

**Centro de Investigación Científica y de Educación  
Superior de Ensenada, Baja California**



---

**Programa de Posgrado en Ciencias de la Tierra  
con orientación en Geofísica Aplicada**

---

**Estudio y simulación numérica para el modelado  
de yacimientos geotérmicos**

Tesis

para cubrir parcialmente los requisitos necesarios para obtener el grado de  
**Maestro** en Ciencias

Presenta:

**Fernando José Bello Fuentes**

Ensenada, Baja California, México  
2016

Tesis defendida por

**Fernando José Bello Fuentes**

y aprobada por el siguiente Comité

---

Dr. Rogelio Vázquez González  
Director de tesis

Dra. Olga Sarychikhina

Dr. Enrique Gómez Treviño

Dr. Israel Marck Martínez Pérez

Dr. Jorge Ramírez Hernández



---

**Dr. Juan García Abdeslem**  
Coordinador del Posgrado en  
Ciencias de la Tierra

---

**Dra. Rufina Hernández Martínez**  
Directora de Estudios de Posgrado

Resumen de la tesis que presenta **Fernando José Bello Fuentes** como requisito parcial para la obtención del grado de **Maestro** en Ciencias en Ciencias de la Tierra **con orientación en Geofísica Aplicada**.

### **Estudio y simulación numérica para el modelado de yacimientos geotérmicos**

Resumen aprobado por:

---

Dr. Rogelio Vázquez González  
Director de tesis

La energía geotérmica es la energía proveniente del interior de la Tierra y está disponible para aprovecharla mediante diversos dispositivos tecnológicos. Para poder hacerlo de forma correcta, es necesario estudiar el medio geológico que contiene el recurso y los fenómenos físicos que determinan la interacción de las componentes del sistema geotérmico. Dada la complejidad de los modelos matemáticos que representan un yacimiento geotérmico, frecuentemente, dichos modelos se resuelven utilizando simulación numérica, para así modelar el comportamiento del sistema geotérmico. Conocer el sistema físico que se desea representar siempre es el primer paso para poder modelarlo. El objetivo de la simulación de yacimientos geotérmicos es poder predecir el comportamiento de los mismos y poder aprovechar al máximo este recurso energético.

El proyecto de tesis de maestría consiste en estudiar los fenómenos físicos en los yacimientos geotérmicos, las ecuaciones matemáticas que representan dichos fenómenos y las soluciones numéricas que se han desarrollado para la simulación de su comportamiento. La solución del problema parte de las ecuaciones de conservación de la masa, de la energía y las de momento, considerando un fluido como agua pura en estado líquido, ligeramente compresible y viscosa que fluye a través de un medio poroso y permeable con propiedades isotrópicas constantes en el tiempo. El objetivo de la solución es calcular los campos de presiones y temperaturas y mediante el uso de ecuaciones de estado, calcular las propiedades del agua, y así desarrollar un software de simulación en dos dimensiones para estimar el estado estacionario y transitorio de los yacimientos geotérmicos basados en el esquema de discretización de volumen finito.

El sistema de ecuaciones a resolver es no lineal y las ecuaciones que las constituyen están intrínsecamente acopladas debido a que la velocidad del fluido aparece en cada ecuación de momento y en la de continuidad por lo que se resuelve iterativamente. Primero se estiman los campos de presiones y velocidades, luego se acoplan con la ecuación de conservación de energía, siguiendo el procedimiento del algoritmo iterativo SIMPLE, propuesto por Patankar y Spalding en 1972.

Los datos necesarios para las simulaciones se generaron sintéticamente siguiendo las especificaciones del modelo de caja de M.L. Sorey (1978). Los resultados obtenidos en este trabajo se compararon con los del M.L Sorey y a su vez permitieron llegar a las siguientes conclusiones: La roca funciona como aislante térmico, en los procesos geotérmicos el fenómeno que transporta la temperatura es la convección y ésta es afectada por la viscosidad del fluido y las propiedades del medio. El factor que determina que tan importante es la convección en relación a la difusión es el número de Peclet, para valores mayores a 2.0 la convección es dominante, y en consecuencia las ecuaciones se vuelven altamente inestables. Para resolver problemas de inestabilidad se propusieron otros esquemas de discretización a parte del esquema de diferencias centrales, como el de diferencias hacia adelante “upwind”, de potencia, y exponencial.

**Palabras clave:** yacimientos geotérmicos, ecuaciones de transporte, dinámica computacional de fluidos, energía, entalpía, convergencia, estabilidad

Abstract of the thesis presented **by Fernando José Bello Fuentes** as a partial requirement to obtain the Master of Science degree in Earth's Sciences with orientation in Applied Geophysics.

### **Study and numerical simulation for the modeling of geothermal reservoirs**

Abstract approved by:

---

Dr. Rogelio Vázquez González  
Director thesis

Geothermal energy is the energy coming from the inner of the Earth and it is available to use it by diverse technologic devices. In order to do so in the right way, it is necessary to study the geological medium which contains the resource and the physical phenomena that determine the interaction of the components of the geothermical system. Given the complexity of the mathematical models that represents a geothermal reservoir, they are usually solved using numerical simulation for the geothermal system's behavior modeling. Knowing the physical system that wants to be represented is always the first step for modeling it. The goal of the geothermal reservoir simulation is to foresee the behavior of them and be able to seize to the maximum this energetic resource.

This master's thesis project consist to study the physical phenomena in the geothermal reservoirs, the mathematical equations which represent those phenomena and the numerical solutions that have been developed for their behavior simulation. The solution of the problem starts with the mass and energy conservation laws and the momentum equation, considering the fluid as pure water in liquid state, lightly compressible and viscous that flows through a porous and permeable media with constants isotropic properties through time. The goal of the solution is to compute the pressure and temperature fields, and by using the state equations, calculate water properties and then to develop two dimension simulation software to estimate the stationary and transitory state of the geothermal reservoir based on the finite volume discretization method.

The equation system to solve is non linear and the equations that constitute it are intrinsically coupled due to the fluid velocity which appears in every momentum equation and in the continuity equation so it is solved iteratively. First it the pressure and velocity fields are guessed, then they are coupled with the energy conservation following the procedure of the SIMPLE algorithm, proposed by Patankar and Spalding in 1972.

The necessary data for the simulation were generated synthetically following the M.L. Sorey's (1978) box model specifications. The results achieved in this thesis were compared with the obtained by M.L. Sorey and they also allowed reach the following

conclusions: The rock works as a thermal insulator, in the geothermic processes the phenomenon which transports the temperature is convection y this is affected by the fluid's viscosity and the medium properties. The factor that determines how important is the convection in relation to diffusion is the Peclet's number, for Peclet's numbers higher than 2.0 the convection is dominant, and in consequence the equations turns highly unstable. To solve the problems of instability, other discretization schemes were proposed beside of the central differential scheme, as the upwind, power law and exponential.

**Keywords:** Geothermal reservoirs, transport equations, computational fluids dynamics, energy, enthalpy, convergence, stability.

