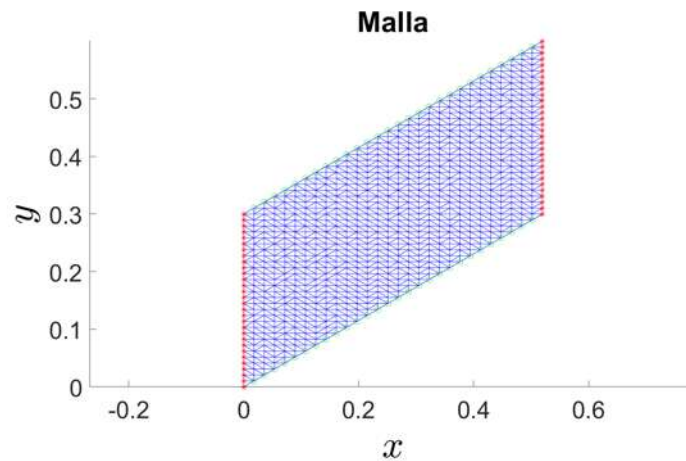


Figura 4.17: Malla generada automáticamente usando los parámetros geométricos $H = 0.3$, $H/L = 0.5$, $\theta = 30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (o) representan la frontera con condiciones de Neumann.

En la figura siguiente se muestran los resultados obtenidos de la función de flujo.

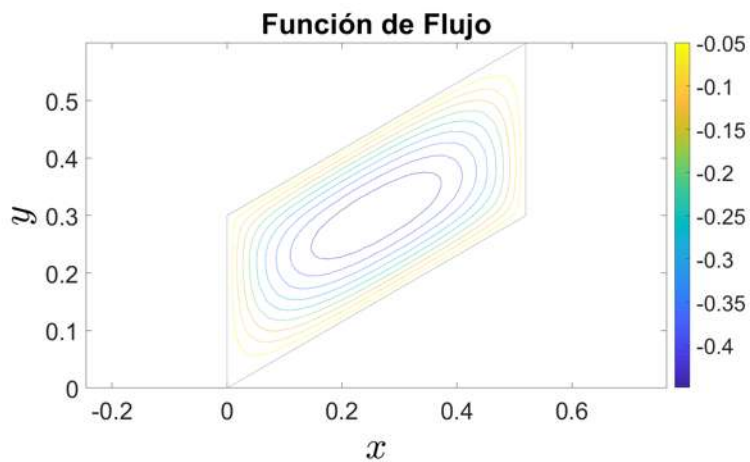


Figura 4.18: Función de flujo.

El campo de velocidades obtenido a partir de la función de flujo se muestra en la figura siguiente.

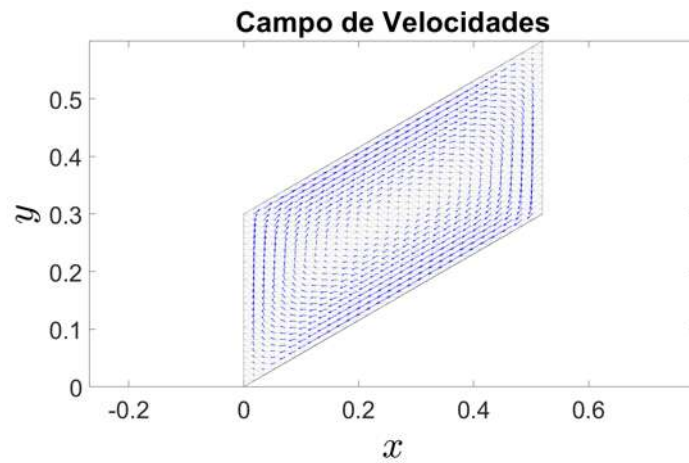


Figura 4.19: Campo de velocidades obtenido de la función de flujo.

En la siguiente figura se muestran los datos de la función de temperatura.

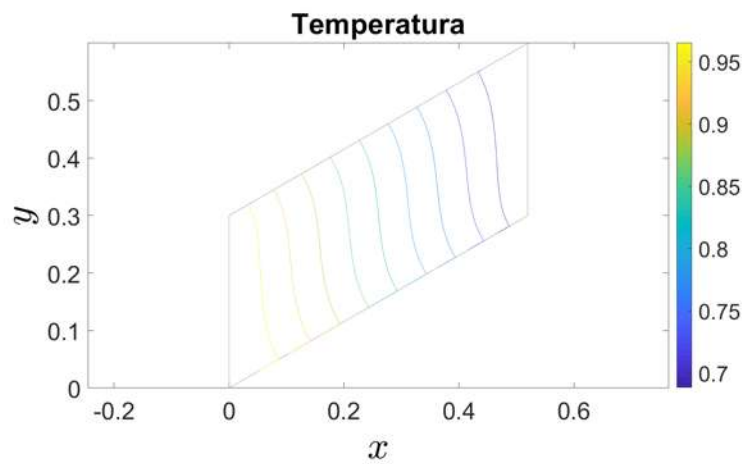


Figura 4.20: Función de temperatura.

Y por último, en la siguiente figura se muestran los datos de la función de concentración.

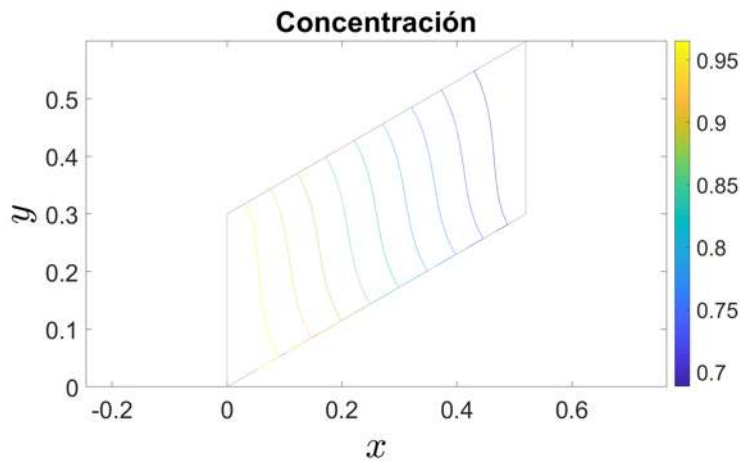


Figura 4.21: Función de temperatura.

Ejemplo 6

Para este ejemplo se usaron los parámetros siguientes: $Le = 0.8$, $N = 0$, $Ra = 100$, $H = 0.46$, $H/L = 1$, $\theta = 30^\circ$. Con los parámetros geométricos se obtiene la siguiente malla de puntos.

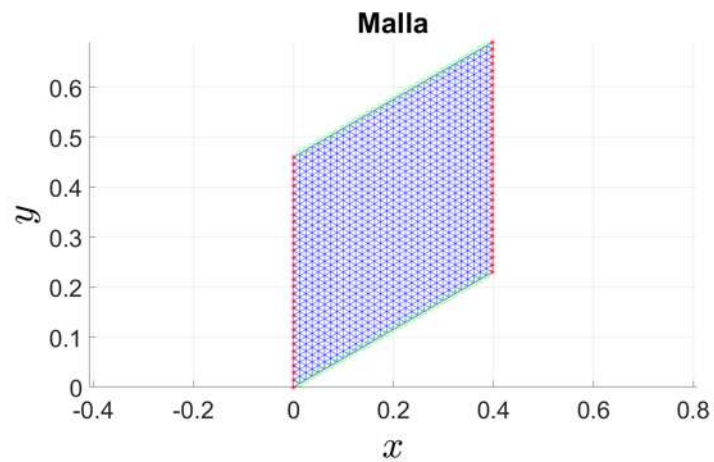


Figura 4.22: Malla generada automáticamente usando los parámetros geométricos $H = 0.46$, $H/L = 1$, $\theta = 30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (o) representan la frontera con condiciones de Neumann.

Los resultados obtenidos para la función de flujo se muestran en la siguiente figura.

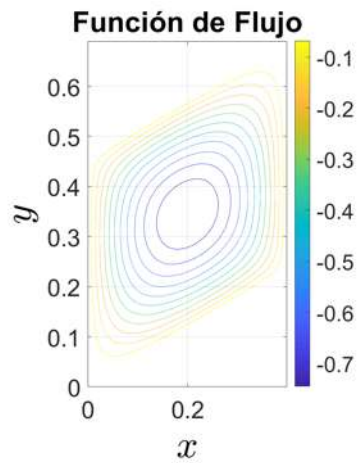


Figura 4.23: Función de flujo.

El campo de velocidades obtenido a partir de la función de flujo se muestra en la figura siguiente.

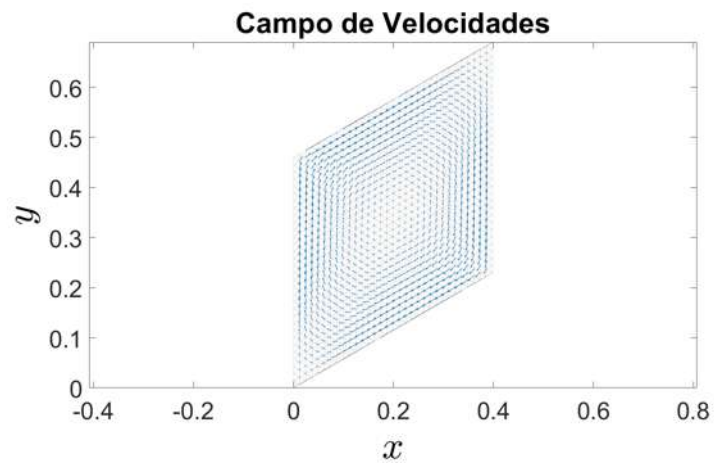


Figura 4.24: Campo de velocidades obtenido de la función de flujo.

En la siguiente figura se muestran los datos de la función de temperatura.

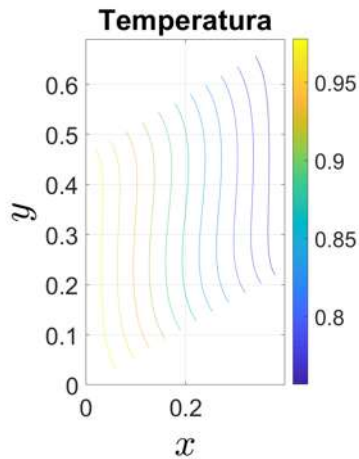


Figura 4.25: Función de temperatura.

Ejemplo 7

Para este ejemplo se usaron los parámetros siguientes: $Le = 0.8$, $N = 0$, $Ra = 100$, $H = 0.31$, $H/L = 0.25$, $\theta = 30^\circ$. Con los parámetros geométricos se obtiene la siguiente malla de puntos.

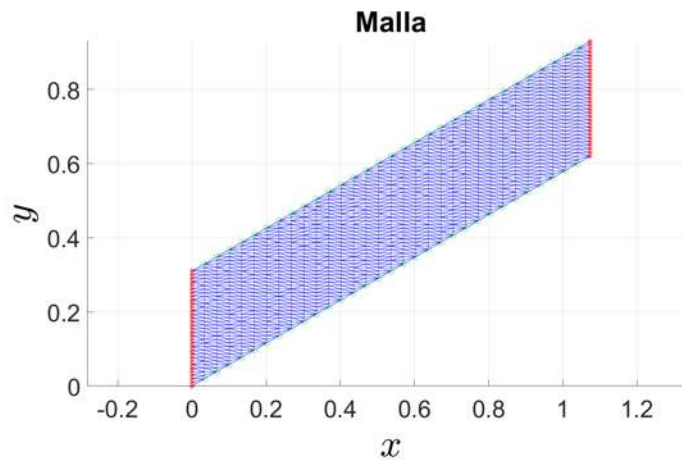


Figura 4.26: Malla generada automáticamente usando los parámetros geométricos $H = 0.31$, $H/L = 0.25$, $\theta = 30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (o) representan la frontera con condiciones de Neuman.

Los resultados obtenidos para la función de flujo se muestran en la siguiente figura.

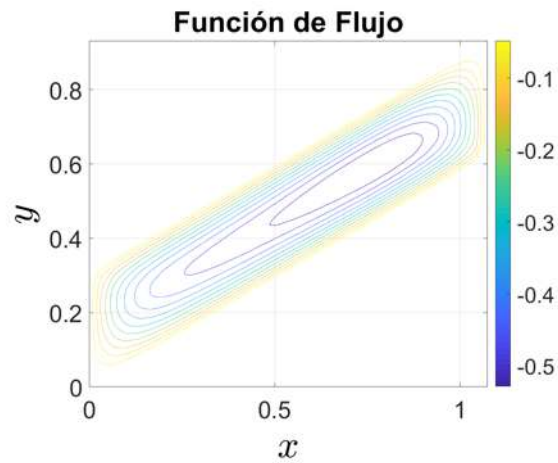


Figura 4.27: Función de flujo.

El campo de velocidades obtenido a partir de la función de flujo se muestra en la figura siguiente.

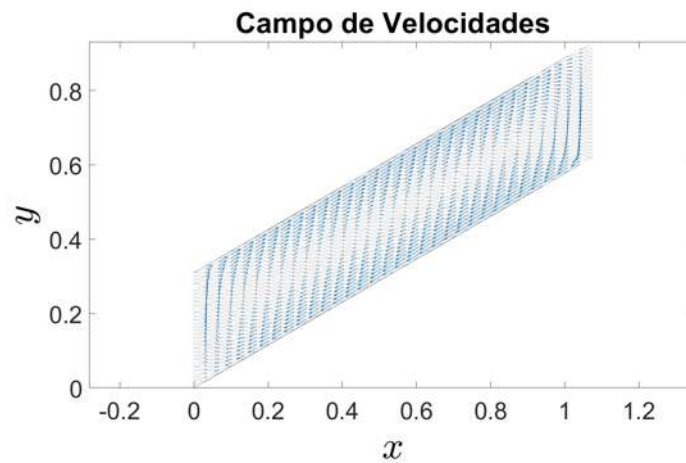


Figura 4.28: Campo de velocidades obtenido de la función de flujo.

En la siguiente figura se muestran los datos de la función de temperatura.

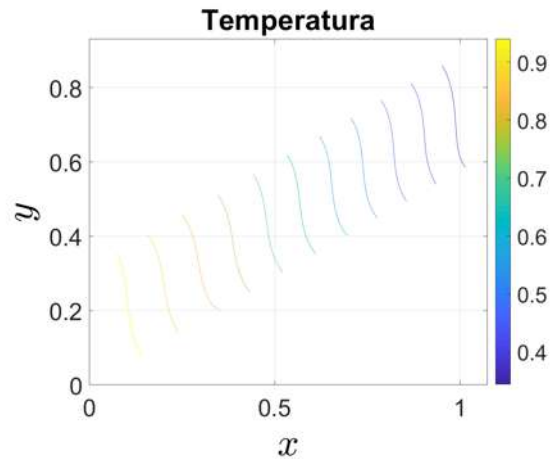


Figura 4.29: Función de temperatura.

Ejemplo 8

Para este ejemplo se usaron los parámetros siguientes: $Le = 0.8$, $N = 0$, $Ra = 100$, $H = 0.31$, $H/L = 0.25$, $\theta = -30^\circ$. Con los parámetros geométricos se obtiene la siguiente malla de puntos.

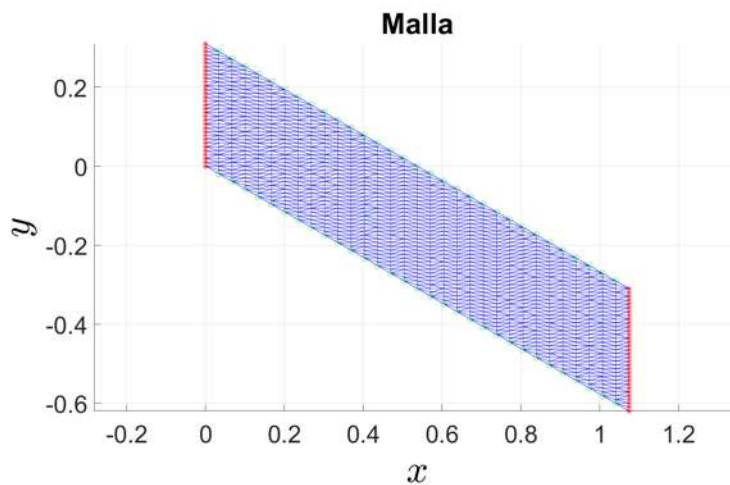


Figura 4.30: Malla generada automáticamente usando los parámetros geométricos $H = 0.31$, $H/L = 0.25$, $\theta = -30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (o) representan la frontera con condiciones de Neumann.

Los resultados obtenidos para la función de flujo se muestran en la figura siguiente.

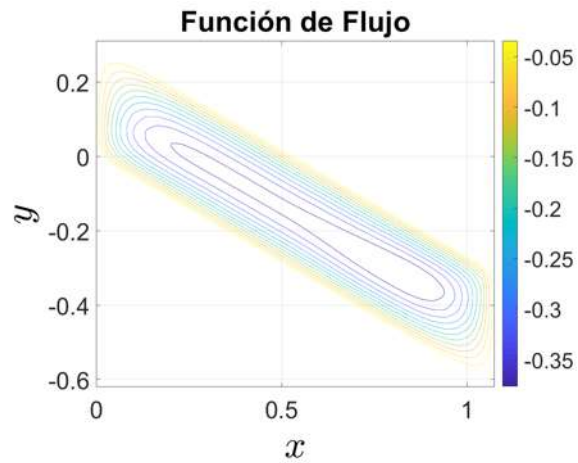


Figura 4.31: Función de flujo.

El campo de velocidades obtenido a partir de la función de flujo se muestra en la figura siguiente.

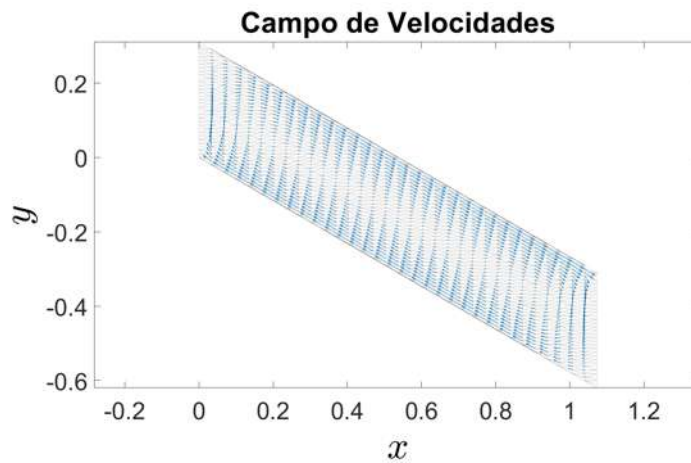


Figura 4.32: Campo de velocidades obtenido de la función de flujo.

En la siguiente figura se muestran los datos de la función de temperatura.

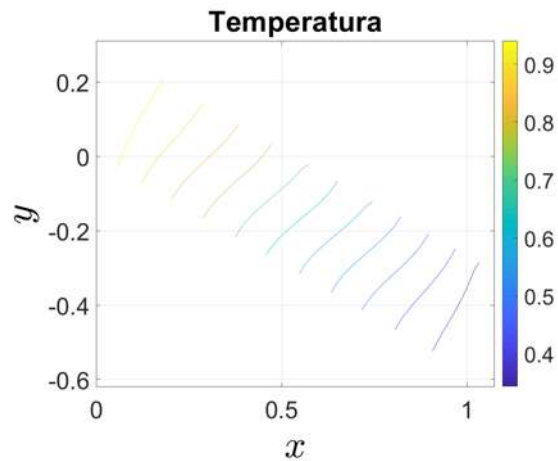


Figura 4.33: Función de temperatura.