## **Dedicatoria**

**A mi madre, por su incansable labor, por ser mi guía, mi amiga, mi consejera. Bien dice el dicho, "madre solo hay una", y que fortuna tengo al tenerte a ti. Este logro es la materialización de todos tus años de esfuerzos, desveladas cuando estuve enfermo y regaños para que dejara de jugar y me pusiera a estudiar. Al principio no entendía y me quejaba, pero hoy te lo agradezco.**

**A mi padre, todos tus consejos y enseñanzas siguen en mi, y procuro seguir tus pasos y compartirlos con el resto del mundo.**

**A mis hermanos, todos Uds. han sido un ejemplo para mí. Desde pequeño solo he podido admirarlos, los amo. Rafael con su rectitud y paciencia, Micky con ese gran amor y compasión, Carmen con su alegría y locuras, Linda con su sonrisa incansable y hambre de aventura, Isabel con su dedicación a hacer lo que ama sin doblegarse ante ninguna adversidad, por último la menor de todos Ynejosé, con sus sueños, imaginación y dulzura. Ustedes me enseñaron a predicar con el ejemplo.**

## **Agradecimientos**

**A CICESE y Conacyt por las facilidades otorgadas para la realización de este proyecto.**

**A todo el personal de Biblioteca de CICESE y de la UABC, por la paciencia y la ayuda brindada en la búsqueda del material bibliográfico.**

**A los miembros de comité doctores, Enrique Gómez Treviño, Olga Sarychikhina, Israel Martínez y Jorge Ramírez por el apoyo, correcciones, y enseñanzas brindadas durante la realización de este proyecto.**

**A mi director de tesis Dr. Rogelio Vázquez González, por toda la paciencia por explicarme todas las veces que fueron necesarias las mismas cosas hasta que me quedaran claras. Por su confianza y palabras de aliento durante mis momentos difíciles y en especial por brindarme más que asesorías académicas, una amistad.**

**Al Dr. Efraín Gómez Arias, por su gran disposición en todo momento, y quien fungió como si fuese miembro de mi comité de tesis, pues me apoyó mucho en la realización de ésta.**

**A todos los profesores y personal administrativo, en especial a Ana Rosa Soto, José Frez y Jonás De Basabe, que siempre me escucharon cuando les pedí un minuto de su tiempo.**

**A los profesores Miguel Castillejo, Fanny Sanchez de la Universidad Central de Venezuela, quienes siempre me dieron aliento y me motivaron a superarme cada día más.**

**A Ana Karen Loera, por su compañía, apoyo, aliento, paciencia y cariño durante todo este tiempo.**

**A la familia la familia López Agüero, mamá Maritza, Katia, Karla e Isaac, quienes me recibieron y acogieron como un miembro más de su familia y me brindaron su cariño incondicional desde mi llegada a México.**

**A Miguel Oliver Ocaño, Agustín Mastache Román, y Julio González, por todos sus consejos, gran amistad, por escucharme durante mis momentos difíciles, por ofrecerme aliento y por ser una influencia positiva en mi vida.**

**A Rogelio Arce, por sus consejos personales y académicos, lo que empezó como rivalidad, terminó como una gran amistad.**

**A Ismael Yarbuh y a Karla Lara, por sus consejos en lo personal y académico, y por enseñarme a surfear.**

**A Marlyne Jaimes, por estos casi 14 años de amistad llenos de apoyo e infinitos gestos de cariño.**

**A Valeria Reyes y a Cécile Blanch Fanny Laurence Savry, Uds. se convirtieron en mis mejores amigas. Siempre contarán conmigo.**

**A todos mis compañeros y a las personas que he conocido en CICESE, en especial a Mayra Cuellar, Claudia Quinteros, Yéssica Contreras, Adrián León, Andrea Lotero, Armando Calderón, Daniela Tazzo, Karen Legorreta, Valdemar Belmonte, Marianggy Gómez, Omar Hernandez, Saúl Encarnación, Yosvany Batista. Uds. han hecho que venir a México haya sido la mejor experiencia de mi vida.**

**A mis amigos Luis Jiménez y Gabriel Mejía, cuya ayuda fue crucial para materializar este proyecto.**

**Y por último en memoria de mi amiga Yohana Bonilla Gutierrez, Dios te tenga en santa gloria.**

## Tabla de contenido

---

	Página
<b>Resumen español</b> .....	ii
<b>Resumen inglés</b> .....	iv
<b>Dedicatoria</b> .....	vi
<b>Agradecimientos</b> .....	vii
<b>Lista de figuras</b> .....	xi
<b>Lista de tablas</b> .....	xiv
<b>Capítulo 1. Introducción</b>	
1.1 Transferencia de calor. Origen de la geotermia.....	1
1.1.1 Conducción.....	1
1.1.2 Convección.....	2
1.1.3 Radiación.....	2
1.1.4 Gradiente Geotérmico y conductividad térmica. Temperatura en el interior de la Tierra.....	2
1.2 Termodinámica.....	4
1.2.1 Principios y conceptos termodinámicos.....	4
1.2.2 Entropía y Entalpía.....	5
1.2.3 Leyes de la Termodinámica.....	6
1.2.3.1 Principio cero.....	6
1.2.3.2 Primera Ley.....	7
1.2.3.3 Segunda Ley.....	7
1.2.3.4 Tercera Ley.....	7
1.3 Dinámica computacional de fluidos (DCF).....	8
<b>Capítulo 2. Ecuaciones gobernantes del transporte de masa y energía en los yacimientos geotérmicos</b>	
2.1 Ecuaciones para el transporte de masa y energía.....	10
2.2 Relaciones termodinámicas y de transporte para las propiedades del agua.....	13
<b>Capítulo 3. Discretización de ecuaciones de transporte por el método del Volumen Finito. Algoritmo Simple</b>	
3.1 Discretización de ecuaciones de transporte. Método del Volumen Finito.....	15
3.2 Esquemas para aproximar $\phi$ en la interface de cada nodo.....	17
3.2.1 Esquema centrado.....	17
3.2.2 Esquema de diferencias centrales hacia adelante “upwind”.....	18

3.2.3	Esquema exponencial.....	20
3.3	Linealización de la Fuente.....	21
3.4	Forma final de los coeficientes de discretización.....	21
3.5	Cálculo de las propiedades termodinámicas en las intercaras.....	24
3.6	Algoritmo SIMPLE.....	24
3.7	Convergencia del algoritmo SIMPLE.....	29
<b>Capítulo 4. Funcionamiento del programa y problemas</b>		
4.1	Estructura del simulador.....	32
4.1.1	Módulo principal.....	35
4.1.2	Módulo de algoritmos.....	40
4.1.3	Módulo de propiedades del agua.....	41
<b>Capítulo 5. Validación y problemas propuestos</b>		
5.1	Modelo de caja (Problema 1).....	43
5.2	Modelo de caja (Problema 2).....	46
5.3	Modelo de caja (Problema 3).....	58
<b>Capítulo 6. Conclusiones.....</b>		66
<b>Lista de referencias bibliográficas.....</b>		70
<b>Apéndices.....</b>		72

## Lista de figuras

---

Figura		Página
1	Gradiente geotérmico normal.....	3
2	Regiones termodinámicas del agua en función de la presión y Temperatura.....	14
3	Nodos vecinos (N,W,E,S) y localizaciones vecinas en la interface (n,w,e,s).....	16
4	Mallado escalonado.....	25
5	Mallado escalonado indicando nodos vecinos e intercaras.....	26
6	Ejemplo del mallado escalonado para la matriz de temperatura y velocidades $\mathbf{u}$ y $\mathbf{v}$ .....	27
7	Sistema de ecuaciones para el mallado escalonado.....	28
8	Representación de los residuales por iteración para varios intervalos de tiempo.....	31
9	Diagrama de flujo general del simulador.....	32
10	Diagrama de flujo general del simulador.....	33
11	Diagrama de flujo general del simulador. Solución numérica.....	34
12	Interfase de loadmat.m.....	37
13	Interfase savemat.m.....	37
14	Condiciones de frontera del modelo de caja.....	42
15	Resultados esperados del modelo de caja.....	43
16	Isotermas: Resultado del modelo de caja.....	43
17	Isotermas del modelo simulado.....	44
18	Isobaras del modelo simulado.....	44