Ejemplo 9

Para este ejemplo se usaron los parámetros siguientes: $Le = 0.8$, $N = 2$, $Ra = 100$, $H = 0.3$, $H/L = 0.5$, $\theta = 30^\circ$. Con los parámetros geométricos se obtiene la siguiente malla de puntos.

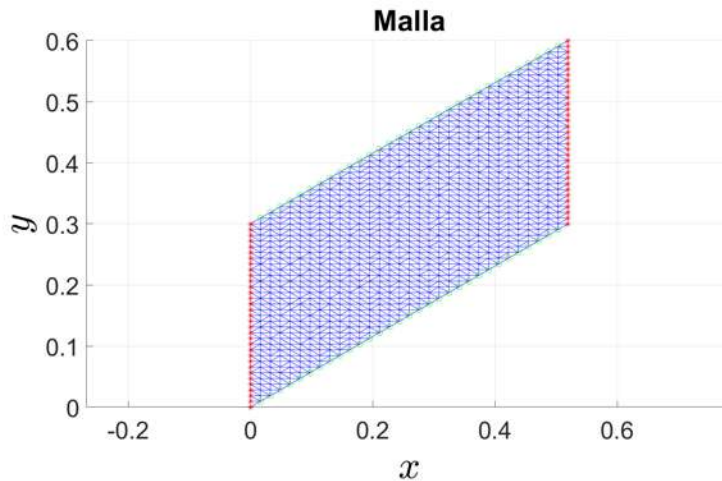


Figura 4.34: Malla generada automáticamente usando los parámetros geométricos $H = 0.3$, $H/L = 0.5$, $\theta = 30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (o) representan la frontera con condiciones de Neumann.

Los resultados obtenidos para la función de flujo se muestran en la figura siguiente.

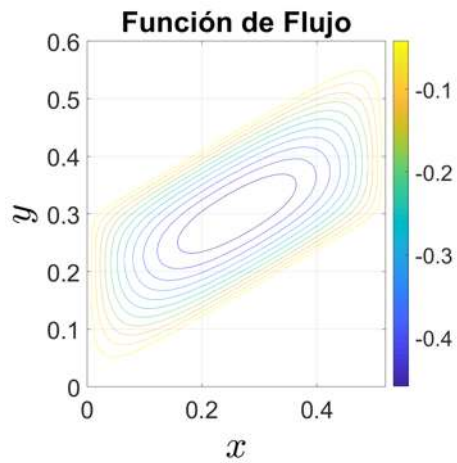


Figura 4.35: Función de flujo.

El campo de velocidades obtenido a partir de la función de flujo se muestra en la figura siguiente.

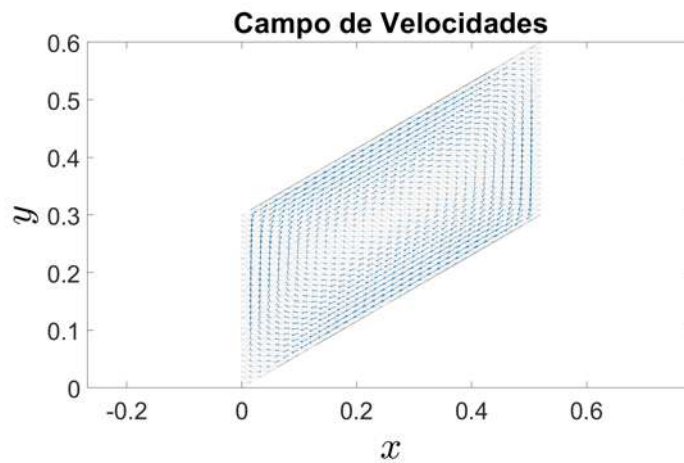


Figura 4.36: Campo de velocidades obtenido de la función de flujo.

En la siguiente figura se muestran los datos de la función de temperatura.

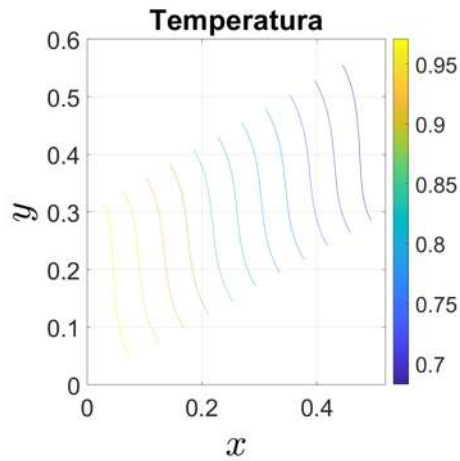


Figura 4.37: Función de temperatura.

Por último tenemos los resultados obtenidos para la función de concentración.

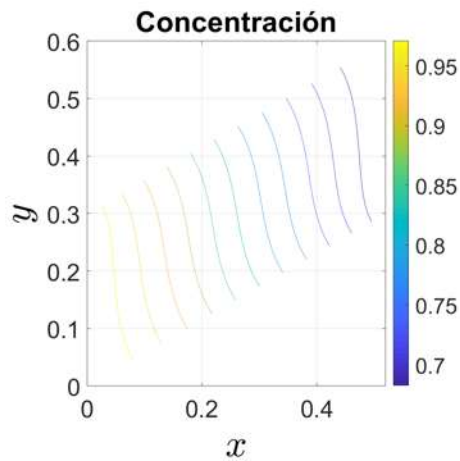


Figura 4.38: Función de concentración.

Ejemplo 10

Para este ejemplo se usaron los parámetros siguientes: $Le = 0.8$, $N = 5$, $Ra = 100$, $H = 0.3$, $H/L = 0.5$, $\theta = 30^\circ$. Con los parámetros geométricos se obtiene la siguiente malla de puntos.

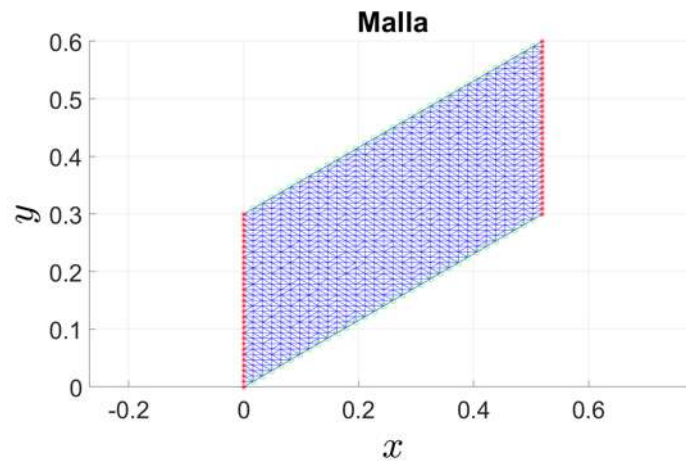


Figura 4.39: Malla generada automáticamente usando los parámetros geométricos $H = 0.3$, $H/L = 0.5$, $\theta = 30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (o) representan la frontera con condiciones de Neumann.

Los resultados obtenidos para la función de flujo se muestran en la figura siguiente.

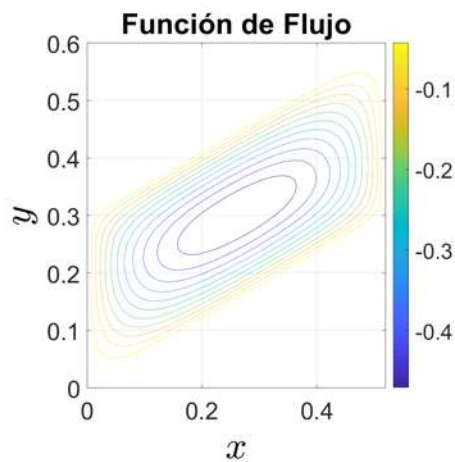


Figura 4.40: Función de flujo.

El campo de velocidades obtenido a partir de la función de flujo se muestra en la figura siguiente.

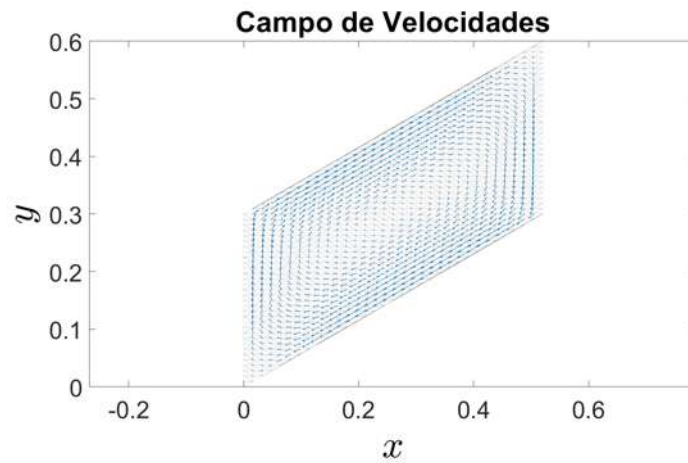


Figura 4.41: Campo de velocidades obtenido de la función de flujo.

En la siguiente figura se muestran los datos de la función de temperatura.

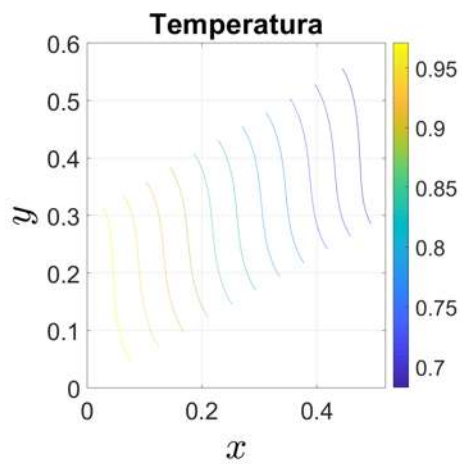


Figura 4.42: Función de temperatura.

Por último tenemos los resultados obtenidos para la función de concentración.

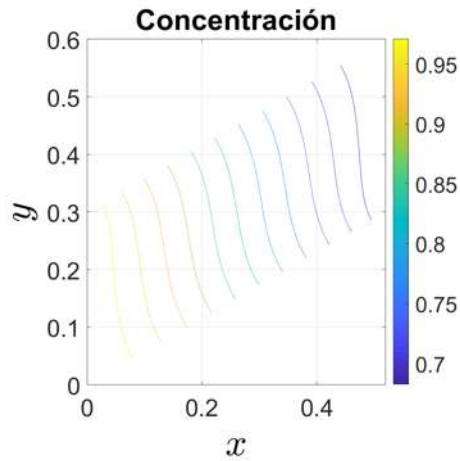


Figura 4.43: Función de concentración.

Ejemplo 11

Para este ejemplo se usaron los parámetros siguientes: $Le = 0.8$, $N = 5$, $Ra = 100$, $H = 0.6$, $H/L = 0.5$, $\theta = 30^\circ$. Con los parámetros geométricos se obtiene la siguiente malla de puntos. Una particularidad de este ejemplo es que el flujo de temperatura y de concentración están invertidos, en otras palabras, las condiciones en la frontera izquierda son $T = 0$ y $C = 0$, mientras que en la frontera derecha son $T = 1$ y $C = 1$.

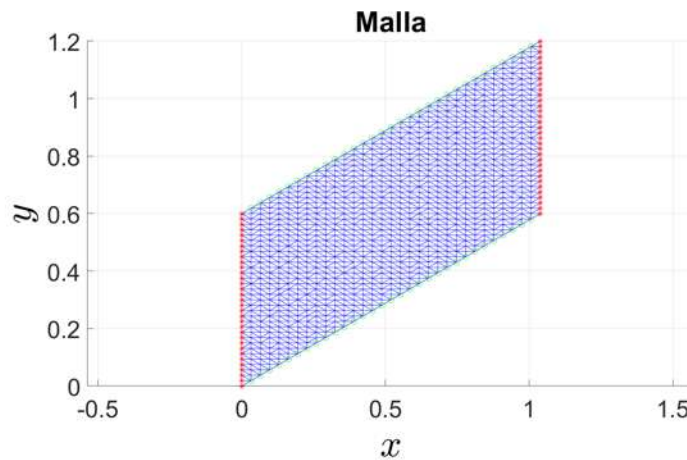


Figura 4.44: Malla generada automáticamente usando los parámetros geométricos $H = 0.3$, $H/L = 0.5$, $\theta = 30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (o) representan la frontera con condiciones de Neumann.

Los resultados obtenidos para la función de flujo se muestran en la figura siguiente.

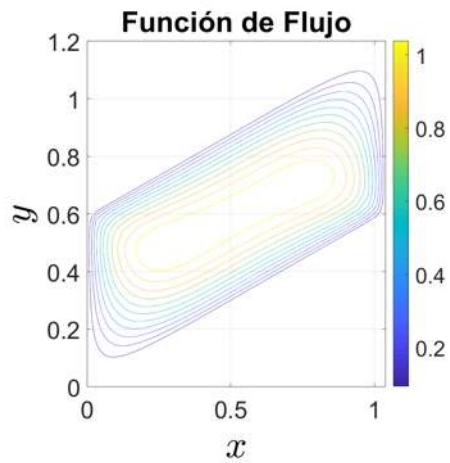


Figura 4.45: Función de flujo.

El campo de velocidades obtenido a partir de la función de flujo se muestra en la figura siguiente.

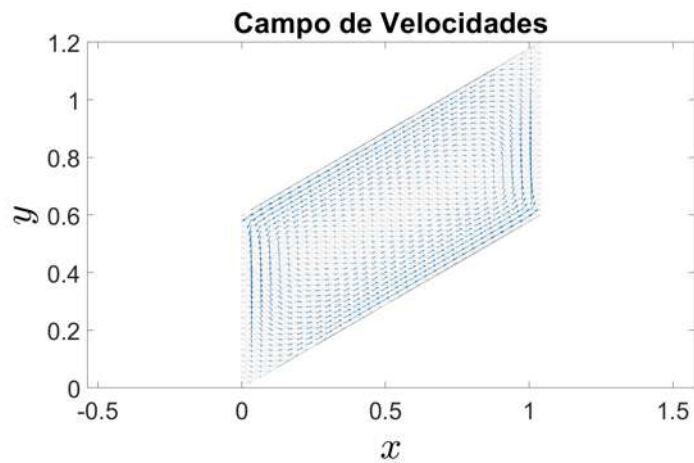


Figura 4.46: Campo de velocidades obtenido de la función de flujo.

En la siguiente figura se muestran los datos de la función de temperatura.

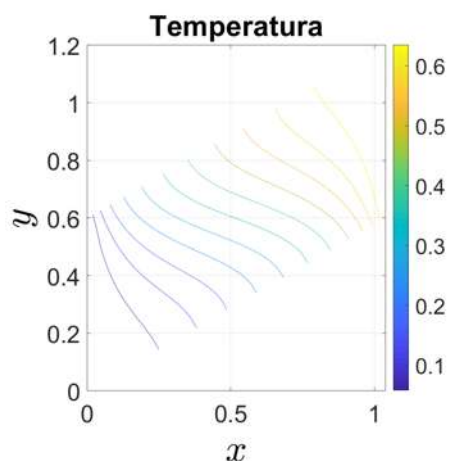


Figura 4.47: Función de temperatura.

Por último tenemos los resultados obtenidos para la función de concentración.

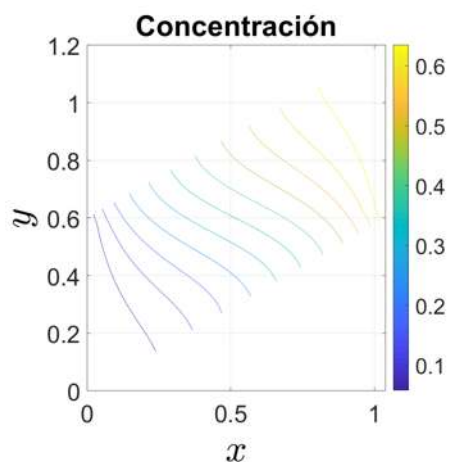


Figura 4.48: Función de concentración.

4.2. Ejemplos con mallas no estructuradas

En estos ejemplos se crearon las mallas a partir de los parámetros propuestos en los ejemplos anteriores, pero se utilizó la herramienta *pdetool* de MatLAB para crearlas, la cual llama a las funciones *initmesh* y *jigglemesh*, las cuales inicializan y suavizan la malla respectivamente. El valor de H que se consideró para cada malla es aquel que daba mejores resultados.

4.2.1. Ejemplos tomados de Fan

Para estos ejemplos se tomaron como referencia aquellos propuestos por Fan (Fan et al., 2013), con el objetivo de comparar los resultados obtenidos con la metodología propuesta por Fan y la metodología propuesta en este trabajo. El proceso de iteraciones es el mismo que para los

dos ejemplos pasados, primero se resuelve la primera ecuación gobernante para la función de flujo, después se usa el resultado para resolver las siguientes dos ecuaciones gobernantes para la temperatura y para la concentración, y se repite el proceso hasta minimizar la norma de la resta entre los resultados de la iteración actual y la iteración anterior.

Ejemplo 12

En esta primera comparación se considera que no hay transporte de concentración, por lo cual $C = 0$. Los parámetros empleados son los siguientes: $Ra = 100$, $N = 2$, $Le = 0.8$, $H = 0.7$, $H/L = 0.25$, $\theta = 30^\circ$. Con los parámetros geométricos se obtiene la siguiente malla:

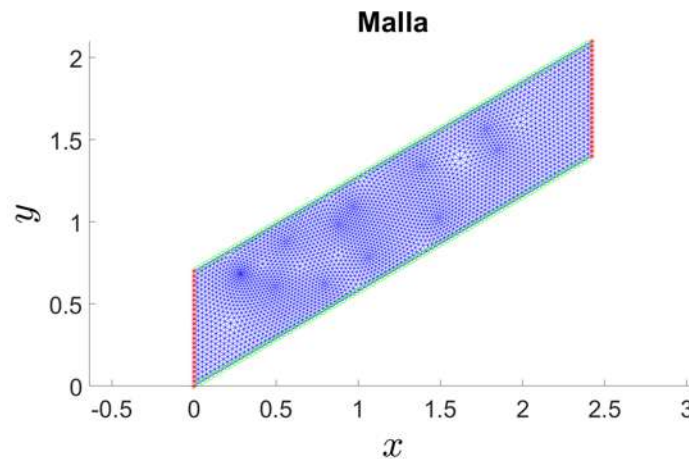


Figura 4.49: Malla generada usando los parámetros geométricos $H = 0.7$, $H/L = 0.25$, $\theta = 30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (o) representan la frontera con condiciones de Neumann.

En la figura siguiente se muestra del lado izquierdo los resultados obtenidos de la función de flujo, mientras que del lado derecho se muestran los resultados para la función de flujo obtenidos por Fan.

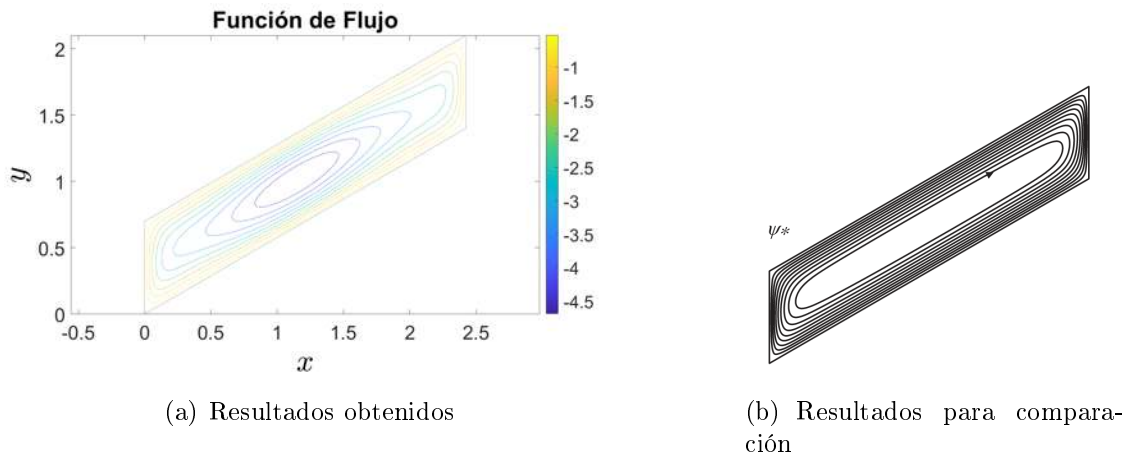


Figura 4.50: Comparación de la función de flujo obtenida con la metodología propuesta y la obtenida por Fan.

El campo de velocidades obtenido a partir de la función de flujo se muestra en la figura siguiente.

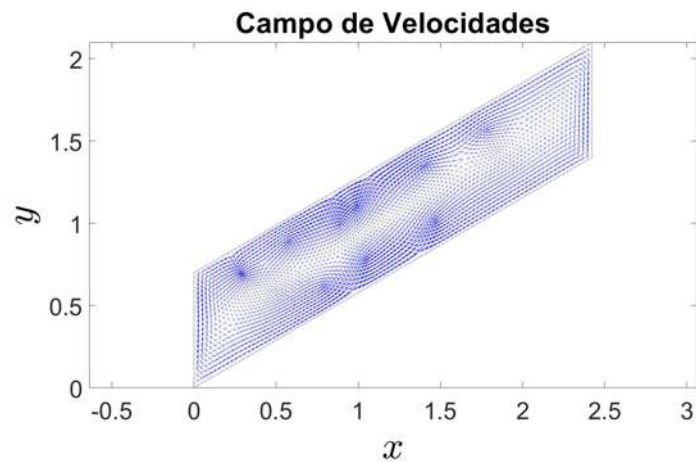


Figura 4.51: Campo de velocidades obtenido de la función de flujo.

En la siguiente figura se comparan los datos de la función de temperatura.

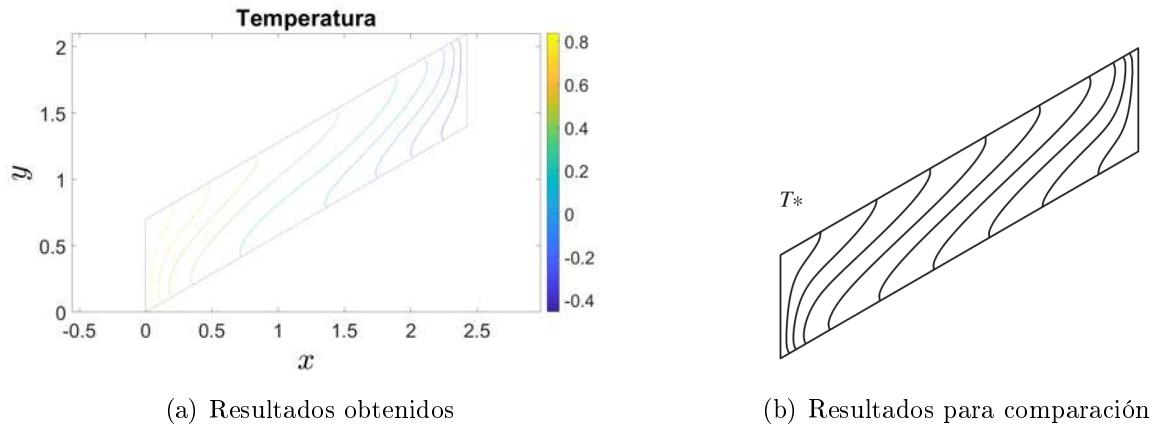


Figura 4.52: Comparación de la función de temperatura obtenida con la metodología propuesta y la obtenida por Fan.

Ejemplo 13

Este ejemplo es muy parecido al anterior, el único parámetro que se modifica respecto al ejemplo 3 es el ángulo de inclinación del paralelogramo. En este caso este ángulo es $\theta = -30^\circ$. Con estos datos geométricos se obtiene la siguiente malla;

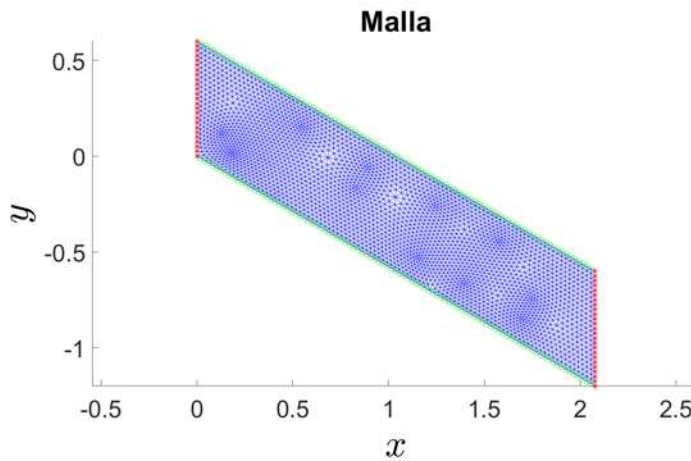


Figura 4.53: Malla generada usando los parámetros geométricos $H = 0.6$, $H/L = 0.25$, $\theta = -30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (o) representan la frontera con condiciones de Neumann.

En la figura siguiente se muestra del lado izquierdo los resultados obtenidos de la función de flujo, mientras que del lado derecho se muestran los resultados para la función de flujo obtenidos por Fan.

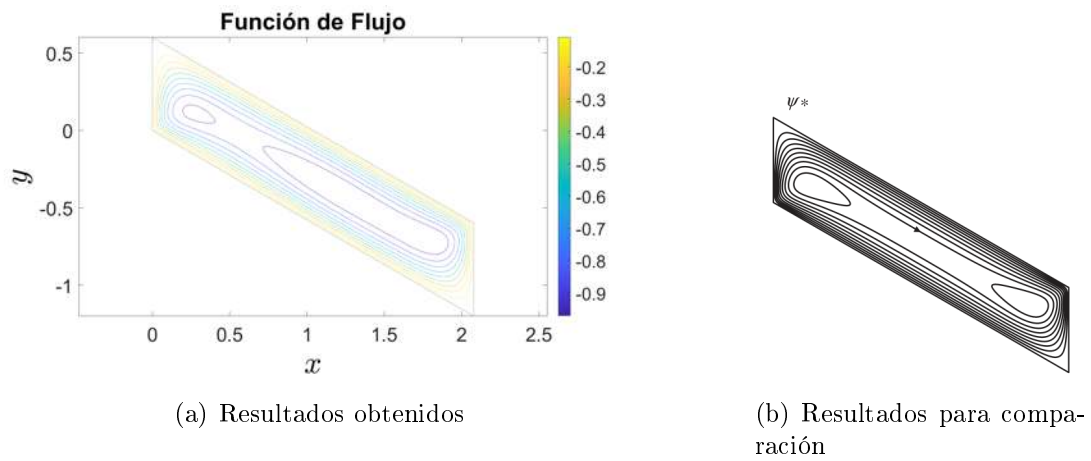


Figura 4.54: Comparación de la función de flujo obtenida con la metodología propuesta y la obtenida por Fan.

El campo de velocidades obtenido a partir de la función de flujo se muestra en la figura siguiente.

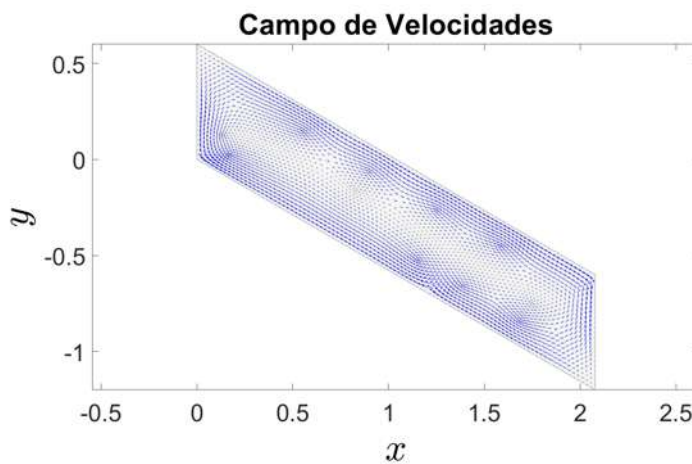


Figura 4.55: Campo de velocidades obtenido de la función de flujo.

En la siguiente figura se comparan los datos de la función de temperatura.

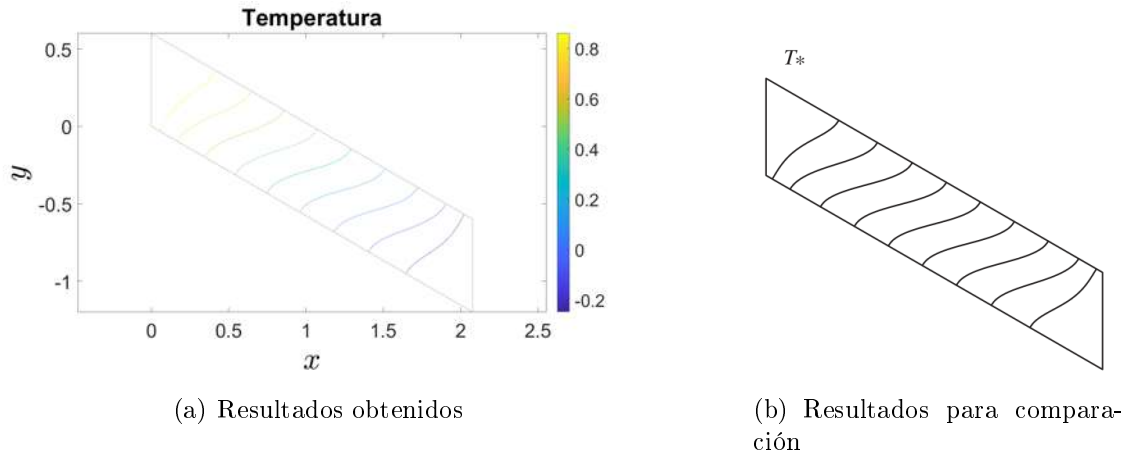


Figura 4.56: Comparación de la función de temperatura obtenida con la metodología propuesta y la obtenida por Fan.

Ejemplo 14

En este ejemplo se considera ahora que existe transporte de concentración, por lo cual $C \neq 0$. Los parámetros empleados son los siguientes: $Ra = 100$, $N = 2$, $Le = 0.8$, $H = 1$, $H/L = 0.5$, $\theta = 30^\circ$. De los parámetros geométricos se obtiene la siguiente malla:

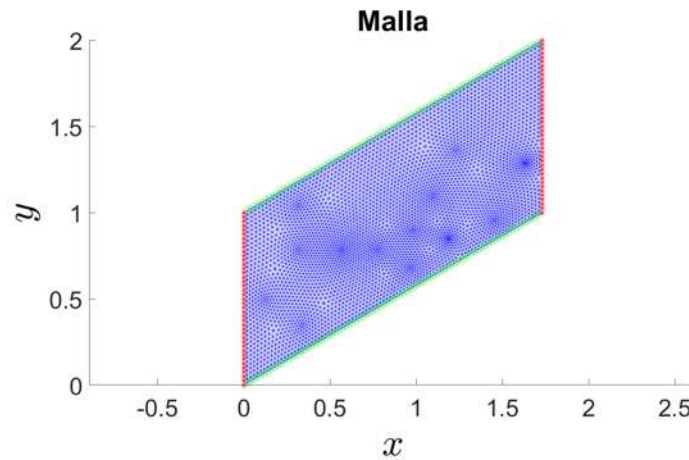


Figura 4.57: Malla generada usando los parámetros geométricos $H = 1$, $H/L = 0.5$, $\theta = 30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (o) representan la frontera con condiciones de Neumann.

En la figura siguiente se muestra del lado izquierdo los resultados obtenidos de la función de flujo, mientras que del lado derecho se muestran los resultados para la función de flujo obtenidos por Fan.

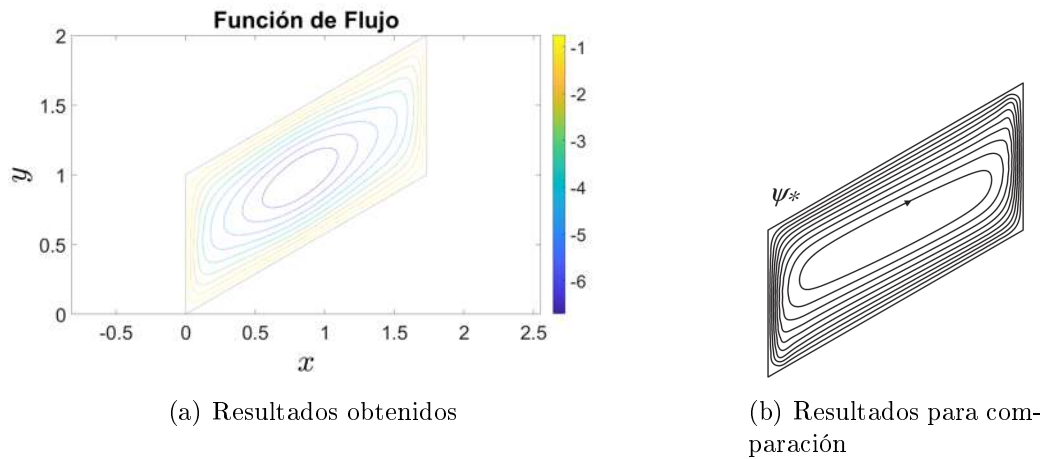


Figura 4.58: Comparación de la función de flujo obtenida con la metodología propuesta y la obtenida por Fan.

El campo de velocidades obtenido a partir de la función de flujo se muestra en la figura siguiente.

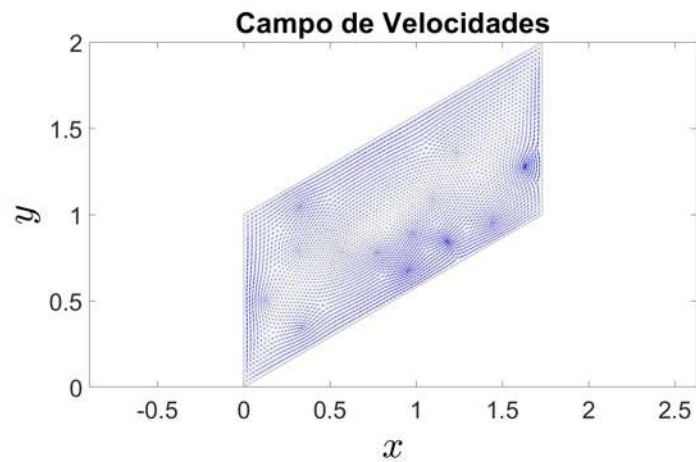


Figura 4.59: Campo de velocidades obtenido de la función de flujo.

En la siguiente figura se comparan los datos de la función de temperatura.

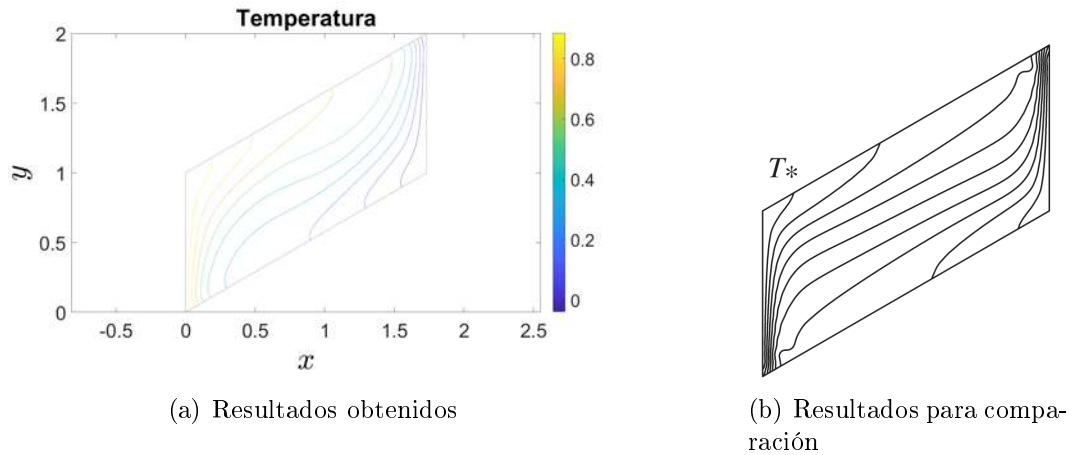


Figura 4.60: Comparación de la función de temperatura obtenida con la metodología propuesta y la obtenida por Fan.

Y por último, en la siguiente figura se comparan los datos de la función de concentración.

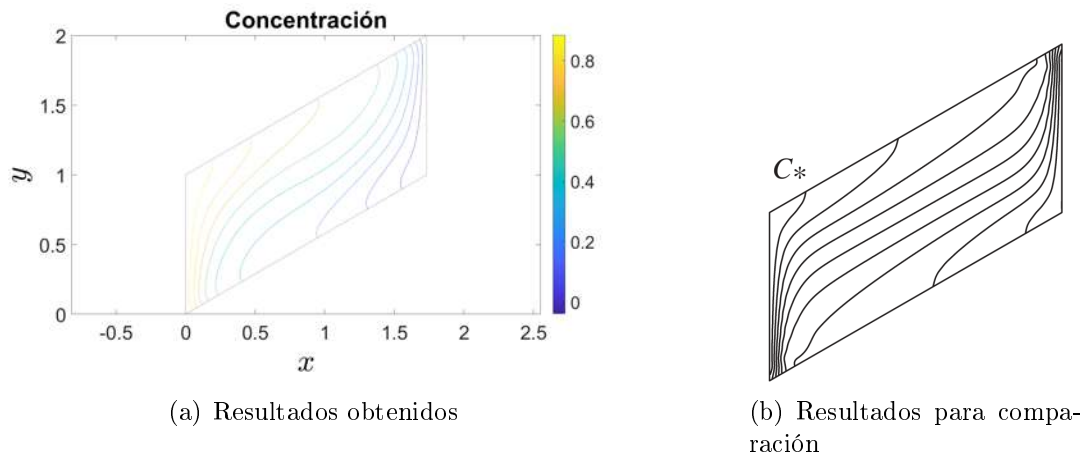


Figura 4.61: Comparación de la función de concentración obtenida con la metodología propuesta y la obtenida por Fan.

4.2.2. Ejemplos tomados de Costa

Los siguientes ejemplos son aquellos propuestos por Costa (Costa, 2004), con el objetivo de comparar los resultados obtenidos con la metodología empleada en este trabajo y aquella usada por Costa.

Ejemplo 15

Para este ejemplo se usaron los parámetros siguientes: $Le = 0.8$, $N = 0$, $Ra = 100$, $H = 0.9$, $H/L = 1$, $\theta = 30^\circ$. Con los parámetros geométricos se obtiene la siguiente malla de puntos.

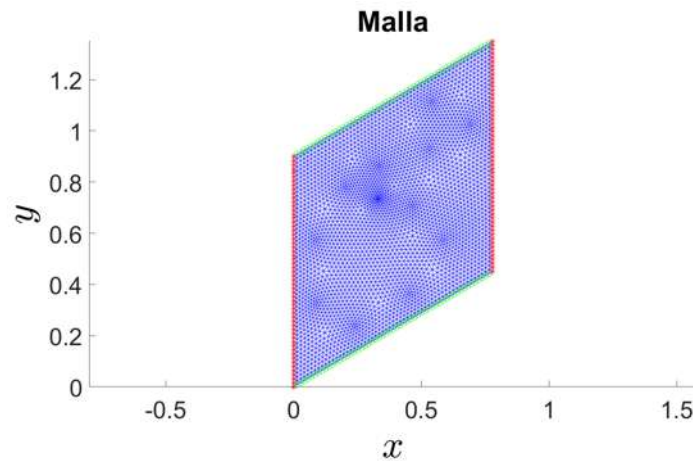


Figura 4.62: Malla generada usando los parámetros geométricos $H = 0.9$, $H/L = 1$, $\theta = 30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (o) representan la frontera con condiciones de Neuman.