19	Patrón de flujo del modelo simulado.....	45
20	Comparación del modelo de Sorey contra el modelo generado por el simulador de esta tesis .....	45
21	Condiciones de flujo y temperatura iniciales.....	48
22	Condiciones de flujo y temperatura a los 0.0056 días.....	48
23	Condiciones de flujo y temperatura a los 20.396 días.....	48
24	Condiciones de flujo y temperatura a los 3440 días.....	49
25	Condiciones de flujo y temperatura a los 566.72 días.....	49
26	Condiciones de flujo y temperatura a los 632.43 días.....	49
27	Gráfico de convergencia.....	50
28	Mallado escalonado.....	50
29	Representación de presión, temperatura y velocidades u y v durante 630 días de simulación.....	51
30	Contornos de presión y temperatura al inicio de la simulación.....	52
31	Contornos de presión y temperatura a los 2 días.....	53
32	Contornos de presión y temperatura a los 20 días.....	54
33	Contornos de presión y temperatura a los 60 días.....	55
34	Contornos de presión y temperatura a los 345 días.....	56
35	Contornos de presión y temperatura al final de la simulación.....	57
36	Convergencia de las ecuaciones.....	58
37	Condiciones de flujo y temperatura iniciales.....	59
38	Condiciones de flujo y temperatura a los 12.75 días.....	59
39	Condiciones de flujo y temperatura a los 326.70 días.....	59
40	Condiciones de flujo y temperatura a los 480.60 días.....	61
41	Condiciones de flujo y temperatura a los 787.25 días.....	61

42	Condiciones de flujo y temperatura a los 1513.59 días.....	61
43	Contornos de presión y temperatura al inicio de la simulación.....	62
44	Contornos de presión y temperatura a los 1513.59 días.....	63
45	Lámparas de lava.....	65
46	Funcionamiento de lámpara de lava.....	65
47	Flujo grama del método simple.....	73

## Lista de tablas

---

Tabla		Página
1	Función $A( P )$ para los diferentes esquemas.....	23
2	Fórmulas para calcular F y D en las caras del volumen de control.....	23
3	Volúmenes y áreas en las caras del volumen de control.....	24
4	Coeficientes de discretización para las ecuaciones de momento.....	29
5	Ecuaciones de residuales.....	30
6	Entradas para el problema 2.....	47
7	Entradas para el problema 3.....	60

# Capítulo 1. Introducción

---

## 1.1. Transferencia de calor, el origen de la geotermia.

Los procesos geodinámicos que suceden en la Tierra tienen su origen en la generación y transferencia de calor en el interior de la Tierra. El calor es el causante del movimiento de las placas tectónicas, involucra a la litósfera y astenósfera, así como diversos procesos que ocurren a mayor profundidad, como la convección del manto y el núcleo externo. Estos fenómenos de convección del manto y del núcleo dan origen al campo magnético terrestre. La geotermia, no es más, que el resultado de dichos procesos, pero a pesar de esto, entender los mecanismos mediante los cuales funciona y describirlos no ha sido tan sencillo.

El transporte de calor en el interior de la Tierra se lleva a cabo por medio de tres mecanismos: conducción, convección y radiación; cada uno de ellos, tienen diferente grado de importancia en las diferentes capas: en la corteza el principal medio de transporte de calor es la conducción mientras que en el manto es la convección y radiación.

Para poder entender cómo funciona un yacimiento geotérmico primero debemos tener claro éstas definiciones respecto al transporte de calor.

### 1.1.1 Conducción

La conducción es la forma como se transporta el calor de un cuerpo más caliente a uno más frío con el cual se encuentra en contacto. La eficiencia de ésta depende de la conductividad térmica y que nos dice cuál será la diferencia de temperatura provocada por un flujo de calor: a mayor conductividad menor será la diferencia de temperatura a través del material.

### 1.1.2 Convección

La convección es un proceso un poco más complejo que se da solamente en fluidos (líquidos y gases). Al estar en contacto una fuente de calor con la parte inferior de un fluido, ésta se expande y se vuelve menos densa que la parte superior más fría, por lo cual tiende a subir, así que la parte fría queda en contacto con la fuente de calor repitiéndose de esta forma el proceso, **dando origen a lo que se llama celdas de convección**, en las cuales existen corrientes y por lo tanto flujos de masa, ascendentes y descendentes. Este mecanismo se va a generar a partir de un cierto valor de la diferencia de temperatura y depende de la viscosidad y densidad del fluido.

### 1.1.3 Radiación

La radiación es una forma de transporte de calor que es importante a temperaturas altas. En realidad todos los cuerpos que tienen temperatura por arriba del cero absoluto (cero grados Kelvin o  $-273.15^{\circ}\text{C}$ ) emiten radiación, pero la frecuencia de la radiación emitida es proporcional a la temperatura del material: ejemplo de esto somos los seres humanos, que emitimos radiación en el infrarrojo, pero un trozo de hierro calentado a temperaturas muy altas empezará a emitir en el espectro visible.

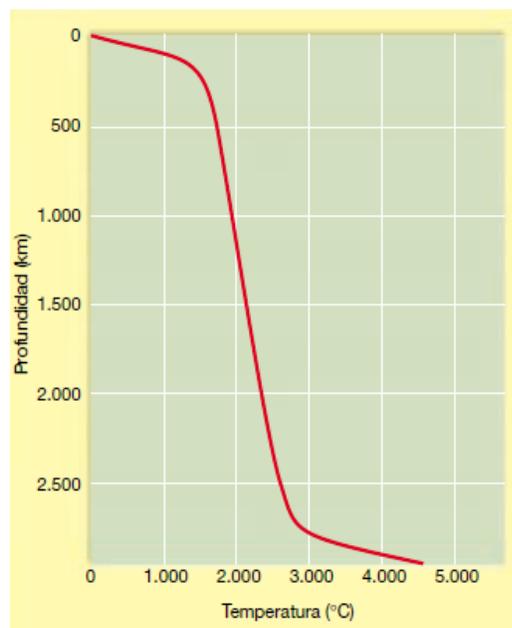
### 1.1.4 Gradiente geotérmico y conductividad térmica. Temperatura en el interior de la Tierra.

A medida que se va profundizando hacia el centro de la Tierra, se puede apreciar por lo general un incremento en la temperatura; es a esa variación de la temperatura con la profundidad a lo que se le llama gradiente geotérmico. La mayoría de las zonas del planeta en las que no se tienen anomalías térmicas se agrupan alrededor de un promedio de 20 a  $30^{\circ}\text{C}/\text{km}$ , a este valor se le considera el gradiente geotérmico normal. (Tarbuck, E. J., & Lutgens, F. K.). En la Figura 1 se representa la temperatura en función de la profundidad para condiciones normales.

El flujo de calor en la superficie de la Tierra se calcula como el producto del gradiente geotérmico por la conductividad térmica de las rocas, siendo estos dos

parámetros determinados directamente. El gradiente geotérmico se determina midiendo la temperatura a lo largo de pozos suficientemente profundos para evitar las perturbaciones en las capas superficiales (hasta 200 m) provocadas por las variaciones diurnas y estacionales de temperatura, así como por el flujo de aguas subterráneas. Para determinar o estimar la variación de la temperatura dentro del manto se requiere de algunas hipótesis. Por otra parte, la conductividad térmica de los materiales en la naturaleza, es la propiedad que tienen las rocas o los fluidos para permitir el flujo de calor a través de ellos. Para determinar la conductividad térmica existen diferentes métodos, un estándar convencional es el método de la barra dividida. Este método consiste en colocar la muestra de roca en forma de disco entre dos barras metálicas cilíndricas mantenidas a temperatura constante. Luego de alcanzar el estado estable, se estima la conductividad térmica de la muestra mediante la comparación de la caída de temperatura en sus caras con la caída que se produce en las caras de los materiales de referencia de conductividad conocida que rodean la muestra.