Los resultados obtenidos para la función de flujo con la metodología propuesta se muestran del lado izquierdo, mientras que los obtenidos por Costa se muestran del lado derecho.

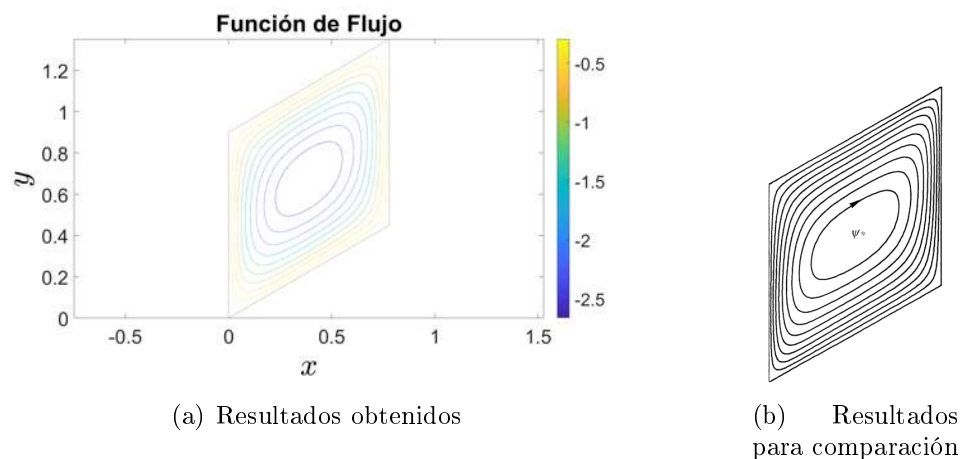


Figura 4.63: Comparación de la función de flujo obtenida con la metodología propuesta y la obtenida por Costa.

El campo de velocidades obtenido a partir de la función de flujo se muestra en la figura siguiente.

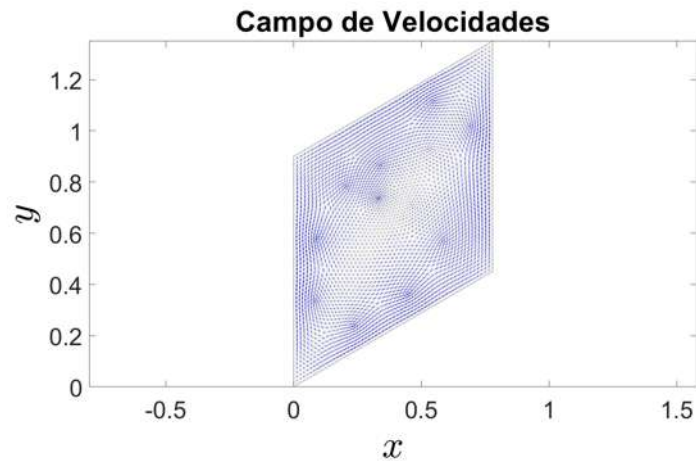


Figura 4.64: Campo de velocidades obtenido de la función de flujo.

En la siguiente figura se comparan los datos de la función de temperatura.

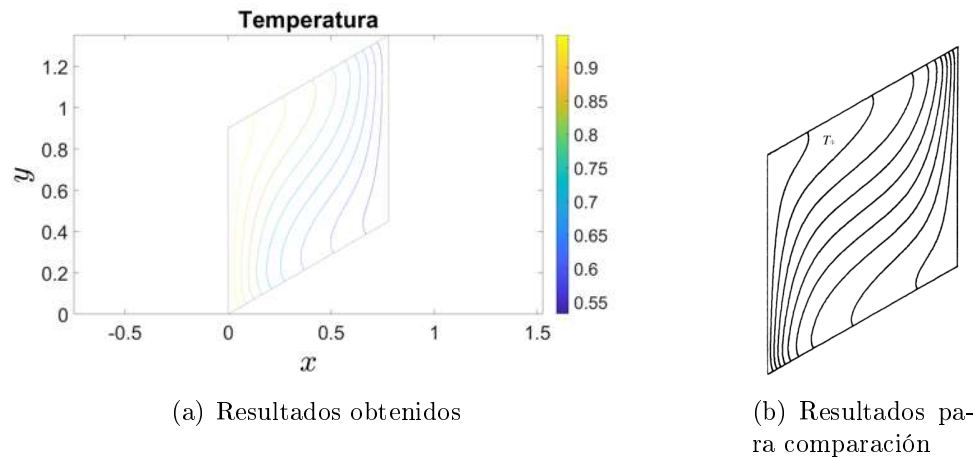


Figura 4.65: Comparación de la función de temperatura obtenida con la metodología propuesta y la obtenida por Costa.

Ejemplo 16

Para este ejemplo se usaron los parámetros siguientes: $Le = 0.8$, $N = 0$, $Ra = 100$, $H = 0.6$, $H/L = 0.25$, $\theta = 30^\circ$. Con los parámetros geométricos se obtiene la siguiente malla de puntos.

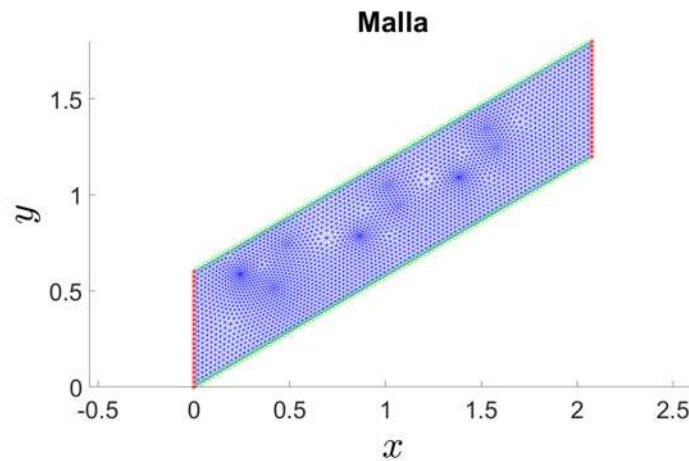


Figura 4.66: Malla generada usando los parámetros geométricos $H = 0.6$, $H/L = 0.25$, $\theta = 30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (o) representan la frontera con condiciones de Neumann.

Los resultados obtenidos para la función de flujo con la metodología propuesta se muestran del lado izquierdo, mientras que los obtenidos por Costa se muestran del lado derecho.

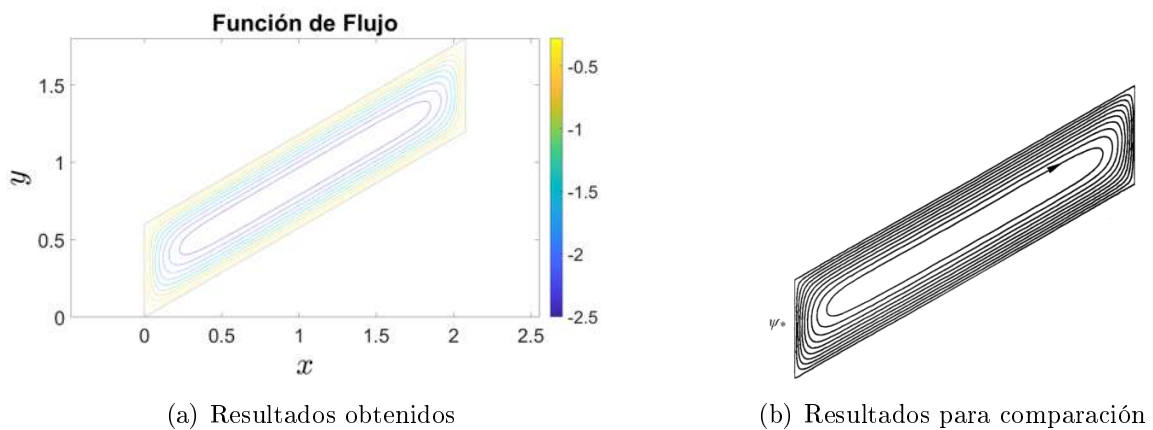


Figura 4.67: Comparación de la función de flujo obtenida con la metodología propuesta y la obtenida por Costa.

El campo de velocidades obtenido a partir de la función de flujo se muestra en la figura siguiente.

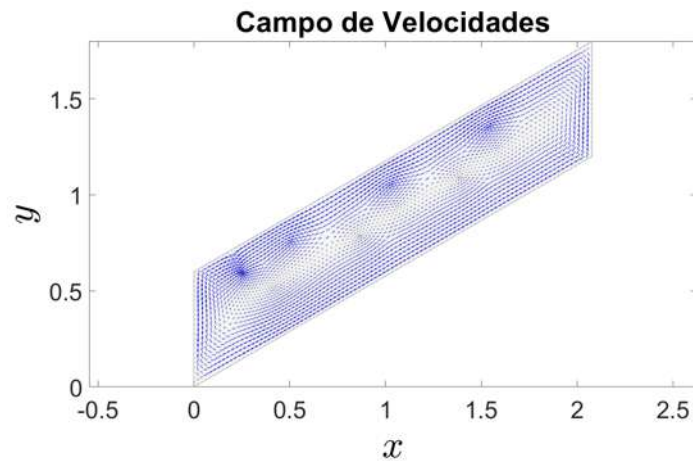
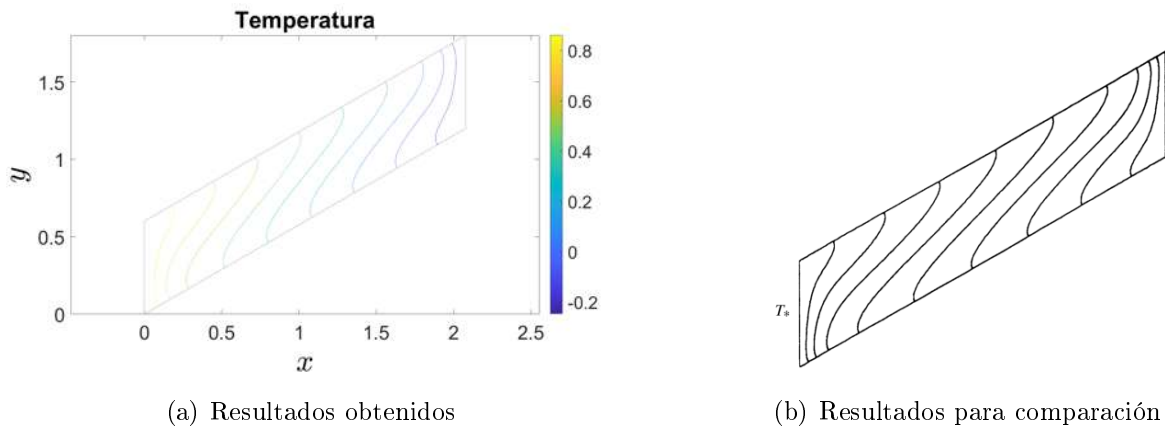


Figura 4.68: Campo de velocidades obtenido de la función de flujo.

En la siguiente figura se comparan los datos de la función de temperatura.



(a) Resultados obtenidos

(b) Resultados para comparación

Figura 4.69: Comparación de la función de temperatura obtenida con la metodología propuesta y la obtenida por Costa.

Ejemplo 17

Para este ejemplo se usaron los parámetros siguientes: $Le = 0.8$, $N = 0$, $Ra = 100$, $H = 0.5$, $H/L = 0.25$, $\theta = -30^\circ$. Con los parámetros geométricos se obtiene la siguiente malla de puntos.

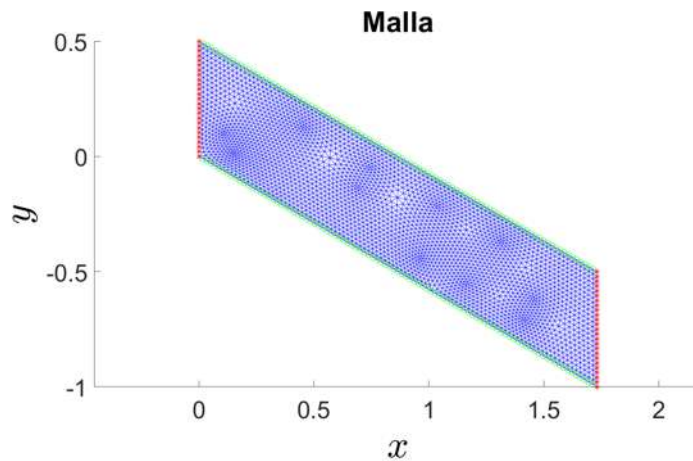


Figura 4.70: Malla generada usando los parámetros geométricos $H = 0.5$, $H/L = 0.25$, $\theta = -30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (o) representan la frontera con condiciones de Neumann.

Los resultados obtenidos para la función de flujo con la metodología propuesta se muestran del lado izquierdo, mientras que los obtenidos por Costa se muestran del lado derecho.

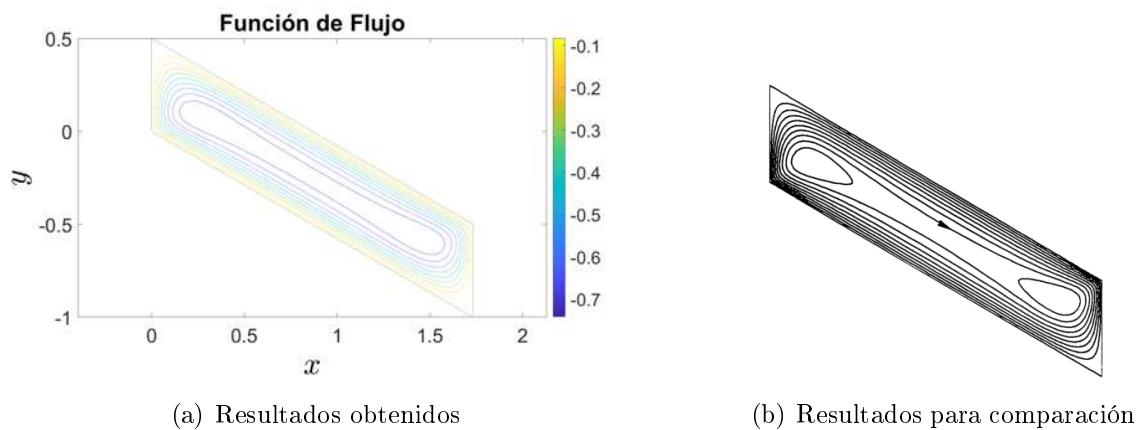


Figura 4.71: Comparación de la función de flujo obtenida con la metodología propuesta y la obtenida por Costa.

El campo de velocidades obtenido a partir de la función de flujo se muestra en la figura siguiente.

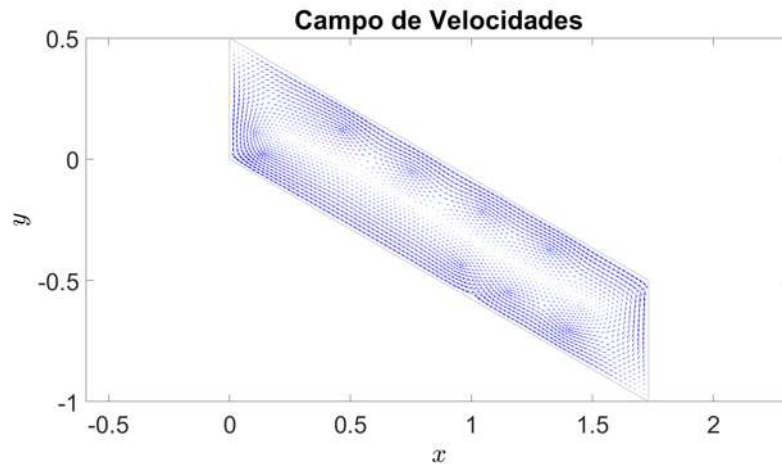
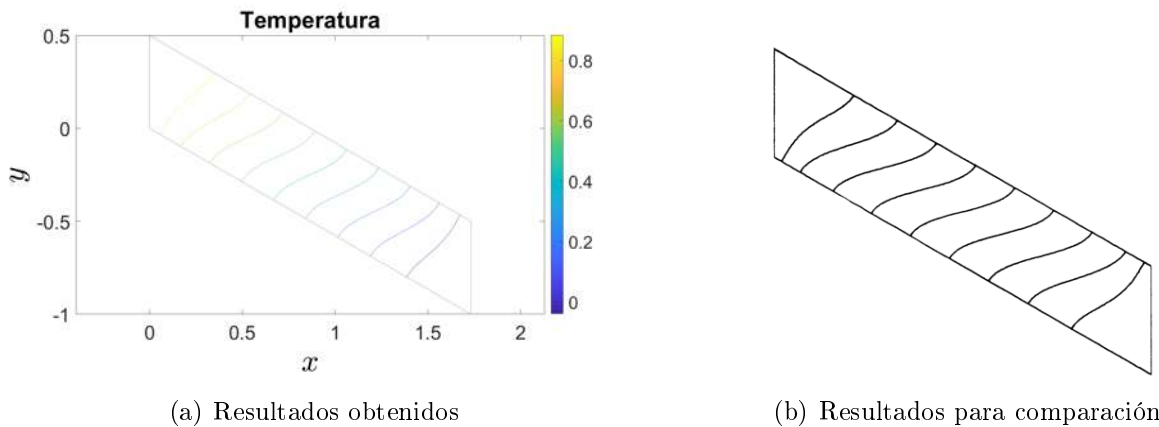


Figura 4.72: Campo de velocidades obtenido de la función de flujo.

En la siguiente figura se comparan los datos de la función de temperatura.



(a) Resultados obtenidos

(b) Resultados para comparación

Figura 4.73: Comparación de la función de temperatura obtenida con la metodología propuesta y la obtenida por Costa.

Ejemplo 18

Para este ejemplo se usaron los parámetros siguientes: $Le = 0.8$, $N = 2$, $Ra = 100$, $H = 0.7$, $H/L = 0.5$, $\theta = 30^\circ$. Con los parámetros geométricos se obtiene la siguiente malla de puntos.

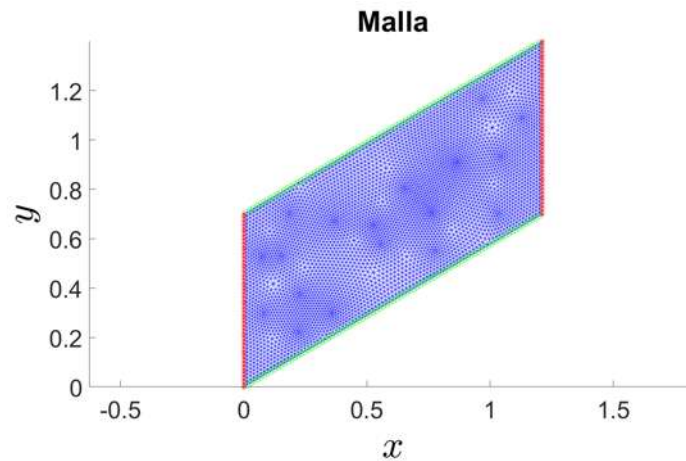


Figura 4.74: Malla generada usando los parámetros geométricos $H = 0.7$, $H/L = 0.5$, $\theta = 30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (o) representan la frontera con condiciones de Neumann.

Los resultados obtenidos para la función de flujo con la metodología propuesta se muestran del lado izquierdo, mientras que los obtenidos por Costa se muestran del lado derecho.