La conductividad térmica del agua por ejemplo, es de  $0.6 \text{ W/m}^{\circ}\text{K}$  aproximadamente, mientras que la de las rocas en general, es entre  $0.5\text{-}6.5 \text{ W/m}^{\circ}\text{K}$ . (Chekhonin E., Parshin A., Pissarenko D, *et. al* 2013).



**Figura1. Gradiente geotérmico normal. (Tarbuck, E. J., & Lutgens, F. K.)**

El flujo calórico terrestre promedio en los continentes y en los océanos es de 65 y 101 mWm<sup>-2</sup>, respectivamente; que considerados aleatoriamente entregan un promedio mundial de 87 mWm<sup>-2</sup> (H.N Pollack, Suzanne J. Hurter and Jeffrey R. Johnson, 1993).

En zonas geológicamente activas, el valor del gradiente geotérmico puede ser superior a 3° C/100 m, este fenómeno también se presenta en un sistema de fallas profundas o debido a la existencia de acuíferos que pueden transportar el calor en sentido vertical. Esto es denominado como un gradiente geotérmico anómalo y surge en lugares idóneos para la extracción de calor con fines de generación de electricidad.

## **1.2 Termodinámica**

La palabra termodinámica se refiere a la dinámica del calor y a todos los tipos de formas de energía (mecánica, eléctrica, química, molecular, etc.). Se puede decir que "la termodinámica es la ciencia que estudia las transformaciones energéticas" (Erich A. Müller, 2002).

La termodinámica describe las relaciones entre un conjunto de variables de estado (como pueden ser temperatura, presión, volumen, etc.), y las leyes de la termodinámica, permitiendo así, conocer la evolución del sistema.

La termodinámica estudia cualquier sistema constituido por un gran número de partículas: un simple hilo metálico, un cristal, una reacción química, un rayo de luz, los océanos, la atmósfera, etc. Numerosas teorías científicas han ido modificándose a lo largo del tiempo como la mecánica clásica que fue reemplazada por la mecánica relativista, en cambio la termodinámica no puede cambiar, ya que en ese caso la naturaleza estaría violando sus propias leyes (Müller, 2002).

### **1.2.1 Principios y conceptos termodinámicos**

Los principios termodinámicos se han extendido al cálculo y a la predicción de las propiedades de una sustancia relacionada con la energía, así como a las condiciones de equilibrio de los procesos físicos o químicos. Los elementos que conforman a estos principios son:

- **Sistema y proceso:** es una cantidad de materia que forma parte del universo pero que se considera aparte para su estudio. Un proceso es el conjunto de cambios que tienen lugar en el sistema, de esta manera, el hidrógeno, el oxígeno y el agua constituyen un sistema y la combustión del hidrógeno para formar agua constituye un proceso (Jiménez-Morales y Lemos-Fernández, 2001).
- **Estado del sistema:** se determina por el valor de ciertas magnitudes medibles experimentalmente y se denominan variables de estado. Estas son volumen, presión, temperatura, etc. Los sistemas termodinámicos alcanzan “estados de equilibrio termodinámico”; en éstos los valores de las variables de estado permanecen fijos mientras no se modifiquen las condiciones impuestas al sistema (Müller, 2002).
- **Medio exterior:** es todo aquello que no constituye el sistema. El sistema junto con el medio exterior constituyen el “Universo”.
- **Paredes (o fronteras):** el sistema se encuentra separado del medio exterior por estas barreras denominadas paredes. Estas se clasifican en móviles-rígidas, diatérmica-adiabática y permeables-impermeables.
- **Sistemas abiertos y cerrados:** un sistema se encuentra cerrado cuando no existe intercambio de materia con sus alrededores. Por otro lado, se denomina abierto cuando tiene lugar dicho intercambio. En un sistema de varias fases cada una de ellas constituye un sistema abierto, puesto que la materia tiene libertad de entrar o salir, aunque el sistema globalmente pueda ser cerrado.
- **Sistema aislado:** un sistema cerrado puede cambiar libremente calor y trabajo con sus alrededores. En cambio un sistema cerrado se considera térmicamente aislado cuando las paredes que le limitan no permiten el flujo de calor; mecánicamente aislado cuando está limitado por paredes rígidas y totalmente aislado cuando son imposibles de todo punto los intercambios de materia y de energía (Boltzmann, 1986).

### 1.2.2 Entropía y Entalpía

La entropía es una propiedad intrínseca de la materia caracterizada porque su valor se incrementa al crecer la ineficiencia de la energía total del sistema. Su magnitud depende sólo de la naturaleza de la materia considerada y del estado en que se

encuentra y es independiente de su posición externa o de su movimiento respecto a otros cuerpos (Jiménez-Morales y Lemos-Fernández, 2001).

La medida de la entropía aumenta de manera considerable con el aumento de calor, sin embargo, la cantidad de calor añadida no es la única medida de incremento. Resulta que el incremento de entropía que acompaña la adición de una cantidad determinada de calor a un sistema, crece al disminuir la temperatura a la cual se cede el calor. En el caso de un gas el cual puede expandirse libremente tanto térmica como mecánicamente dentro de un sistema aislado, existe un proceso isoterma en el cual hay crecimiento de entropía no por adición de calor, si no por el rozamiento de las partículas dentro del mismo sistema. En este caso el calor es cedido al sistema por degradación de trabajo mecánico. La entropía es una propiedad intrínseca de la materia, cuyo cambio en cualquier proceso depende sólo de los estados inicial y final y no de la trayectoria (Müller, 2002).

Por otro lado, la entalpía es una función de estado de la termodinámica donde la variación permite expresar la cantidad de calor puesto en juego durante la transformación isobárica, es decir, a presión constante en un sistema termodinámico. Se trata de una transformación en el curso de la cual se puede recibir o aportar energía. En este sentido la entalpía es numéricamente igual al calor intercambiado con el ambiente exterior al sistema en cuestión. Generalmente se mide en joules.

En síntesis, para medir la eficacia o degradación de la energía de un sistema se usa la entropía como referencia. Por otra parte, para entender la cantidad de energía que un sistema intercambia con el entorno, la referencia es la entalpía.

## **1.2.3 Leyes de la Termodinámica**

### **1.2.3.1 Principio cero**

Establece que existe una propiedad denominada temperatura empírica que es común para todos los estados en equilibrio termodinámico. Es decir, dos cuerpos de diferente temperatura, al estar en contacto uno con otro, en algún punto llegarán a una temperatura equilibrada para ambos cuerpos.

### 1.2.3.2 Primera ley de la termodinámica

Es también conocida como principio de conservación de la energía para la termodinámica. Establece que si se realiza trabajo sobre un sistema o bien éste intercambia calor con otro, la energía interna del sistema cambiará. Visto de otra forma, esta ley permite definir el calor como la energía necesaria que debe intercambiar el sistema para compensar las diferencias entre trabajo y energía interna. Con la colaboración y redefinición de varios autores, se llegaron a formular las bases de la termodinámica de una manera matemática:

$$\Delta U = Q - W, \quad (1)$$

donde  $U$  es la energía del sistema aislado,  $Q$  es la cantidad de calor aportado al sistema y  $W$  es el trabajo realizado (Reif, 1985).