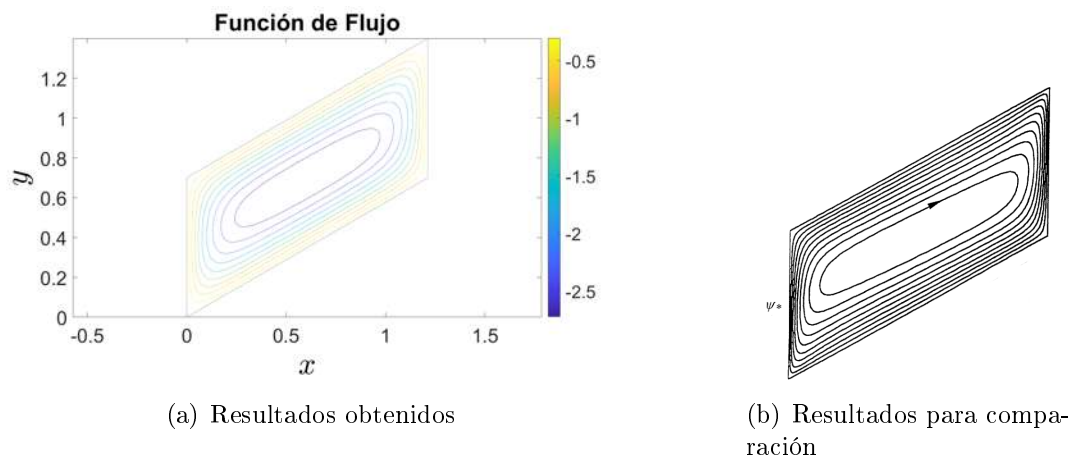


Figura 4.75: Comparación de la función de flujo obtenida con la metodología propuesta y la obtenida por Costa.

El campo de velocidades obtenido a partir de la función de flujo se muestra en la figura siguiente.

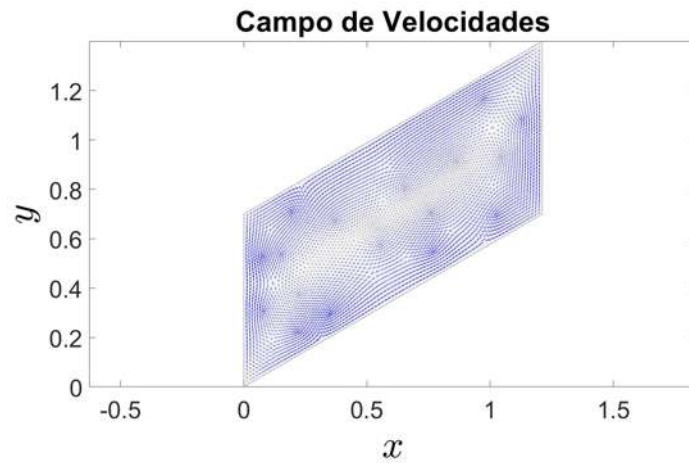


Figura 4.76: Campo de velocidades obtenido de la función de flujo.

En la siguiente figura se comparan los datos de la función de temperatura.

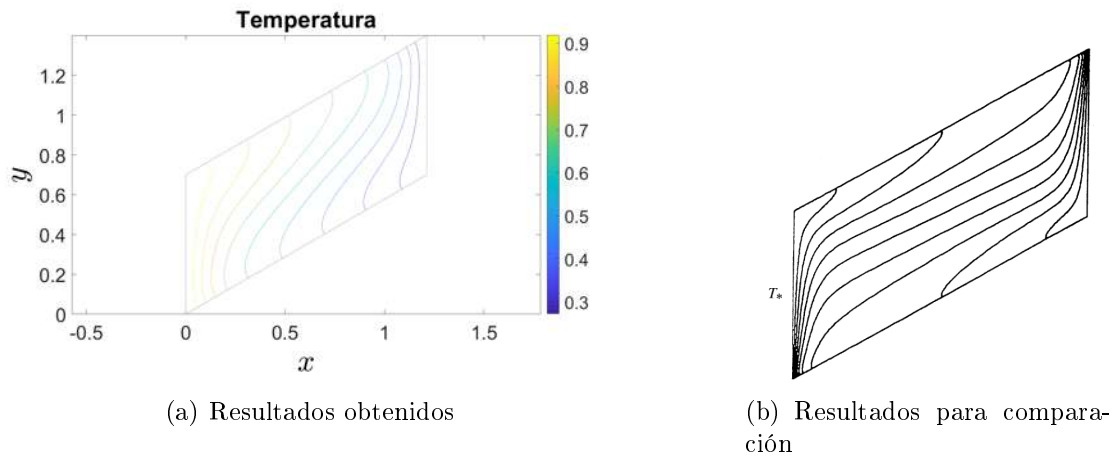


Figura 4.77: Comparación de la función de temperatura obtenida con la metodología propuesta y la obtenida por Costa.

Por último tenemos una comparación de los resultados obtenidos para la función de concentración.

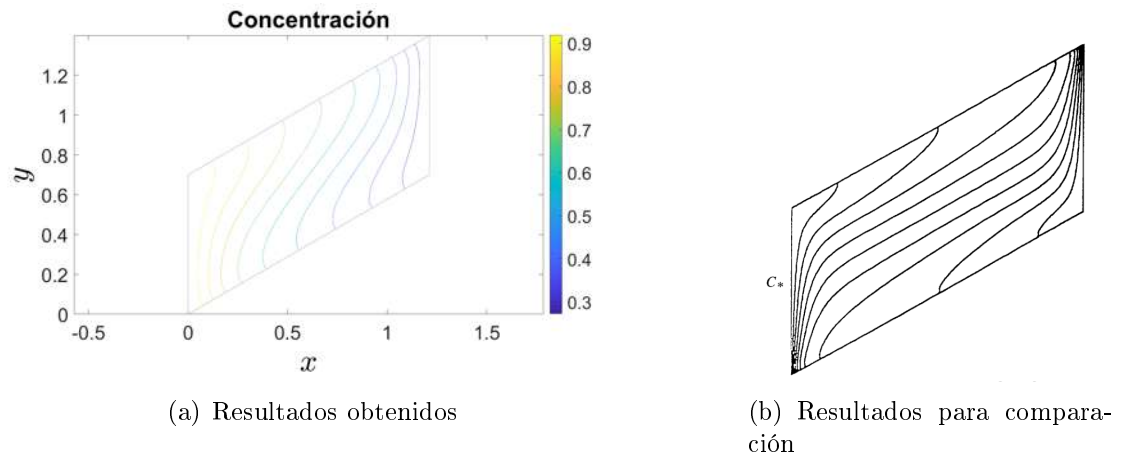


Figura 4.78: Comparación de la función de concentración obtenida con la metodología propuesta y la obtenida por Costa.

Ejemplo 19

Para este ejemplo se usaron los parámetros siguientes: $Le = 0.8$, $N = 5$, $Ra = 100$, $H = 0.8$, $H/L = 0.5$, $\theta = 30^\circ$. Con los parámetros geométricos se obtiene la siguiente malla de puntos.

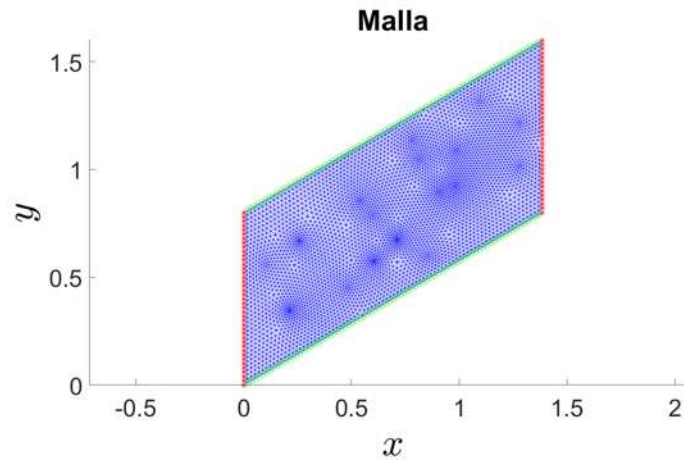


Figura 4.79: Malla generada usando los parámetros geométricos $H = 0.8$, $H/L = 0.5$, $\theta = 30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (o) representan la frontera con condiciones de Neumann.

Los resultados obtenidos para la función de flujo con la metodología propuesta se muestran del lado izquierdo, mientras que los obtenidos por Costa se muestran del lado derecho.

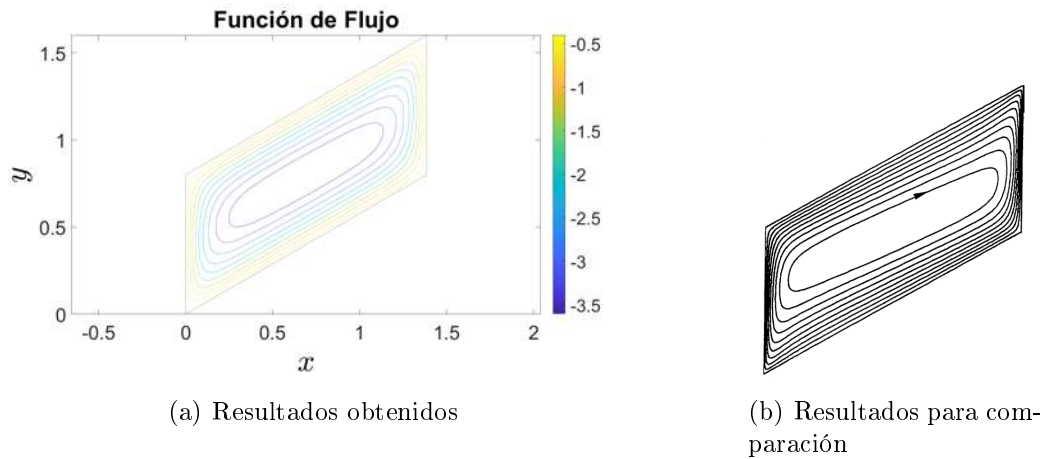


Figura 4.80: Comparación de la función de flujo obtenida con la metodología propuesta y la obtenida por Costa.

El campo de velocidades obtenido a partir de la función de flujo se muestra en la figura siguiente.

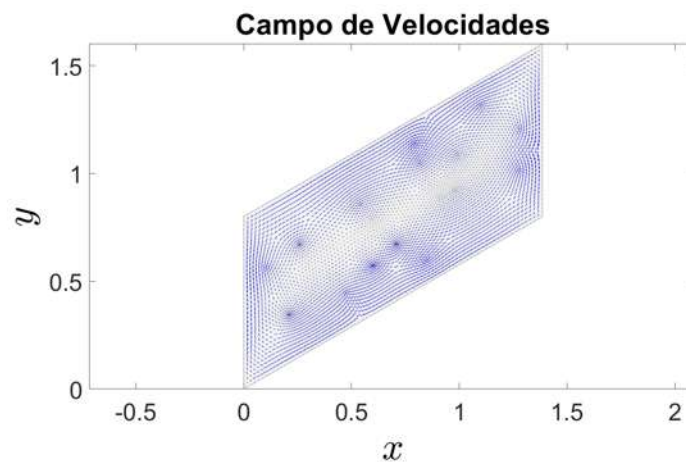


Figura 4.81: Campo de velocidades obtenido de la función de flujo.

En la siguiente figura se comparan los datos de la función de temperatura.

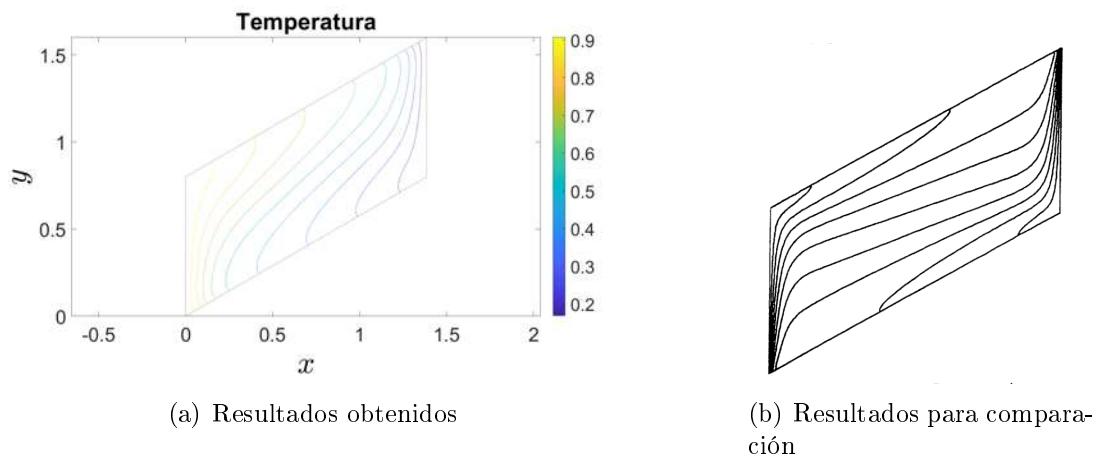


Figura 4.82: Comparación de la función de temperatura obtenida con la metodología propuesta y la obtenida por Costa.

Por último tenemos una comparación de los resultados obtenidos para la función de concentración.

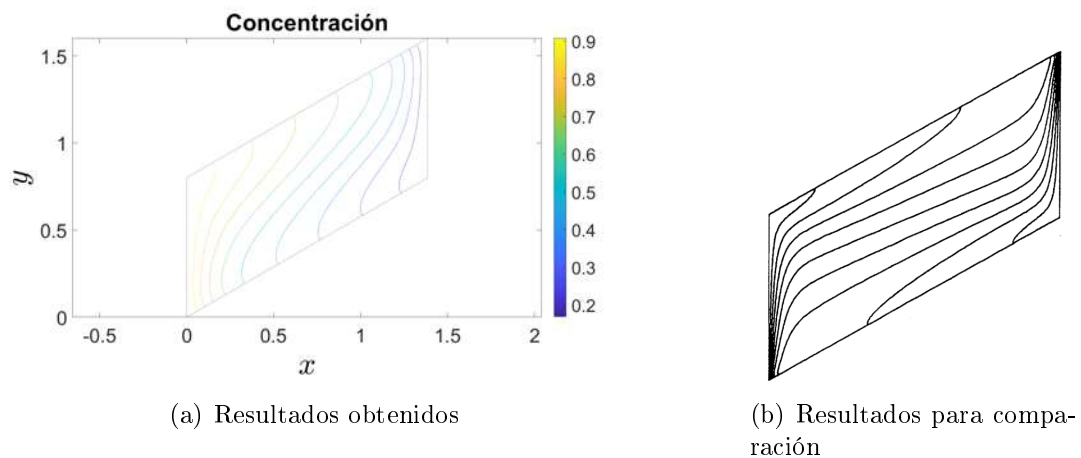


Figura 4.83: Comparación de la función de temperatura obtenida con la metodología propuesta y la obtenida por Costa.

Ejemplo 20

Para este ejemplo se usaron los parámetros siguientes: $Le = 0.8$, $N = 5$, $Ra = 100$, $H = 0.7$, $H/L = 0.5$, $\theta = 30^\circ$. Con los parámetros geométricos se obtiene la siguiente malla de puntos. Una particularidad de este ejemplo es que el flujo de temperatura y de concentración están invertidos, en otras palabras, las condiciones en la frontera izquierda son $T = 0$ y $C = 0$, mientras que en la frontera derecha son $T = 1$ y $C = 1$.

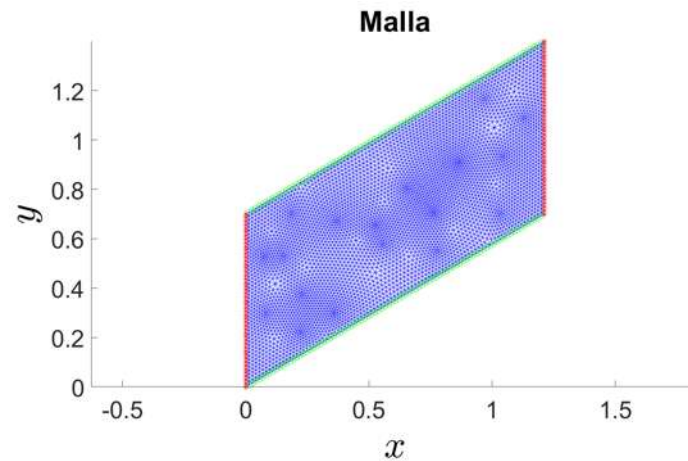


Figura 4.84: Malla generada usando los parámetros geométricos $H = 0.7$, $H/L = 0.5$, $\theta = 30^\circ$. Las líneas rojas con estrellas (*) representan la frontera con condiciones de Dirichlet, y las líneas verdes con círculos (o) representan la frontera con condiciones de Neumann.

Los resultados obtenidos para la función de flujo con la metodología propuesta se muestran del lado izquierdo, mientras que los obtenidos por Costa se muestran del lado derecho.

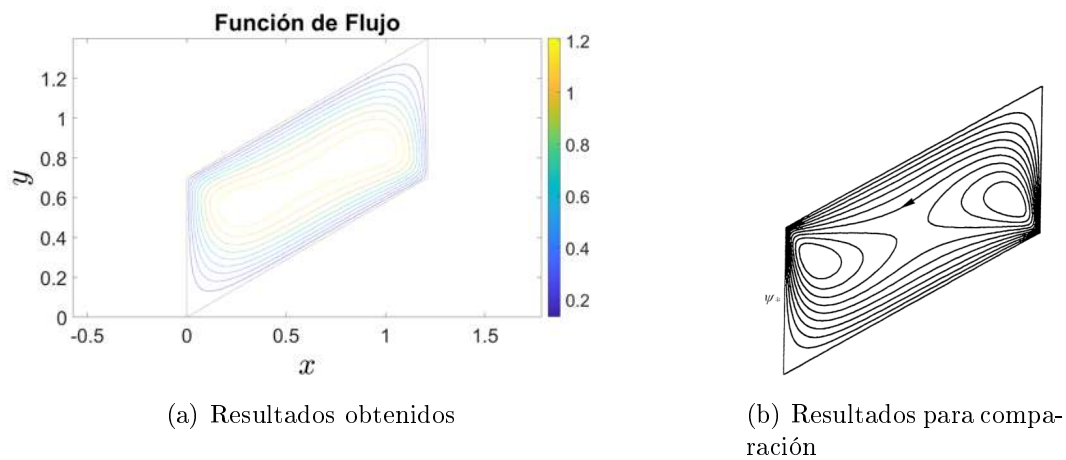


Figura 4.85: Comparación de la función de flujo obtenida con la metodología propuesta y la obtenida por Costa.

El campo de velocidades obtenido a partir de la función de flujo se muestra en la figura siguiente.

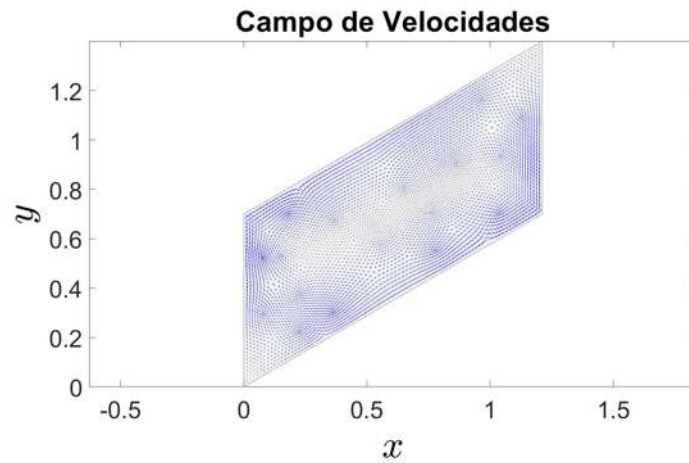


Figura 4.86: Campo de velocidades obtenido de la función de flujo.