### 1.2.3.3 Segunda ley de la termodinámica

Indica la dirección en la que deben llevarse a cabo los procesos termodinámicos y, por lo tanto, la imposibilidad de que ocurran en el sentido contrario. También establece, en algunos casos, la imposibilidad de convertir completamente toda la energía de un tipo a otro sin pérdidas. De esta forma, la segunda ley impone restricciones para las transferencias de energía que hipotéticamente pudieran llevarse a cabo teniendo en cuenta sólo el primer principio. Esta ley apoya todo su contenido aceptando la existencia de la entropía, de tal manera que, para un sistema aislado (que no intercambia materia ni energía con su entorno), la variación de la entropía siempre debe ser mayor que cero.

Debido a esta ley también se tiene que el flujo espontáneo de calor siempre es unidireccional, desde los cuerpos de mayor temperatura hacia los de menor temperatura, hasta lograr un equilibrio térmico (Callen, 1985).

### 1.2.3.4 Tercera ley de la termodinámica

El tercer principio no permite hallar el valor absoluto de la entropía. Sin embargo, la constancia de la entropía cuando la temperatura tiende a cero, da la posibilidad de elegir esta constante como punto de referencia de la entropía, por ende, permite

determinar la variación de la misma en los procesos. El Principio de Nernst como lo enunció Planck, dice que es imposible alcanzar una temperatura igual al cero absoluto mediante un número finito de procesos físicos. Puede formularse también como que a medida que un sistema dado se aproxima al cero absoluto, su entropía tiende a un valor constante específico. La entropía de los sólidos cristalinos puros puede considerarse cero bajo temperaturas iguales al cero absoluto.

Es importante remarcar que los principios o leyes de la termodinámica son válidas siempre para los sistemas macroscópicos, pero inaplicables a nivel microscópico (Callen, 1985).

### **1.3 Dinámica computacional de fluidos (DCF)**

La **Dinámica Computacional de Fluidos**, (CFD por sus siglas en inglés), consiste en el análisis del flujo de fluido, transferencia de calor y de otros fenómenos asociados como mezclas químicas, a través del uso de simulaciones por computadora. Es la parte de la mecánica de fluidos que analiza los métodos discretos para resolver las ecuaciones de continuidad, cantidad de movimiento y transporte de energía de forma simultánea.

La aplicación de esta ciencia abarca muchas áreas, como la aerodinámica, hidrodinámica de embarcaciones, geohidrología, oceanografía, etc. La simulación numérica se hace con el objetivo de determinar el comportamiento de un sistema real, aproximando el resultado de las ecuaciones diferenciales parciales que definen el sistema mediante métodos numéricos, utilizando el poder de cómputo de los ordenadores modernos.

Debido a la naturaleza no-lineal de las ecuaciones, variables implícitas ligadas y la incorporación de varias dimensiones incluyendo la variable tiempo en su formulación, es prácticamente imposible resolver las ecuaciones de estado de forma analítica, a menos que una gran cantidad de suposiciones y aproximaciones sean introducidas. Es por esto que la dinámica computacional de fluidos es una herramienta poderosa para simular el flujo de fluidos y permite resolver las ecuaciones de forma simultánea de tal manera que pueda llevarse a cabo en la práctica. Además, algunas de las propiedades físicas de un sistema real dependen de la temperatura y presión, y tomando en cuenta

que los términos convectivos no son factores lineales, hace necesario utilizar métodos iterativos para la solución de las ecuaciones.

Particularmente para el caso del estudio de yacimientos geotérmicos, es posible utilizar la DCF para reproducir y simular el comportamiento del yacimiento en diferentes escenarios.

Por supuesto, que el uso de la DCF tiene ventajas y desventajas; una de las principales ventajas es que la simulación podrá contener tanta cantidad de detalle como el usuario indique según las ecuaciones y parámetros que decida considerar, pero a mayor nivel de detalle y mayor cantidad de variables, se requerirá mayor poder de cómputo para resolver dichas ecuaciones. También hay que tener en consideración que no hay una garantía de la certeza de los resultados obtenidos mediante estas simulaciones, por lo que hay que validar los resultados de ser posible con datos experimentales.

Los ingredientes principales para tener éxito en el estudio de la DCF son la experiencia en la simulación numérica de sistemas complejos, el correcto entendimiento de la física del flujo de fluido y contar con información suficiente de las propiedades físicas del sistema así como de la respuesta observada con la densidad espacial y temporal que se requiera de acuerdo a los objetivos del estudio.

Adicionalmente hay que tener en claro dos conceptos muy importantes en la DCF: **convergencia**, la cual es la propiedad de un método numérico para llegar a una solución la cual aproxime a la solución exacta, **estabilidad**, que está asociada al efecto que tienen pequeñas variaciones en los parámetros o condiciones de operación, en la respuesta del sistema. Es decir, la DCF será más estable a medida que pequeños errores en los datos, por truncamiento o por aproximaciones de las derivadas, no produzcan una respuesta del sistema tan oscilante. Si un método numérico no es estable, incluso un mal redondeo puede causar oscilaciones y causar que el método no converja (diverja). (H. K. Versteeg & W. Malalasekera, 1995)

## Capítulo 2. Ecuaciones gobernantes del transporte de masa y energía en los yacimientos geotérmicos

---

### 2.1 Ecuaciones para el transporte de masa y energía.

Para describir cuantitativamente el comportamiento físico de un sistema geotérmico, es necesario conocer el conjunto de ecuaciones matemáticas que definen el sistema y que rigen los fenómenos físicos de dicho sistema. Se requiere saber las condiciones iniciales correspondientes al estado de equilibrio en las que se encuentra el sistema previo a su modificación y las condiciones de frontera que establecen el tipo de interacción del sistema con su entorno. Un aspecto básico es disponer de los valores de los parámetros que determinan las propiedades físicas del sistema respecto a los fenómenos relevantes que se estén estudiando. Luego, se procede a resolver el conjunto de ecuaciones mediante el uso de técnicas numéricas, y así se conforma el simulador numérico del sistema físico en investigación.

A continuación se presenta el desarrollo de las ecuaciones diferenciales utilizadas para definir el comportamiento del sistema. Estas ecuaciones han sido propuestas por diferentes autores como Patankar en 1972, Pruess en 1988, Malalasekera en 1995 y muchos otros. Cada autor propone métodos diferentes partiendo de los mismos principios físicos de conservación, es decir de la conservación de la masa o ecuación de continuidad y la ecuación de conservación de la energía.

Se eligió el **método del volumen finito** propuesto por Patankar y Spalding, 1972 y que también fue aplicado por Malalasekera en 1995.

Las suposiciones en el planteamiento del problema son:

- La formación rocosa a través de la cual fluye el fluido geotérmico se considera como un medio poroso, homogéneo e isotrópico.
- El flujo puede ser en una, o dos dimensiones.
- La presión capilar es despreciable a causa de las altas temperaturas.

- Se tiene equilibrio termodinámico, es decir, las mismas condiciones de presión y temperatura en la roca y el fluido.
- El fluido geotérmico es agua pura en una sola fase, y en estado líquido.
- El medio se encuentra 100% saturado.