En la siguiente figura se comparan los datos de la función de temperatura.

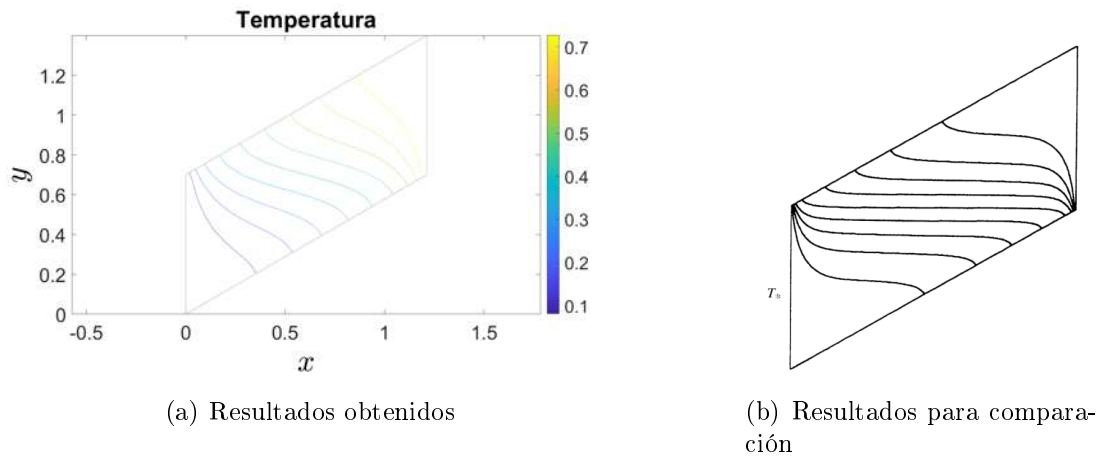
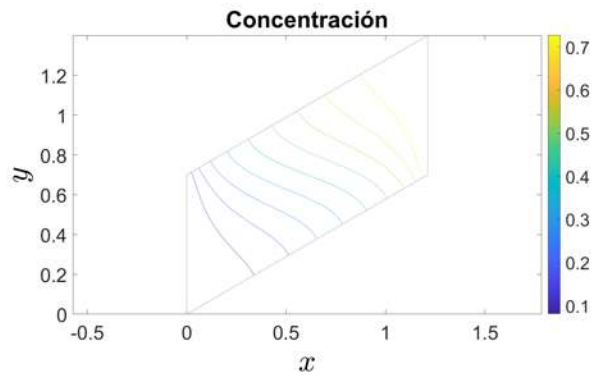
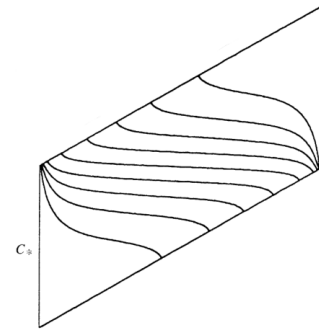


Figura 4.87: Comparación de la función de temperatura obtenida con la metodología propuesta y la obtenida por Costa.

Por último tenemos una comparación de los resultados obtenidos para la función de concentración.



(a) Resultados obtenidos



(b) Resultados para comparación

Figura 4.88: Comparación de la función de concentración obtenida con la metodología propuesta y la obtenida por Costa.

Capítulo 5

DISCUSIÓN Y CONCLUSIONES

Comparando las gráficas de temperatura obtenidas con aquellas de los artículos de Costa (Costa, 2004) y de Fan (Fan et al., 2013) se puede distinguir que las segundas resultan ser más alargadas, mientras que en el primer caso se obtienen resultados menos pronunciados. Lo mismo se puede notar con las gráficas de concentración obtenidas en este trabajo al compararles con las de los mismos autores. Sin embargo, las gráficas de la función de flujo tienen una similitud mayor a las de los artículos antes mencionados.

Una dificultad que se presentó es que en los artículos de comparación no se menciona el valor de H que utilizan, por lo que los valores utilizados en este trabajo son propuestas propias, como consecuencia puede que esto también afecte en la variación de los resultados obtenidos y los de los artículos de comparación. Otra observación importante es que, si se usan valores de H mayores que 1, entonces las soluciones comienzan a perder buen comportamiento. Por otro lado, el valor H que mejor funcionó está entre 0.6 y 0.7.

Aunque no se realizó un procedimiento para resolver el problema no lineal directamente, se obtuvo una aproximación a la solución, lo cual se logró al desacoplar las tres ecuaciones diferenciales haciendo una aproximación lineal de las funciones de temperatura y concentración para obtener la función de flujo, o en otras palabras, convirtiendo el sistema de ecuaciones no lineales en un sistema de ecuaciones lineales. Una vez hecho esto se resolvió el sistema mediante un algoritmo predictor-corrector, gracias a esto ya hemos obtenido resultados satisfactorios al obtener las funciones que describen el problema (función de flujo, temperatura y concentración). Cabe mencionar que el sistema de ecuaciones lineales obtuvo una buena aproximación después de varias iteraciones para todos los experimentos.

Se utilizaron las funciones de MatLAB *initmesh* y *jigglemesh*, las cuales inicializan y suavizan una malla respectivamente, para crear las mallas utilizadas en los ejemplos propuestos por Fan y los propuestos por Costa, con las cuales se obtuvieron resultados satisfactorios para dichos ejemplos.

Es importante mencionar que se puso un tope máximo de iteraciones de 50, el cual es un valor propuesto, y sirve como criterio de seguridad. El número de iteraciones requeridas para obtener una solución depende del valor de H del problema, ya que, a mayor H mayor es el número de iteraciones necesarias para alcanzar el criterio de convergencia (3.15). Para los ejemplos tomados de los artículos de Fan y de Costa, el límite máximo de iteraciones se alcanzó para todas las simulaciones.

La metodología de diferencias finitas generalizadas se puede implementar en diversos problemas físicos descritos con ecuaciones diferenciales, uno de ellos puede ser el problema resuelto en este trabajo, el cual es el problema de convección natural con doble difusividad, este problema que se puede implementar para simular el comportamiento del agua en el subsuelo, y también incluir alguna sustancia que se disuelva en esta agua, como por ejemplo algún contaminante.

ANEXOS

Anexo I: Código de la primera aproximación

A continuación se muestra el código de MatLAB utilizado en la primera aproximación con la primera malla propuesta.

```
1 close
2 %Par \ 'a metros
3 Ra=100;N=2;Le=0.8;
4 load malla
5
6 tola=1e-2;tolr=1e-4;
7 err=[];
8 flag=1;iter=1;itermax=100;
9
10 finicial=@(x) 2/3*Ra+2/3*N*Le;
11 cero=@(x) zeros(size(x,1),1);
12 g2=@(x) 1-x(:,1)*2/3;
13
14 psi=dirichgfd(p,B,t,finicial,cero,1,0,1,0,0,0);
15
16 [~,DX,~]=dirichgfd(p,B,t,cero,cero,0,0,0,1,0,0);
17 [~,DY,~]=dirichgfd(p,B,t,cero,cero,0,0,0,0,1,0);
18 pp=DY*full(psi);
19 qq=-DX*full(psi);
20
21 T=robingfd_conv(p,b,bn,t,cero,g2,@dn,-pp,-qq,@VN3);
22
23 C=robingfd_conv(p,b,bn,t,cero,g2,@dn,-Le*pp,-Le*qq,@VN3);
24
25 r0=norm(psi);
26
27 while (flag)&&(iter<itermax),
28
29 [~,DX,~]=dirichgfd(p,[],t,cero,cero,0,0,0,1,0,0);
30 DXT=DX*(T);
31 DXC=DX*(C);
32 f=-(DXT*Ra+DXC*N*Le);
33 psinew=dirichgfd(p,B,t,f,cero,1,0,1,0,0,0);
```

```

34 [~,DX,~]=dirichgfd(p,B,t,cero,cero,0,0,0,1,0,0);
35 [~,DY,~]=dirichgfd(p,B,t,cero,cero,0,0,0,0,1,0);
36 pp=DY*full(psinew);
37 qq=-DX*full(psinew);
38 Tnew=robingfd_conv(p,b,bn,t,cero,g2,@dn,-pp,-qq,@VN3);
39 Cnew=robingfd_conv(p,b,bn,t,cero,g2,@dn,-Le*pp,-Le*qq,@VN3);
40
41 error=norm(psi-psinew)+norm(T-Tnew)+norm(C-Cnew);
42 err=[err error];
43
44 display(['iter=' num2str(iter) ' error=' num2str(error)])
45
46 if error<tolr*r0+tola; %error<(norm(x0)+1)*tol,
47     flag=0; %alida
48 end
49 psi=psinew;T=Tnew;C=Cnew;
50 iter=iter+1;
51 end
52
53 figure('units','normalized','outerposition',[0 0 1 1])
54 subplot(221)
55 trisurf(t,p(:,1),p(:,2),full(psinew));view(2);shading interp;colorbar
56 subplot(222)
57 quiver(p(:,1),p(:,2),pp,qq)
58 subplot(223)
59 trisurf(t,p(:,1),p(:,2),full(T));view(2);shading interp;colorbar
60 subplot(224)
61 trisurf(t,p(:,1),p(:,2),full(C));view(2);shading interp;colorbar
62
63 figure
64 tricontour(t,p(:,1),p(:,2),full(psi),[-2.5:0.1:-0.1]);
65 figure
66 tricontour(t,p(:,1),p(:,2),full(C),[0.1:0.1:1]);
67 figure
68 tricontour(t,p(:,1),p(:,2),full(T),[0.1:0.1:1]);

```

Anexo II: Código *dirichgfd.m*

Este código hace la diferenciación con el esquema GFDM propuesto para la malla, con condiciones de Dirichlet a la frontera.

```

1 function [U,K,F]=dirichgfd(p,b,t,f,g,A,B,C,D,E,FF)
2 %diferencias dfg
3 N=size(p,1);T=size(t,1); %n|'umero de nodos, n|'umero de tri|'angulos
4 K=sparse(N,N);F=sparse(N,1);
5 %Ahora necesitamos una estructura de datos poderosa
6 nb=setdiff(1:N,b); %Estos no son frontera

```

```

7
8 K=matrizd(p,t,A,B,C,D,E,FF);
9
10 % for i=1:N
11 %         if isfloat(f)
12 %             F(i)=F(i)+f(i);
13 %         else
14 %             F(i)=F(i)+f(p(i,:));
15 %         end
16 %         Gamma=[];
17 % end
18 if isfloat(f)
19     F=f([1:N]');
20 else
21     F=f(p);
22 end
23
24 % Ahora de a~naden las condiciones de Dirichlet en K and F
25 F=F-sum(kron(g(p(b,:))',ones(N,1)).*K(:,b),2); %Valores que pasan al lado derecho
26 F(b)=[g(p(b,:))]; %Valores en la frontera
27 K(b,:)=0; K(:,b)=0; %Ponemos ceros para no modificar los valores en la frontera
28 K(b,b)=speye(length(b),length(b)); % se pone I en la submatriz "frontera" de K
29
30 % Ahora resolvemos
31 U=K\F; %La aproximaci~on es U_1 phi_1 + ... + U_N phi_N

```

Anexo III: C3gigo *matrizd.m*

Juntan con el c3digo *dirichgfd.m* implementan el esquema GFDM con condiciones de Dirichlet para la malla propuesta.

```

1 function K=matrizd(p,t,A,B,C,D,E,FF)
2 %diferencias dfg
3 N=size(p,1); %A=size(t,1); %n'umero de nodos, n'umero de tri'angulos
4 K=sparse(N,N);
5 %Ahora necesitamos una estructura de datos poderosa
6
7 for i=1:N
8     I=any(ismember(t,i),2); %% buscamos los indices donde anda i
9     tt=t(I,:); tt=tt(:);
10    I2=setdiff(tt,i); %%Estos son los vecinos de i
11
12    u=p(I2,1)'; v=p(I2,2)';
13
14        dx=u-p(i,1); dy=v-p(i,2);
15
16    M=[dx; dy; dx.*dx; dx.*dy; dy.*dy];

```

```

17         Gamma=pinv(M)*[D E 2*A B 2*C]';
18         Gamma=[FF-sum(Gamma);Gamma];
19
20         K(i,[i;I2])=Gamma;
21
22         Gamma=[];
23 end

```

Anexo IV: Código *robngfd_conv.m*

Este código implementa el esquema GFDM para condiciones mixtas (condiciones de Dirichlet y condiciones de Neumann), o también llamadas condiciones de Robin. Este sólo se usa para las soluciones de las funciones de temperatura T y de concentración C , debido a que estas funciones son las que tienen condiciones de frontera mixtas.

```

1 function [U,K,F,NB]=robngfd_conv(p,b,bn,t,f,g,dn,DX,DY,VN)
2 %diferencias fem a ver que pasa
3 N=size(p,1);T=size(t,1); %n\ 'umero de nodos, n\ 'umero de tri\ 'angulos
4 K=sparse(N,N);F=sparse(N,1);
5 %Ahora necesitamos una estructura de datos poderosa
6 nb=setdiff(1:N,b); %Estos no son frontera
7
8 %L=L(:);%DN=DN(:);
9
10 for i=nb
11     I=any(ismember(t,i),2); %%buscamos los indices donde anda i
12     tt=t(I,:);tt=tt(:);
13     I2=setdiff(tt,i); %%Estos son los vecinos de i
14     NB(nb)=length(I2);
15     u=p(I2,1)';v=p(I2,2)';
16
17     dx=u-p(i,1);dy=v-p(i,2);
18
19     M=[dx;dy;dx.*dx;dx.*dy;dy.*dy];
20     if ismember(i,bn),
21         Gamma=pinv(M(1:2,:))*VN(p(i,:));
22         F(i)=F(i)+dn(p(i,:));
23     else
24         L=[DX(i) DY(i) 2 0 2]';
25         Gamma=pinv(M)*L;
26         F(i)=F(i)+f(p(i,:));
27     end
28     Gamma=[-sum(Gamma);Gamma];
29
30     K(i,[i;I2])=Gamma;
31
32     Gamma=[];

```

```

33 end
34
35 % Ahora de a~naden las condiciones de Dirichlet en K and F
36 F=F-sum(kron(g(p(b,:))',ones(N,1)).*K(:,b),2); %Valores que pasan al lado derecho
37 F(b)=[g(p(b,:))]; %Valores en la frontera
38 K(b,:)=0; K(:,b)=0; %Ponemos ceros para no modificar los valores en la frontera
39 K(b,b)=speye(length(b),length(b)); % se pone I en la submatriz "frontera" de K
40
41 % Ahora resolvemos
42 U=K\F; %La aproximaci~on es U_1 phi_1 + ... + U_N phi_N

```

Anexo V: Segunda aproximación

Este código realiza la segunda aproximación, donde la malla utilizada es una malla rectangular.

```

1 close; clear; clc;
2 %% load par~ametros
3 Ra=100;N=2;Le=0.8;
4 load mallar2
5 %B = nodos de frontera
6 % b = nodos frontera con condiciones de Dirichlet
7 % bn = nodos frontera con condiciones de Neumman
8 % p = coordenadas x y de los nodos
9 % t = tri~angulos
10
11 %% Tolerancias y errores
12 tola=1e-2;tolr=1e-4; %tola=0.01;tolr=0.0001;
13 err=[];
14 flag=1;iter=1;itermax=100; %iteraci~on m~axima
15
16 %% ecuaci~on del plano de aproximaci~on para T y C
17 % z=1-2/3*p(:,1);
18 % trimesh(t,p(:,1),p(:,2),z); colorbar
19 % close
20
21 %% Funciones
22 finicial=@(x) 2/3*Ra+2/3*N*Le; %tercer t~ermino de la ec. gobernante 1
23 cero=@(x) zeros(size(x,1),1);
24 g2=@(x) 1-x(:,1)*2/3; %ecuaci~on del plano de aproximaci~on T y C
25
26 %% Primera aproximaci~on
27 %% Funci~on de flujo
28 psi=dirichgfd(p,B,t,finicial,cero,1,0,1,0,0,0); %primera aproximaci~on a psi
29 %plano %lado derecho
30
31 %% Velocidades de flujo
32 [~,DX,~]=dirichgfd(p,B,t,cero,cero,0,0,0,1,0,0);

```

```

33 [~,DY,~]=dirichgfd(p,B,t,cero,cero,0,0,0,0,1,0);
34 pp=DY*full(psi);
35 qq=DX*full(psi);
36 %% Aproximaci\ 'on de las funciones Temperatura y Concentraci\ 'on
37 T=robingfd_conv2(p,b,bn,t,cero,g2,@dn,-pp,-qq,@VN3);
38 C=robingfd_conv2(p,b,bn,t,cero,g2,@dn,-Le*pp,-Le*qq,@VN3);
39
40
41 %% Diferentes formas de c\ 'alculo para DX and DY
42
43 %DX=matrizd(p,t,0,0,0,1,0,0);
44 %DY=matrizd(p,t,0,0,0,0,1,0);
45 [~,DX,~]=dirichgfd(p,[],t,cero,cero,0,0,0,1,0,0);
46 [~,DX,~]=robingfd_2(p,[],B,t,cero,cero,@dn,[1 0 0 0 0 0],@VN3);
47
48 [~,DX,~]=robingfd_3(p,B,[],t,cero,cero,@dn,[1 0 0 0 0 0],@VN3);
49 [~,DY,~]=robingfd_3(p,B,[],t,cero,cero,@dn,[0 1 0 0 0 0],@VN3);
50
51
52 %% Corrector Predictor
53 while (flag)&&(iter<itermax)
54 DX=matrizd_3(p,t,[],cero,cero,[1 0 0 0 0 0],@VN3);
55 DY=matrizd_3(p,t,[],cero,cero,[0 1 0 0 0 0],@VN3);
56
57 DXT=DX*(T);
58 DXC=DX*(C);
59 f=-(DXT*Ra+DXC*N*Le);
60 psinew=dirichgfd(p,B,t,f,cero,1,0,1,0,0,0);
61 [~,DX,~]=dirichgfd(p,B,t,cero,cero,0,0,0,1,0,0);
62 [~,DY,~]=dirichgfd(p,B,t,cero,cero,0,0,0,0,1,0);
63 pp=DY*full(psinew);
64 qq=-DX*full(psinew);
65 Tnew=robingfd_conv2(p,b,bn,t,cero,g2,@dn,-pp,-qq,@VN3);
66 Cnew=robingfd_conv2(p,b,bn,t,cero,g2,@dn,-Le*pp,-Le*qq,@VN3);
67
68 error=norm(psi-psinew)+norm(T-Tnew)+norm(C-Cnew);
69
70 display(['iter=' num2str(iter) '\_error=' num2str(error)])
71
72 % err=[err error];
73 % if iter >=2
74 % if abs(err(end-1)-err(end)) < 0.00001
75 %     flag=0;
76 % end
77 % end
78
79 r0=norm(psi);
80 if error < tol*r0+tola; %error < (norm(x0)+1)*tol,

```

```

81     flag=0; %salida
82 end
83 psi=psinew;T=Tnew;C=Cnew;
84 iter=iter+1;
85 end
86
87 %% Gráfica de los Resultados
88
89 % trisurf(t,p(:,1),p(:,2),full(reshape(psinew,33,33)))
90 % pause
91
92 figure('units','normalized','outerposition',[0 0 1 1])
93
94 subplot(233)
95 trisurf(t,p(:,1),p(:,2),full(psinew));view(2);shading interp;colorbar
96 subplot(231)
97 trisurf(t,p(:,1),p(:,2),full(T));view(2);shading interp;colorbar
98 subplot(232)
99 trisurf(t,p(:,1),p(:,2),full(C));view(2);shading interp;colorbar
100
101 X=reshape(p(:,1),33,33);
102 Y=reshape(p(:,2),33,33);
103
104 subplot(236)
105 contour(X,Y,full(reshape(psinew,33,33)));colorbar
106 subplot(234)
107 contour(X,Y,full(reshape(Tnew,33,33)));colorbar
108 subplot(235)
109 contour(X,Y,full(reshape(Cnew,33,33)));colorbar
110
111 figure
112 % plot([0 1.5 1.5 0 0],[0 0.4 1 0.6 0],'k')
113 % hold on
114 quiver(p(:,1),p(:,2),2*pp,2*qq)