Ecuación de continuidad:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0. \quad (2)$$

Ecuación de cantidad de movimiento en  $x$ :

$$\frac{\partial(\rho u)}{\partial t} + \nabla \cdot (\rho \mathbf{u}u) = -\frac{\partial p}{\partial x} + \nabla \cdot (\mu \nabla u) + \rho g_x + \Phi_x. \quad (3)$$

Ecuación de cantidad de movimiento en  $y$ :

$$\frac{\partial(\rho v)}{\partial t} + \nabla \cdot (\rho \mathbf{u}v) = -\frac{\partial p}{\partial y} + \nabla \cdot (\mu \nabla v) + \rho g_y + \Phi_y. \quad (4)$$

Ecuación de transporte de energía

$$\frac{\partial(\rho C_P T)}{\partial t} + \nabla \cdot (\rho \mathbf{u} C_P T) = \nabla \cdot (k \nabla T) + S_h. \quad (5)$$

Donde  $\rho$  es la densidad,  $\mathbf{u}$  es la velocidad en  $x$ ,  $v$  es la velocidad en  $y$ ,  $\mu$  es la viscosidad,  $P$  es la presión,  $g$  es la gravedad,  $\partial x$  y  $\partial y$  son discretizaciones en el espacio,  $\partial t$  es la variación en el tiempo,  $C_p$  es la capacidad calorífica,  $T$  es la temperatura,  $K$  es la conductividad térmica,  $S_h$  es el término de fuente o sumidero, y finalmente  $\phi_x$  y  $\phi_y$  son otros efectos externos en las ecuaciones de cantidad de movimiento, que en este trabajo de tesis serán despreciados.

Estas ecuaciones pueden escribirse de forma general tal como se indica en la Ecuación (6) y a su vez tienen un significado físico tal como se indica

$$\underbrace{\frac{\partial(\rho\phi)}{\partial t}}_{\text{acumulación}} + \underbrace{\nabla \cdot (\rho\mathbf{u}\phi)}_{\text{convección}} = \underbrace{\nabla \cdot (\Gamma\nabla\phi)}_{\text{transporte difusivo}} + \underbrace{S_\phi}_{\text{generación}} \quad (6)$$

Donde  $\phi = 1$  para la ecuación de continuidad,  $\phi = u$  ó  $\phi = v$  para la ecuación de cantidad de movimiento y  $\phi = C_p \cdot T$  para la ecuación de transporte de energía. La difusividad es  $\Gamma \equiv \mu$  para la ecuación de cantidad de movimiento y  $\Gamma \equiv K / C_p$  para la ecuación de energía. En el Capítulo 3 retomaremos esta ecuación general.

Pueden simplificarse las ecuaciones (3-5) si se sustituye la ecuación de continuidad (ecuación 2) en las demás. También por conveniencia, se define la presión modificada  $\mathcal{P}$  mediante la siguiente ecuación que contiene al término de disipación viscosa  $\Phi$ :

$$\mathcal{P} = p - \frac{1}{3}\mu\nabla \cdot \mathbf{u}. \quad (7)$$

Además, si no hay gravedad en la coordenada  $x$  ( $g_x = 0$ ), la gravedad en  $y$  es  $g_y = -g$ , no hay generación de calor por unidad de volumen  $S_h = 0$ , y resolviendo algunos operadores vectoriales, se obtienen las ecuaciones (8-11):

$$\frac{\partial\rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} = 0, \quad (8)$$

$$\rho \frac{\partial u}{\partial t} + \rho \mathbf{u} \cdot \nabla u = -\frac{\partial \mathcal{P}}{\partial x} + \mu \nabla^2 u + \nabla \mu \cdot \nabla u, \quad (9)$$

$$\rho \frac{\partial v}{\partial t} + \rho \mathbf{u} \cdot \nabla v = -\frac{\partial \mathcal{P}}{\partial y} + \mu \nabla^2 v + \nabla \mu \cdot \nabla v - \rho g, \quad (10)$$

$$\rho \frac{\partial(C_p T)}{\partial t} + \rho \mathbf{u} \cdot \nabla(C_p T) = k \nabla^2 T + \nabla k \cdot \nabla T, \quad (11)$$

(R.B. Bird, W.E Stewart, E.N. Lightfoot, 2002).

La ley de Darcy describe el flujo de fluidos en medios porosos para **números de Reynolds bajos**, que indiquen que el flujo es laminar.

$$u = -\frac{k}{\mu} \left[ \frac{P_m - P_n}{D_{nm}} - \rho g \right], \quad (12)$$

$$\underline{F} = \rho u,$$

donde,  $u$  es la velocidad,  $k$  es la permeabilidad,  $\mu$  es la viscosidad,  $P$  es la presión,  $D$  es la distancia entre nodos  $n$  y  $m$ ,  $\rho$  es la densidad,  $g$  es la gravedad y  $F$  es el flujo másico (Cruz, 1998).

También vale la pena destacar que existe un factor el cual permite saber la relación de importancia del término convectivo sobre el difusivo y este es llamado el número de Peclet.

$$Pe = \frac{F}{D} = \frac{\rho u dx}{\Gamma}, \quad (13)$$

En este caso  $D$  representa el término difusivo que viene dado por la siguiente relación:

$$D = \frac{\Gamma}{dx}, \quad (14)$$

En el Capítulo 3 se explican los mismos términos para el caso de 2 dimensiones.

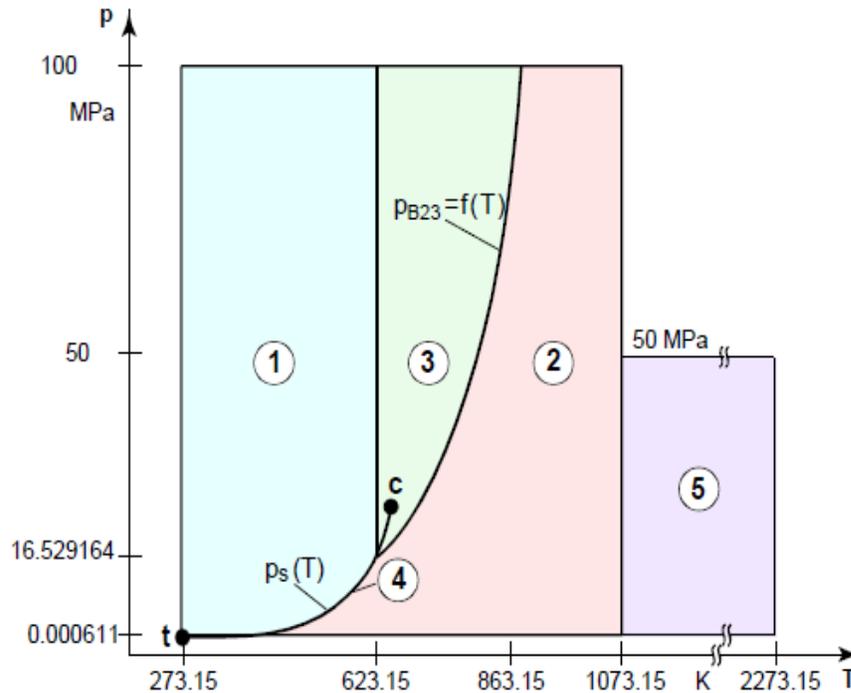
## 2.2 Relaciones termodinámicas y de transporte para las propiedades del agua.

En la Figura 2 se representan varias zonas para los estados físicos del agua pura.

**La zona 1** es el líquido sub enfriado, **la zona 2** es la zona de vapor sobre calentado, **la zona 3 y el punto c**, representan zonas críticas donde cualquier cambio de presión o temperatura genera un gran cambio en las propiedades del agua; **la línea 4** representa el equilibrio líquido-vapor.