```

Anexo VI: Código *robingfd_conv2.m*

Es una modificación del código *robingfd_conv.m*, donde se considera un nodo fantasma para aproximar las condiciones de Neumann.

```

1 function [U,K,F,NB]=robingfd_conv2(p,b,bn,t,f,g,dn,DX,DY,VN)
2 % resuelve la eciación gobernante 2 (Temperatura) o 3 (Concentración)
3 %diferencias fem a ver que pasa
4 N=size(p,1);T=size(t,1); %n número de nodos, n número de triángulos
5 K=sparse(N,N);F=sparse(N,1);
6 %Ahora necesitamos una estructura de datos poderosa
7 nb=setdiff(1:N,b); %Estos no son frontera

```

```

8
9   $\mathcal{L}=L(:); \%DN=DN(:);$ 
10
11 for i=nb
12     I=any(ismember(t,i),2); %% buscamos los indices donde anda i
13     tt=t(I,:); tt=tt(:);
14     I2=setdiff(tt,i); %%Estos son los vecinos de i
15      $\mathcal{NB}(nb)=length(I2);$ 
16     u=p(I2,1)'; v=p(I2,2)';
17
18     if ismember(i,bn),
19         u=[u 2*p(i,1)-mean(u)]; v=[v 2*p(i,2)-mean(v)];
20     end
21
22         dx=u-p(i,1); dy=v-p(i,2);
23
24         M=[dx; dy; dx.*dx; dx.*dy; dy.*dy];
25
26         Gamma=pinv(M)*[DX(i) DY(i) 2 0 2]'; %%resuelve (M,L);
27         Gamma=[-sum(Gamma);Gamma];
28
29         F(i)=F(i)+f(p(i,:));
30
31         if ismember(i,bn),
32             GammaN=pinv(M)*[VN(p(i,:)); zeros(3,1)]; %%resuelve (M,[VN(p(i,:)); zeros
33             GammaN=[-sum(GammaN);GammaN];
34             %%F(i)=F(i)+dn(p(i,:))/GammaN(end); 270919
35             F(i)=F(i)+Gamma(end)*dn(p(i,:))/GammaN(end);
36             GammaN=-GammaN(1:(end-1))/GammaN(end)*Gamma(end);
37             Gamma=Gamma(1:(end-1))+GammaN;
38         end
39
40         K(i,[i;I2])=Gamma;
41
42         Gamma=[];
43 end
44
45 %% Ahora de a|~naden las condiciones de Dirichlet en K and F
46 F=F-sum(kron(g(p(b,:))',ones(N,1)).*K(:,b),2); %%Valores que pasan al lado derecho
47 F(b)=[g(p(b,:))]; %%Valores en la frontera
48 K(b,:)=0; K(:,b)=0; %%Ponemos ceros para no modificar los valores en la frontera
49 K(b,b)=speye(length(b),length(b)); %% se pone I en la submatriz "frontera" de K
50
51 %% Ahora resolvemos
52 U=K\F; %% La aproximaci|'on es U_1 phi_1 + ... + U_N phi_N

```

Anexo VII: Código *matrizd_3.m*

Es una modificación del código *matrizd.m*, donde se considera un nodo fantasma para aproximar las condiciones de Neumman.

```

1 function [K,F]=matrizd_3(p,t,bn,f,g,L,VN)
2 %diferencias dfg
3 N=size(p,1);
4 K=sparse(N,N);
5 F=sparse(N,1);
6 FF=L(6);
7 L=L(1:5)';
8
9 for i=1:N
10     I=any(ismember(t,i),2); %% buscamos los indices donde anda i
11     tt=t(I,:); tt=tt(:);
12     I2=setdiff(tt,i); %%Estos son los vecinos de i
13     u=p(I2,1)'; v=p(I2,2)';
14
15     if ismember(i,bn)
16         u=[u 2*p(i,1)-mean(u)];
17         v=[v 2*p(i,2)-mean(v)];
18     end
19
20         dx=u-p(i,1); dy=v-p(i,2);
21     M=[dx;dy;dx.*dx;dx.*dy;dy.*dy];
22     Gamma=pinv(M)*L;
23     Gamma=[FF-sum(Gamma);Gamma];
24     F(i)=F(i)+f(p(i,:));
25
26     if ismember(i,bn),
27         GammaN=pinv(M)*[VN(p(i,:));zeros(3,1)];
28         GammaN=[-sum(GammaN);GammaN];
29         F(i)=F(i)-g(p(i,:))/GammaN(end)*Gamma(end);
30         GammaN=-GammaN(1:(end-1))/GammaN(end)*Gamma(end);
31         Gamma=Gamma(1:(end-1))+GammaN;
32     end
33
34     K(i,[i;I2])=Gamma;
35     Gamma=[];
36 end

```

Anexo VIII: Código para ejemplos con mallas generadas automáticamente

Con este código se resuelven ejemplos para mallas generadas automáticamente.

```

1  clear; clc; close;
2  %% Par \ 'ametros
3  Ra=100;N=2;Le=0.8;
4  H=0.7; % Altura del paralelogr \ 'amo
5  HsobreL=0.25; % Proporción H/L
6  L=H/HsobreL; % Longitud del paralelogr \ 'amo
7  theta=30; % \ 'Angulo en grados del paralelogr \ 'amo
8  theta=theta*pi/180; % \ 'Angulo en radianes
9  %% Malla
10 delta=0.01;
11 NN=round(H/delta); % NN*NN = Nodos en la malla
12 [p,t,b,bn,B]=GenM3(L,theta,H,NN);
13
14 % B = nodos de frontera
15 % b = nodos frontera con condiciones de Dirichlet
16 % bn = nodos frontera con condiciones de Neumann
17 % p = coordenadas x y de los nodos
18 % t = tri \ 'angulos
19 %% Tolerancias y errores
20 tola=1e-2;tolr=1e-4; % tola=0.01;tolr=0.0001;
21 err=[];
22 flag=1;iter=1;itermax=50; % iteraci \ 'on m \ 'axima
23
24 %% ecuaci \ 'on del plano de aproximaci \ 'on para T y C
25 % z=1-2/3*p(:,1);
26 % trimesh(t,p(:,1),p(:,2),z); colorbar
27 % close
28
29 %% Funciones
30 finicial=@(x) 2/3*Ra+2/3*N*Le; % tercer t \ 'ermino de la ec. gobernante 1
31 cero=@(x) zeros(size(x,1),1);
32 g2=@(x) 1-x(:,1)*2/3; % ecuaci \ 'on del plano de aproximaci \ 'on T y C
33 % g2=@(x) x(:,1)*2/3; % ecuaci \ 'on del plano de flujo invertido
34 %% Primera aproximaci \ 'on
35 %% Funci \ 'on de flujo
36 psi=dirichgfd(p,B,t,finicial,cero,1,0,1,0,0,0); % primera aproximaci \ 'on a psi
37 %plano %ado derecho
38
39 %% Velocidades de flujo
40 [~,DX,~]=dirichgfd(p,B,t,cero,cero,0,0,0,1,0,0);
41 [~,DY,~]=dirichgfd(p,B,t,cero,cero,0,0,0,0,1,0);
42 pp=DY*full(psi);
43 qq=DX*full(psi);
44 %% Aprximaci \ 'on de las funciones Temperatura y Concentraci \ 'on
45 T=robingfd_conv3(p,b,bn,t,cero,g2,@dn,-pp,-qq,@VN4,theta,H,NN);
46 C=robingfd_conv3(p,b,bn,t,cero,g2,@dn,-Le*pp,-Le*qq,@VN4,theta,H,NN);
47
48

```

```

49 %% Diferentes formas de c\ 'alculo para DX and DY
50
51 %DX=matrizd(p,t,0,0,0,1,0,0);
52 %DY=matrizd(p,t,0,0,0,0,1,0);
53 [%~,DX,~]=dirichgfd(p,[],t,cero,cero,0,0,0,1,0,0);
54 [%~,DX,~]=robingfd_2(p,[],B,t,cero,cero,@dn,[1 0 0 0 0 0],@VN3);
55
56 [%~,DX,~]=robingfd_3(p,B,[],t,cero,cero,@dn,[1 0 0 0 0 0],@VN3);
57 [%~,DY,~]=robingfd_3(p,B,[],t,cero,cero,@dn,[0 1 0 0 0 0],@VN3);
58
59
60 %% Corrector Predictor
61 while (flag)&&(iter<itermax)
62 DX=matrizd_4(p,t,[],cero,cero,[1 0 0 0 0 0],@VN4,theta); %Derivada en x para condiciones de D
63 DY=matrizd_4(p,t,[],cero,cero,[0 1 0 0 0 0],@VN4,theta); %Derivada en y para condiciones de D
64
65 DXT=DX*(T);
66 DXC=DX*(C);
67 f= -(DXT*Ra+DXC*N*Le);
68 psinew=dirichgfd(p,B,t,f,cero,1,0,1,0,0,0);
69 [%~,DX,~]=dirichgfd(p,B,t,cero,cero,0,0,0,1,0,0); %Derivada en x para condiciones de D
70 [%~,DY,~]=dirichgfd(p,B,t,cero,cero,0,0,0,0,1,0); %Derivada en y para condiciones de D
71 pp=DY*full(psinew);
72 qq=-DX*full(psinew);
73 Tnew=robingfd_conv3(p,b,bn,t,cero,g2,@dn,-pp,-qq,@VN4,theta,H,NN);
74 Cnew=robingfd_conv3(p,b,bn,t,cero,g2,@dn,-Le*pp,-Le*qq,@VN4,theta,H,NN);
75
76 error=norm(psi-psinew)+norm(T-Tnew)+norm(C-Cnew);
77
78 display(['iter=' num2str(iter) ' error=' num2str(error)])
79
80 r0=norm(psi);
81 if error<tolr*r0+tola %error<(norm(x0)+1)*tol,
82     flag=0; %salida
83 end
84
85 psi=psinew;T=Tnew;C=Cnew;
86 iter=iter+1;
87 end
88 psi=psinew;T=Tnew;C=Cnew;
89 %% Gr\ 'afica de los Resultados
90 %run GuardaFig.m
91
92 % [pol]=subfigura(L,H,theta,p,t,psi,T,C,NN,pp,qq); %plot para mallas con estructura
93 [pol]=trisubfigura(L,H,theta,p,t,psi,T,C,pp,qq); %plot para mallas sin estructura

```

Anexo IX: Código *GenM3.m*

Código que genera automáticamente mallas con los parámetros geométricos del dominio en forma de paralelogramo, estos parámetros son: altura H , longitud L y ángulo de inclinación θ .

```

1  function [pp,t,b,bn,B]=GenM3(L,theta,H,NN)
2  xmax=L*cos(theta);
3  if sin(theta)>0
4      ymax=L*sin(theta)+H;
5  else
6      ymax=L*sin(theta);
7  end
8  s=tan(theta);
9  [x,y]=meshgrid(linspace(0,xmax,NN));
10 y=y*H/xmax+x*s;
11 t=delaunay(x,y);
12 p=[x(:) y(:)];
13 N=size(p,1);
14 r=H/NN*0.9;
15
16 %% Frontera Neumman
17 bn=zeros(N,1);
18 for i=1:N
19     if p(i,2)<s*p(i,1)+r && p(i,2)>s*p(i,1)-r
20         bn(i)=i;
21     elseif p(i,2)<s*p(i,1)+r+H && p(i,2)>s*p(i,1)-r+H
22         bn(i)=i;
23     end
24 end
25 bn=setdiff(bn,0);
26
27
28 %% Frontera Dirichlet
29 b=zeros(N,1);
30 for i=1:N
31     if p(i,1)==0 || p(i,1)==xmax
32         b(i)=i;
33     end
34 end
35 b=setdiff(b,0);
36 bn=setdiff(bn,b);
37 B=[b;bn];
38
39 %% Coordenadas base
40 xbn=p(bn,1);
41 xb=p(b,1);
42 ybn=p(bn,2);
43 yb=p(b,2);

```

```

44 p=[xbn ybn ; xb yb];
45 bn=[1:size(xbn,1)];
46 b=[size(xbn,1)+1:size(xbn,1)+size(xb,1)];
47 B=[bn,b];
48
49 %% Nodos aleatorios
50 Nal = NN*NN-4*NN+4; %N\ 'umero de nodos aleatorios
51 pp = zeros(Nal,2);
52 i = size(B,2);
53
54 if sin(theta)>0
55     while (i < Nal + size(B,2))
56         i = i + 1;
57         x_i = xmax*rand; % genera un n\ 'umero entre 0 y xmax
58         y_i = ymax*rand; % genera un n\ 'umero entre 0 y ymax
59         if (y_i<s*x_i) || (y_i>s*x_i+H) % si los puntos estan fuera del paralelogramo
60             i = i - 1; % decrece el n\ 'umero de la iteraci\ 'on para intentarlo de nue
61         else % si el punto est\ 'a dentro del paralelogramo
62             pp(i,:) = [x_i y_i]; % adiciona los puntos a la lista
63         end
64     end
65 else
66     while (i < Nal + size(B,2))
67         i = i + 1;
68         x_i = xmax*rand; % genera un n\ 'umero entre 0 y xmax
69         y_i = H+(ymax-H)*rand; % genera un n\ 'umero entre 0 y ymax
70         if (y_i>s*x_i+H) || (y_i<s*x_i) % si los puntos est\ 'an fuera del paralelogramo
71             i = i - 1; % decrece el n\ 'umero de la iteraci\ 'on para intentarlo de nue
72         else % si el punto est\ 'a dentro del paralelogramo
73             pp(i,:) = [x_i y_i]; % adiciona los puntos a la lista
74         end
75     end
76 end
77
78 pp(1:size(B,2),:)=p;
79
80 %%
81 t=delaunay(pp(:,1),pp(:,2));
82 triplot(t,pp(:,1),pp(:,2),'color','b')
83 hold on
84 plot(pp(b,1),pp(b,2),'*r')
85 plot(pp(bn,1),pp(bn,2),'og')
86 axis equal; grid on
87 end

```

Anexo X: Código *matrizd_4.m*

Modificación del código *matrizd_3.m* para que pueda funcionar con las mallas automáticas.

```

1 function [K,F]=matrizd_4(p,t,bn,f,g,L,VN,theta,H,d)
2 %diferencias dfg
3 N=size(p,1);
4 K=sparse(N,N);
5 F=sparse(N,1);
6 FF=L(6);
7 L=L(1:5)';
8
9 for i=1:N
10     I=any(ismember(t,i),2); %% buscamos los indices donde anda i
11     tt=t(I,:); tt=tt(:);
12     I2=setdiff(tt,i); %%Estos son los vecinos de i
13     u=p(I2,1)'; v=p(I2,2)';
14
15     if ismember(i,bn)
16         u=[u 2*p(i,1)-mean(u)];
17         v=[v 2*p(i,2)-mean(v)];
18     end
19
20         dx=u-p(i,1); dy=v-p(i,2);
21     M=[dx; dy; dx.*dx; dx.*dy; dy.*dy];
22     Gamma=pinv(M)*L;
23     Gamma=[FF-sum(Gamma); Gamma];
24     F(i)=F(i)+f(p(i,:));
25
26     if ismember(i,bn)
27         GammaN=pinv(M)*[VN(p(i,:),theta,H,d); zeros(3,1)];
28         GammaN=[-sum(GammaN); GammaN];
29         F(i)=F(i)-g(p(i,:))/GammaN(end)*Gamma(end);
30         GammaN=-GammaN(1:(end-1))/GammaN(end)*Gamma(end);
31         Gamma=Gamma(1:(end-1))+GammaN;
32     end
33
34     K(i,[i;I2])=Gamma;
35     Gamma=[];
36 end

```

Anexo XI: Código *robngfd_conv3.m*

Modificación del código *robngfd_conv3.m* para que pueda funcionar con las mallas automáticas.

```

1 function [U,K,F,NB]=robngfd_conv3(p,b,bn,t,f,g,dn,DX,DY,VN,theta,H,NN)
2 % resuelve la eciaci'on gobernante 2 (Temperatura) o 3 (Concentraci'on)

```

```

3  %diferencias fem a ver que pasa
4  N=size(p,1);T=size(t,1); %n|'umero de nodos, n|'umero de tri|'angulos
5  K=sparse(N,N);F=sparse(N,1);
6  %Ahora necesitamos una estructura de datos poderosa
7  nb=setdiff(1:N,b); %Estos no son frontera
8
9  L=L(:);%DN=DN(:);
10
11 for i=nb
12     I=any(ismember(t,i),2); %% buscamos los indices donde anda i
13     tt=t(I,:);tt=tt(:);
14     I2=setdiff(tt,i); %%Estos son los vecinos de i
15     %NB(nb)=length(I2);
16     u=p(I2,1)';v=p(I2,2)';
17
18     if ismember(i,bn)
19         u=[u 2*p(i,1)-mean(u)];v=[v 2*p(i,2)-mean(v)];
20     end
21
22     dx=u-p(i,1);dy=v-p(i,2);
23
24     M=[dx;dy;dx.*dx;dx.*dy;dy.*dy];
25
26     Gamma=pinv(M)*[DX(i) DY(i) 2 0 2]'; %resuelve (M,L);
27     Gamma=[-sum(Gamma);Gamma];
28
29     F(i)=F(i)+f(p(i,:));
30
31     if ismember(i,bn)
32         GammaN=pinv(M)*[VN(p(i,:),theta,H,NN);zeros(3,1)]; %resuelve (M,[VN(p(i
33         GammaN=[-sum(GammaN);GammaN];
34         %F(i)=F(i)+dn(p(i,))/GammaN(end); 270919
35         F(i)=F(i)+Gamma(end)*dn(p(i,))/GammaN(end);
36         GammaN=-GammaN(1:(end-1))/GammaN(end)*Gamma(end);
37         Gamma=Gamma(1:(end-1))+GammaN;
38     end
39
40     K(i,[i;I2])=Gamma;
41
42     Gamma=[];
43 end
44
45 % Ahora de a|~naden las condiciones de Dirichlet en K and F
46 F=F-sum(kron(g(p(b,:))',ones(N,1)).*K(:,b),2); %Valores que pasan al lado derecho
47 F(b)=[g(p(b,:))]; %Valores en la frontera
48 K(b,:)=0; K(:,b)=0; %Ponemos ceros para no modificar los valores en la frontera
49 K(b,b)=speye(length(b),length(b)); % se pone I en la submatriz "frontera" de K
50

```

```

51 % Ahora resolvemos
52 U=K\F; % La aproximación es  $U_1 \phi_1 + \dots + U_N \phi_N$ 

```

Anexo XII: Código *VN4.m*

Función que obtiene los vectores normales a las fronteras.

```

1 function vn=VN4(p,theta,H,NN)
2 s=tan(theta);
3 alpha=theta-pi/2;
4 x=p(:,1);y=p(:,2);
5 r=H/NN*0.9;
6
7 if y<s*x+r && y>s*x-r
8     vn=[cos(alpha);sin(alpha)];
9 elseif y<s*x+r+H && y>s*x-r+H
10    vn=[cos(alpha+pi);sin(alpha+pi)];
11 % elseif x==0
12 %     vn=[-1,0];
13 % elseif x==1.5
14 %     vn=[1,0];
15 else
16     vn=[0;0];
17 end
18 end

```

Anexo XIII: Código *trisubfigura.m*

Función que grafica los resultados obtenidos.

```

1 function [pol]=trisubfigura(L,H,theta,p,t,psi,T,C,pp,qq)
2
3 xmax=L*cos(theta);
4 if sin(theta)>0
5     ymax=L*sin(theta)+H; ymin=0;
6     pol=[0 xmax xmax 0 0 ;0 ymax-H ymax H 0]';
7 else
8     ymax=H; ymin=L*sin(theta);
9     pol=[0 xmax xmax 0 0 ;0 ymin ymin+H H 0]';
10 end
11
12 figure('units','normalized','outerposition',[0 0 1 1])
13 subplot(233)
14 trisurf(t,p(:,1),p(:,2),full(psi));view(2);shading interp;colorbar
15 axis([0 xmax ymin ymax]); axis equal;
16 subplot(231)

```

```

17 trisurf(t,p(:,1),p(:,2),full(T));view(2);shading interp;colorbar
18 axis([0 xmax ymin ymax]); axis equal;
19 subplot(232)
20 trisurf(t,p(:,1),p(:,2),full(C));view(2);shading interp;colorbar
21 axis([0 xmax ymin ymax]); axis equal;
22
23 subplot(236)
24 plot(pol(:,1),pol(:,2),'color',[0.6 0.6 0.6]);hold on
25 psimax=max(psi); psimin=min(psi); psih=(psimax-psimin)/10;
26 tricontour(t,p(:,1),p(:,2),full(psi),[psimin:psih:psimax]); colorbar
27 axis([0 xmax ymin ymax]); axis equal;
28 subplot(234)
29 plot(pol(:,1),pol(:,2),'color',[0.6 0.6 0.6]);hold on
30 Tmax=max(T); Tmin=min(T); Th=(Tmax-Tmin)/10;
31 tricontour(t,p(:,1),p(:,2),full(T),[Tmin:Th:Tmax]); colorbar
32 axis([0 xmax ymin ymax]); axis equal;
33 subplot(235)
34 plot(pol(:,1),pol(:,2),'color',[0.6 0.6 0.6]);hold on
35 Cmax=max(C); Cmin=min(C); Ch=(Cmax-Cmin)/10;
36 if Ch~=0
37 tricontour(t,p(:,1),p(:,2),full(C),[Cmin:Ch:Cmax]); colorbar
38 end
39 axis([0 xmax ymin ymax]); axis equal;
40
41
42 figure
43 plot(pol(:,1),pol(:,2),'color',[0.6 0.6 0.6]);hold on
44 quiver(p(:,1),p(:,2),pp,qq,'color','b')
45 axis([0 xmax ymin ymax]); axis equal;
46 end

```

Anexo XIV: Ejemplos para comparación

Este código se usó para todos los ejemplos que nos sirvieron para comparación con los artículos de referencia. Solamente cambiando los parámetros geométricos y la malla de nodos.

```

1 %% Ejemplo 10
2 clear; clc; close all;
3 %% Par \ 'ametros
4 Ra=100;N=5;Le=0.8;
5 H=0.8; % Altura del paralelogr \ 'amo
6 HsobreL=0.5; % Proporción H/L
7 L=H/HsobreL; % Longitud del paralelogr \ 'amo
8 theta=30; % \ 'Angulo en grados del paralelogr \ 'amo
9 theta=theta*pi/180; % \ 'Angulo en radianes
10 %% Malla
11 delta=0.01; % barra de error para detectar nodos frontera

```

```

12 load MallaEj10.mat %p=coordenadas , t=tri | 'angulos
13 [b,bn,B]=fronteras(L,theta,H,p,t,delta);
14 %% Tolerancias y errores
15 tola=1e-2;tolr=1e-4; %@tola=0.01;tolr=0.0001;
16 err=[];
17 flag=1;iter=1;itermax=50; %@teraci | 'on m | 'axima
18
19 %% Funciones
20 finicial=@(x) 2/3*Ra+2/3*N*Le; % tercer t | 'ermino de la ec. gobernante 1
21 cero=@(x) zeros(size(x,1),1);
22 g2=@(x) 1-x(:,1)*2/3; % ecuaci | 'on del plano de aproximaci | 'on T y C
23 % g2=@(x) x(:,1)*2/3; % ecuaci | 'on del plano de flujo invertido
24 % z=g2(p);
25 % trimesh(t,p(:,1),p(:,2),z); colorbar, view(2) % gr | 'afica del plano de aproxim
26 %% Primera aproximaci | 'on
27 % Funci | 'on de flujo
28 psi=dirichgfd(p,B,t,finicial,cero,1,0,1,0,0,0); % primera aproximaci | 'on a psi
29 % plano % ^ lado derecho
30 % Velocidades de flujo
31 [~,DX,~]=dirichgfd(p,B,t,cero,cero,0,0,0,1,0,0);
32 [~,DY,~]=dirichgfd(p,B,t,cero,cero,0,0,0,0,1,0);
33 pp=DY*full(psi);
34 qq=DX*full(psi);
35 % Aprximaci | 'on de las funciones Temperatura y Concentraci | 'on
36 T=robingfd_conv3b(p,b,bn,t,cero,g2,@dn,-pp,-qq,@VN4b,theta,H,delta);
37 C=robingfd_conv3b(p,b,bn,t,cero,g2,@dn,-Le*pp,-Le*qq,@VN4b,theta,H,delta);
38
39 %% Corrector Predictor
40 while (flag)&&(iter<itermax)
41 DX=matrizd_4b(p,t,[],cero,cero,[1 0 0 0 0 0],@VN4b,theta,H,delta); %Derivada en x para
42 DY=matrizd_4b(p,t,[],cero,cero,[0 1 0 0 0 0],@VN4b,theta,H,delta); %Derivada en y para
43
44 DXT=DX*(T);
45 DXC=DX*(C);
46 f=-(DXT*Ra+DXC*N*Le);
47 psinew=dirichgfd(p,B,t,f,cero,1,0,1,0,0,0);
48 [~,DX,~]=dirichgfd(p,B,t,cero,cero,0,0,0,1,0,0); %Derivada en x para condiciones de D
49 [~,DY,~]=dirichgfd(p,B,t,cero,cero,0,0,0,0,1,0); %Derivada en y para condiciones de D
50 pp=DY*full(psinew);
51 qq=-DX*full(psinew);
52 Tnew=robingfd_conv3b(p,b,bn,t,cero,g2,@dn,-pp,-qq,@VN4b,theta,H,delta);
53 Cnew=robingfd_conv3b(p,b,bn,t,cero,g2,@dn,-Le*pp,-Le*qq,@VN4b,theta,H,delta);
54
55 error=norm(psi-psinew)+norm(T-Tnew)+norm(C-Cnew);
56
57 display([' iter=' num2str(iter) ' _error=' num2str(error)])
58
59 r0=norm(psi);

```

```

60 if error<tolr*r0+tola %error<(norm(x0)+1)*tol ,
61     flag=0; %salida
62 end
63
64 psi=psinew ;T=Tnew ;C=Cnew ;
65 iter=iter+1;
66 end
67 psi=psinew ;T=Tnew ;C=Cnew ;
68
69 %% Gr\ 'afica de los Resultados
70 [pol]=trisubfigura (L,H,theta ,p,t ,psi ,T,C,pp,qq); %plot para mallas sin estructura

```

Anexo XV: Código *fronteras.m*

Esta función define automáticamente los nodos de frontera de Dirichlet y los nodos de frontera de Neumman, a partir de los datos geométricos.

```

1 function [b,bn,B]=fronteras (L,theta ,H,p,t ,d)
2 %B = nodos de frontera
3 %b = nodos frontera con condiciones de Dirichlet
4 %bn = nodos frontera con condiciones de Neumman
5 %p = coordenadas x y de los nodos
6 %t = tri \ 'angulos
7
8 xmax=L*cos (theta );
9 s=tan (theta );
10 N=size (p,1);
11
12 %% Frontera Neumman
13 bn=zeros (N,1);
14 for i=1:N
15     if p(i,2)<s*p(i,1)+d && p(i,2)>s*p(i,1)-d
16         bn(i)=i;
17     elseif p(i,2)<s*p(i,1)+d+H && p(i,2)>s*p(i,1)-d+H
18         bn(i)=i;
19     end
20 end
21 bn=setdiff (bn,0);
22
23
24 %% Frontera Dirichlet
25 b=zeros (N,1);
26 for i=1:N
27     if p(i,1)==0 || p(i,1)>xmax-d
28         b(i)=i;
29     end
30 end

```

```

31 b=setdiff(b,0);
32 bn=setdiff(bn,b);
33 B=[b;bn];
34
35 triplot(t,p(:,1),p(:,2),'color','b')
36 hold on
37 plot(p(b,1),p(b,2),'*r')
38 plot(p(bn,1),p(bn,2),'og')
39 axis equal; grid on
40 end

```

Anexo XVI: Código *robinfo_ conv3b.m*

Código modificado de la función *robinfo_ conv3.m* para poder funcionar con la función *VN4b.m*.

```

1 function [U,K,F,NB]=robinfo_ conv3b(p,b,bn,t,f,g,dn,DX,DY,VN,theta,H,d)
2 % resuelve la ecuación de gobierno 2 (Temperatura) o 3 (Concentración)
3 % diferencias finitas a ver que pasa
4 N=size(p,1);T=size(t,1); % n número de nodos, n número de triángulos
5 K=sparse(N,N);F=sparse(N,1);
6 % Ahora necesitamos una estructura de datos poderosa
7 nb=setdiff(1:N,b); % Estos no son frontera
8
9 L=L(:);DN=DN(:);
10
11 for i=nb
12     I=any(ismember(t,i),2); %% buscamos los índices donde anda i
13     tt=t(I,:);tt=tt(:);
14     I2=setdiff(tt,i); %% Estos son los vecinos de i
15     NB(nb)=length(I2);
16     u=p(I2,1);v=p(I2,2);
17
18     if ismember(i,bn)
19         u=[u-2*p(i,1)-mean(u)];v=[v-2*p(i,2)-mean(v)];
20     end
21
22     dx=u-p(i,1);dy=v-p(i,2);
23
24     M=[dx;dy;dx.*dx;dx.*dy;dy.*dy];
25
26     Gamma=pinv(M)*[DX(i)DY(i)202]'; % resuelve (M,L);
27     Gamma=[-sum(Gamma);Gamma];
28
29     F(i)=F(i)+f(p(i,:));
30
31     if ismember(i,bn)
32         GammaN=pinv(M)*[VN(p(i,:),theta,H,d);zeros(3,1)]; % resuelve (M,[VN(p(i,

```

```

33     GammaN=[-sum(GammaN);GammaN];
34      $\mathcal{F}(i)=F(i)+dn(p(i,:),:)/GammaN(end);$  270919
35     F(i)=F(i)+Gamma(end)*dn(p(i,:),:)/GammaN(end);
36     GammaN=-GammaN(1:(end-1))/GammaN(end)*Gamma(end);
37     Gamma=Gamma(1:(end-1))+GammaN;
38     end
39
40     K(i,[i;I2])=Gamma;
41
42     Gamma=[];
43 end
44
45 % Ahora de a~naden las condiciones de Dirichlet en K and F
46 F=F-sum(kron(g(p(b,:))',ones(N,1)).*K(:,b),2); %Valores que pasan al lado derecho
47 F(b)=[g(p(b,:))]; %Valores en la frontera
48 K(b,:)=0; K(:,b)=0; %Ponemos ceros para no modificar los valores en la frontera
49 K(b,b)=speye(length(b),length(b)); % se pone I en la submatriz "frontera" de K
50
51 % Ahora resolvemos
52 U=K\F; % La aproximaci~on es  $U_1 \phi_1 + \dots + U_N \phi_N$ 

```

Anexo XVII: Código *dn.m*

Función que separa las coordenadas del vector p en vectores separados x, y .

```

1 function dn=dn(P)
2 x=P(:,1);y=P(:,2);
3 dn=0;

```

Anexo XVIII: Código *VN4b.m*

Función modificada del código *VN4.m*, para funcionar con un parámetro δ , el cual es una barra de error para detectar alrededor de las fronteras, mientras que *VN4.m* usa el la cantidad de nodos en la malla para definir este parámetro.

```

1 function vn=VN4b(p,theta,H,d)
2 s=tan(theta);
3 alpha=theta-pi/2;
4 x=p(:,1);y=p(:,2);
5
6 if y<s*x+d && y>s*x-d
7     vn=[cos(alpha);sin(alpha)];
8 elseif y<s*x+d+H && y>s*x-d+H
9     vn=[cos(alpha+pi);sin(alpha+pi)];
10 % elseif x==0
11 %     vn=[-1,0];

```

```

12 % elseif x==1.5
13 %     vn=[1,0];
14 else
15     vn=[0;0];
16 end
17 end

```

Anexo XIX: Código *matrizd_4b.m*

Modificación de la función *matrizd_4.m* para poder funcionar con la función *VN4b.m*.

```

1 function [K,F]=matrizd_4b(p,t,bn,f,g,L,VN,theta,H,d)
2 %diferencias dfg
3 N=size(p,1);
4 K=sparse(N,N);
5 F=sparse(N,1);
6 FF=L(6);
7 L=L(1:5)';
8
9 for i=1:N
10     I=any(ismember(t,i),2); %% buscamos los indices donde anda i
11     tt=t(I,:); tt=tt(:);
12     I2=setdiff(tt,i); %%Estos son los vecinos de i
13     u=p(I2,1)'; v=p(I2,2)';
14
15     if ismember(i,bn)
16         u=[u 2*p(i,1)-mean(u)];
17         v=[v 2*p(i,2)-mean(v)];
18     end
19
20         dx=u-p(i,1); dy=v-p(i,2);
21     M=[dx;dy;dx.*dx;dx.*dy;dy.*dy];
22     Gamma=pinv(M)*L;
23     Gamma=[FF-sum(Gamma);Gamma];
24     F(i)=F(i)+f(p(i,:));
25
26     if ismember(i,bn)
27         GammaN=pinv(M)*[VN(p(i,:),theta,H,d);zeros(3,1)];
28         GammaN=[-sum(GammaN);GammaN];
29         F(i)=F(i)-g(p(i,:))/GammaN(end)*Gamma(end);
30         GammaN=-GammaN(1:(end-1))/GammaN(end)*Gamma(end);
31         Gamma=Gamma(1:(end-1))+GammaN;
32     end
33
34     K(i,[i;I2])=Gamma;
35     Gamma=[];
36 end

```

BIBLIOGRAFÍA

- [Benito et al., 2007] Benito, J., Urena, F., and Gavete, L. (2007). Solving parabolic and hyperbolic equations by the generalized finite difference method. *Journal of computational and applied mathematics*, 209(2):208–233.
- [Bourich et al., 2004] Bourich, M., Hasnaoui, M., and Amahmid, A. (2004). Double-diffusive natural convection in a porous enclosure partially heated from below and differentially salted. *International Journal of Heat and fluid flow*, 25(6):1034–1046.
- [Chávez-Negrete et al., 2018] Chávez-Negrete, C., Domínguez-Mota, F., and Santana-Quinteros, D. (2018). Numerical solution of richard’s equation of water flow by generalized finite differences. *Computers and Geotechnics*, 101:168–175.
- [Costa, 2004] Costa, V. (2004). Double-diffusive natural convection in parallelogrammic enclosures filled with fluid-saturated porous media. *International journal of heat and mass transfer*, 47(12-13):2699–2714.
- [Cushman-Roisin and Beckers, 2011] Cushman-Roisin, B. and Beckers, J.-M. (2011). *Introduction to geophysical fluid dynamics: physical and numerical aspects*. Academic press.
- [Domínguez-Mota et al., 2014] Domínguez-Mota, F. J., Mendoza-Armenta, S., Tinoco-Guerrero, G., and Tinoco-Ruiz, J.-G. (2014). Finite difference schemes satisfying an optimality condition for the unsteady heat equation. *Mathematics and Computers in Simulation*, 106:76–83.
- [Fan et al., 2013] Fan, C.-M., Chien, C.-S., Chan, H.-F., and Chiu, C.-L. (2013). The local rbf collocation method for solving the double-diffusive natural convection in fluid-saturated porous media. *International Journal of Heat and Mass Transfer*, 57(2):500–503.
- [Gaul et al., 2013] Gaul, L., Kögl, M., and Wagner, M. (2013). *Boundary element methods for engineers and scientists: an introductory course with advanced topics*. Springer Science & Business Media.
- [Gockenbach, 2006] Gockenbach, M. S. (2006). *Understanding and implementing the finite element method*, volume 97. Siam.
- [Hutter, 2004] Hutter, K. (2004). *Klaus Jöhnk Continuum Methods of physical Modeling*. Springer-Verlag Berlin Heidelber New York.
- [Kramer et al., 2007] Kramer, J., Jecl, R., and Škerget, L. (2007). Boundary domain integral method for the study of double diffusive natural convection in porous media. *Engineering analysis with boundary elements*, 31(11):897–905.
- [Li et al., 2018] Li, P.-W., Chen, W., Fu, Z.-J., and Fan, C.-M. (2018). Generalized finite difference method for solving the double-diffusive natural convection in fluid-saturated porous media. *Engineering Analysis with Boundary Elements*, 95:175–186.

-
- [López-Garza and Martínez-Ortiz, 2020] López-Garza, G. and Martínez-Ortiz, F. H. (2013 (acceso 14-Enero-2020)). Ecuaciones diferenciales parciales. http://sgpwe.izt.uam.mx/files/users/uami/gabl/ecuaciones_Diferenciales_parciales.pdf.
- [Na, 1980] Na, T. Y. (1980). *Computational methods in engineering boundary value problems*. Academic press.
- [Nield et al., 2006] Nield, D. A., Bejan, A., et al. (2006). *Convection in porous media*, volume 3. Springer.
- [Olmo and Nave, 2020] Olmo, M. and Nave, R. (2010 (acceso 26-Marzo-2020)). Series de taylor. <http://hyperphysics.phy-astr.gsu.edu/hbasees/tayser.html>.
- [Rice University - Houston - Texas et al., 2020] Rice University - Houston - Texas, Hoy, W., and Roos, T. (2005 (acceso 01-Abril-2020)). Grashof numbers: A helpful way to look at convective effects. <http://www.owlnet.rice.edu/~ceng402/proj05/hoy/hoyroos/background/background.html>.
- [The Concord Consortium and Xie, 2020] The Concord Consortium and Xie, C. (2010 (acceso 30-Marzo-2020)). The prandtl number: Conductive vs. convective heat transfer. <https://energy.concord.org/energy2d/prandtl.html>.
- [Universidad Abierta y a Distancia de México et al., 2020] Universidad Abierta y a Distancia de México, García-Jarillo, V. H., and Vargas-Rocha, L. M. (2014 (acceso 13-Marzo-2020)). Definición de las ecuaciones diferenciales. https://www.unadmexico.mx/sitios/aplicaciones-107/LITE_36/_Un_130_EcuacionesDiferencialesDePrimerOrdenI/escenas/2_Inicio_1.html.
- [Universidad de Salamanca (España) and Sánchez-San-Román, 2020] Universidad de Salamanca (España) and Sánchez-San-Román, J. (Ene-2008 (acceso 28-Marzo-2020)). Flujo en medios porosos: Ley de darcy. <http://agronomia.criba.edu.ar/carreras/ia/archivos/Materias/579/archivos/aguassubterraneeas/DARCY.pdf>.
- [Universidad Nacional de Educación a Distancia, Madrid et al., 2020] Universidad Nacional de Educación a Distancia, Madrid, Hernández-Rodríguez, J., and ETSII (2000, 2001 (acceso 28-Marzo-2020)). Flujos bidimensionales. concepto de función de corriente. <https://www2.uned.es/ing-fluidos/IntroMF/node54.html>.