Por último **la zona 5** representa al fluido súper crítico. En este proyecto sólo se trabajará con la región 1 que es la de líquido, debido a que abarcar las otras zonas

requiere desarrollar otras ecuaciones como son las ecuaciones de estado para el cambio de fase de líquido a vapor y eso está fuera de los alcances de este proyecto.



**Figura 2. Regiones termodinámicas del agua en función de la presión y temperatura (Hans-Jachim Kretzschmar - IAPWS, 2007).**

Las ecuaciones de estado que representan la zona 1, están descritas a detalle en los trabajos presentados por la Asociación Internacional de las Propiedades del Agua y el Vapor (IAPWS, 2007, <http://www.iapws.org>).

Y como parte de los apéndices, está incluido el manual que contiene el funcionamiento de éste módulo o add-on para el matlab el cual se encarga de realizar los cálculos de todas las propiedades termodinámicas

El rango de temperaturas de estas ecuaciones abarcan desde los 273.15°K hasta los 1073.13 °K, y rangos de presiones de 0.000611 MPa. hasta los 100 MPa. (IAPWS, 2007).

## Capítulo 3. Discretización de ecuaciones de transporte por el método del Volumen Finito. Algoritmo SIMPLE

---

### 3.1 Discretización de las ecuaciones de transporte. Método de volumen finito (FVM)

La ecuación general de transporte viene dada por:

$$\frac{\partial(\rho\phi)}{\partial t} + \nabla \cdot (\rho\mathbf{u}\phi) = \nabla \cdot (\Gamma\nabla\phi) + S_\phi, \quad (15)$$

donde  $\phi = 1$  para la ecuación de continuidad,  $\phi = u$  ó  $\phi = v$  para la ecuación de cantidad de movimiento y  $\phi = T$  para la ecuación de transporte de energía. La difusividad es  $\Gamma \equiv \mu$  para la ecuación de cantidad de movimiento y  $\Gamma \equiv K / C_p$  para la ecuación de energía. El término de generación por unidad de volumen  $S_\phi$  viene dado por:

$$S_\phi = \begin{cases} S_m & \text{para la ecuación de continuidad,} \\ \rho g_\xi & \text{para la ecuación de cantidad de movimiento,} \\ S_h & \text{para la ecuación de energía.} \end{cases} \quad (16)$$

Integrando la Ecuación (15) en un volumen de control se obtiene la ecuación de transporte macroscópico:

$$\iiint_V \frac{\partial(\rho\phi)}{\partial t} dV = \iiint_V \nabla \cdot (\Gamma\nabla\phi - \rho\mathbf{u}\phi) dV + \iiint_V S_\phi dV. \quad (17)$$

Aplicando el teorema de Gauss al campo vectorial  $\Gamma\nabla\phi - \rho\mathbf{u}\phi$ , se cambia la integral triple por una integral sobre la superficie cerrada del volumen de control:

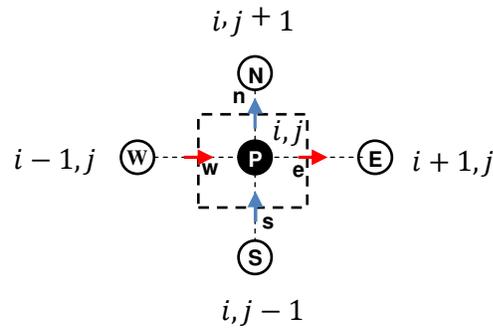
$$\iiint_V \frac{\partial(\rho\phi)}{\partial t} dV = \oiint_S (\Gamma\nabla\phi - \rho\mathbf{u}\phi) \cdot d\mathbf{S} + \iiint_V S_\phi dV. \quad (18)$$

La convención del vector normal exterior del teorema de gauss es usado para escoger los signos de  $(\Gamma \nabla \phi - \rho \mathbf{u} \phi) \cdot d\mathbf{S}$

$$(\Gamma \nabla \phi - \rho \mathbf{u} \phi) \cdot d\mathbf{S} = \text{negativo si entra al volumen,}$$

$$(\Gamma \nabla \phi - \rho \mathbf{u} \phi) \cdot d\mathbf{S} = \text{positivo si sale del volumen.}$$

Discretizando la ecuación anterior mediante volumen finito, denotando las letras mayúsculas (N,W,E,S) a los centros de cada celda y las letras minúsculas (n,w,e,s) a los puntos centrales de las fronteras de la celda:



**Figura 3. Nodos vecinos (N, W, E, S) y localizaciones vecinas en la interface (n, w, e, s)**

Aplicando diferencias centrales a la Ecuación (18) se obtiene:

$$\begin{aligned} \frac{\rho_P \phi_P - \rho_P^0 \phi_P^0}{\Delta t} \Delta x \Delta y &= \left( \Gamma \cdot \frac{\partial \phi}{\partial x} - \rho u \phi \right)_e \Delta y - \left( \Gamma \cdot \frac{\partial \phi}{\partial x} - \rho u \phi \right)_w \Delta y + \dots \\ &\dots + \left( \Gamma \cdot \frac{\partial \phi}{\partial y} - \rho v \phi \right)_n \Delta x - \left( \Gamma \cdot \frac{\partial \phi}{\partial y} - \rho v \phi \right)_s \Delta x + S_\phi \Delta x \Delta y. \end{aligned} \quad (19)$$

Definiendo  $F_e$  como el flujo másico  $F_e = (\rho u) \Delta y$  y también aplicando diferencias finitas a los términos  $\frac{\partial \phi}{\partial x}$ ,  $\frac{\partial \phi}{\partial y}$ :

$$\begin{aligned}
\left(\frac{\rho_P}{\Delta t} \Delta x \Delta y\right) \phi_P &= \left(\Gamma_e \Delta y \cdot \frac{\phi_E - \phi_P}{\Delta x} - F_e \phi_e\right) - \left(\Gamma_w \Delta y \cdot \frac{\phi_P - \phi_W}{\Delta x} - F_w \phi_w\right) + \dots \\
&\dots + \left(\Gamma_n \Delta x \cdot \frac{\phi_N - \phi_P}{\Delta y} - F_n \phi_n\right) - \left(\Gamma_s \Delta x \cdot \frac{\phi_P - \phi_S}{\Delta y} - F_s \phi_s\right) \\
&\quad + \left(\frac{\rho_P^0 \phi_P^0}{\Delta t} + S_\phi\right) \Delta x \Delta y.
\end{aligned} \tag{20}$$

Ahora, definiendo la difusión como  $D_e = \Gamma \frac{\Delta y}{\Delta x}$ , podemos reescribir la Ecuación (20)

como sigue:

$$\begin{aligned}
\left(\frac{\rho_P}{\Delta t} \Delta x \Delta y\right) \phi_P &= (D_e \cdot (\phi_E - \phi_P) - F_e \phi_e) - (D_w \cdot (\phi_P - \phi_W) - F_w \phi_w) + \dots \\
&\dots + (D_n \cdot (\phi_N - \phi_P) - F_n \phi_n) - (D_s \cdot (\phi_P - \phi_S) - F_s \phi_s) + \left(\frac{\rho_P^0 \phi_P^0}{\Delta t} + S_\phi\right) \Delta x \Delta y
\end{aligned} \tag{21}$$

### 3.2 Esquemas para aproximar $\phi$ en la interface de cada nodo

Los valores de  $\phi_n, \phi_w, \phi_e, \phi_s$  deben ser expresados en función de los valores de  $\phi$  en los nodos principales de la Figura 3, para esto, se aproximan los valores de  $\phi_n, \phi_w, \phi_e, \phi_s$  mediante varios esquemas:

#### 3.2.1 Esquema centrado (no recomendado para $|P| > 2$ )

Este esquema simplemente supone que  $\phi_e$  es el promedio de los valores adyacentes  $\phi_E, \phi_P$

$$\phi_e = \frac{\phi_E + \phi_P}{2}. \tag{22}$$

Entonces:

$$D_e \cdot (\phi_E - \phi_P) - F_e \phi_e = \left(D_e - \frac{F_e}{2}\right) \cdot \phi_E - \left(D_e + \frac{F_e}{2}\right) \phi_P. \tag{23}$$

Para escribir esta ecuación de forma compacta recurrimos a la siguiente nomenclatura, la cual es la forma general de los coeficientes de discretización, (será explicado en la Sección 3.4). Cada tipo de esquema tendrá una forma final de los coeficientes. Siendo para diferencias centrales la siguiente:

$$a_P \phi_P = a_W \phi_W + a_E \phi_E + a_S \phi_S + a_N \phi_N + b.$$

**Dónde:**

$$a_P = a_W + a_E + a_S + a_N + b,$$

$$b = \left( \frac{\rho_P^0 \phi_P^0}{\Delta t} + S_C \right) \Delta x \Delta y,$$

$$a_E = D_e - \frac{F_e}{2},$$

$$a_W = D_w + \frac{F_w}{2},$$

$$a_S = D_s - \frac{F_s}{2},$$

$$a_N = D_n + \frac{F_n}{2}.$$

(S.V. Patankar, 1980)

Este esquema de diferencias centradas sólo es recomendado cuando el fluido es muy viscoso  $|P| < 2$ , pero cuando el número de Peclet es muy alto, el esquema centrado arroja valores físicos incorrectos. Para resolver este problema, se emplea las diferencias hacia adelante "*upwind*", definidas a continuación.

### 3.2.2 Esquema de diferencias hacia adelante "*upwind*" (recomendado para $|P| > 2$ ).

El esquema "*upwind*" fue propuesto por Isaacson y Rees (1952), y retomado por Gentry, Martin y Daly (1966), para eliminar los problemas de las diferencias centradas, y se define como siguen:

$$\phi_e \begin{cases} \phi_P \text{ si } F_e > 0, \\ \phi_E \text{ si } F_e < 0. \end{cases} \quad (24)$$

Por lo tanto:

$$D_e \cdot (\phi_E - \phi_P) - F_e \phi_P = (D_e) \cdot \phi_E - (D_e + F_e) \phi_P \text{ si } F_e > 0,$$

$$D_e \cdot (\phi_E - \phi_P) - F_e \phi_E = (D_e - F_e) \cdot \phi_E - (D_e) \phi_P \text{ si } F_e < 0.$$

Es decir

$$D_e \cdot (\phi_E - \phi_P) - F_e \phi_e = (D_e + \max\{-F_e, 0\}) \cdot \phi_E - (D_e + \max\{F_e, 0\}) \cdot \phi_P. \quad (25)$$

Siguiendo la misma metodología que en el esquema de diferencias centrales, obtenemos la forma general de los coeficientes de discretización como sigue:

$$\mathbf{a}_P \phi_P = \mathbf{a}_W \phi_W + \mathbf{a}_E \phi_E + \mathbf{a}_S \phi_S + \mathbf{a}_N \phi_N + \mathbf{b}.$$

**Dónde:**

$$\mathbf{a}_P = \mathbf{a}_W + \mathbf{a}_E + \mathbf{a}_S + \mathbf{a}_N + \mathbf{b},$$

$$\mathbf{b} = \left( \frac{\rho_P^0 \phi_P^0}{\Delta t} + S_C \right) \Delta x \Delta y,$$

$$\mathbf{a}_E = \mathbf{D}_e + \max(-F_e, 0),$$

$$\mathbf{a}_W = \mathbf{D}_w + \max(F_w, 0),$$

$$a_S = D_s + \max(-F_s, 0),$$

$$a_N = D_n + \max(F_n, 0).$$

(S.V. Patankar, 1980)

### 3.2.3 Esquema exponencial

Se obtiene a partir de la ecuación analítica en una dimensión, en estado estacionario, entre dos nodos cualesquiera, por ejemplo para los nodos  $P$  y  $E$ , las condiciones de borde son

$$\phi = \phi_P \text{ si } x = 0,$$

$$\phi = \phi_E \text{ si } x = \Delta x.$$

La Ecuación (18), escrita en estado estacionario, sin generación, para la dirección  $x$  viene dada por:

$$\nabla \cdot (\Gamma \nabla \phi - \rho u \phi) = 0 \quad \Rightarrow \quad \frac{\partial}{\partial x} \left( \Gamma \cdot \frac{\partial \phi}{\partial x} - \rho u \phi \right) = 0.$$

Luego, resolviendo la ecuación diferencial de primer orden, mediante el factor integrante  $\exp\left(\frac{\rho u}{\Gamma} x\right)$ :

$$\Gamma \cdot \frac{\partial \phi}{\partial x} - \rho u \phi = C \Rightarrow \frac{d}{dx} \left( \exp\left(\frac{\rho u}{\Gamma} x\right) \cdot \phi \right) = C \cdot \exp\left(\frac{\rho u}{\Gamma} x\right)$$

$$\exp\left(\frac{\rho u}{\Gamma} x\right) \cdot (\phi - \phi_p) = C \cdot \int_0^x \exp\left(\frac{\rho u}{\Gamma} x\right) dx$$

$$\phi - \phi_p = C \cdot \exp\left(-\frac{\rho u}{\Gamma} x\right) \cdot \frac{\exp\left(\frac{\rho u}{\Gamma} x\right) - 1}{\frac{\rho u}{\Gamma}}.$$

Luego, evaluando en la segunda condición de borde

$$\phi_E - \phi_p = C \cdot \exp\left(-\frac{\rho u}{\Gamma} \Delta x\right) \cdot \frac{\exp\left(\frac{\rho u}{\Gamma} \Delta x\right) - 1}{\frac{\rho u}{\Gamma}}.$$

Dividiendo las dos últimas ecuaciones, se obtiene el perfil de  $\phi$  entre los nodos P y E:

$$\frac{\phi - \phi_p}{\phi_E - \phi_p} = \frac{\exp\left(\frac{\rho u}{\Gamma} x\right) - 1}{\exp\left(\frac{\rho u}{\Gamma} \Delta x\right) - 1} = \frac{\exp\left(P \frac{x}{\Delta x}\right) - 1}{\exp(P) - 1}.$$

Sustituyendo este perfil en la ecuación de flux,  $J_e$ :

$$J_e \equiv \Gamma \cdot \frac{\partial \phi}{\partial x} - \rho u \phi = F_e \left( \phi_p + \frac{\phi_p - \phi_E}{\exp(P) - 1} \right). \quad (26)$$

### 3.3 Linealización de la fuente

En caso que la generación de calor o masa dependa de una de las variables de transporte, entonces se supone que  $S_\phi$  es una función lineal de  $\phi_p$ , según la Ecuación (27):

$$S_\phi = S_C + S_P \cdot \phi_p. \quad (27)$$

### 3.4 Forma final de los coeficientes de discretización.

Anteriormente para el caso de diferencias centrales y “upwind”, vimos cómo quedan los coeficientes de discretización. A continuación se desarrollará la forma

general de estos coeficientes y a partir de este momento será la nomenclatura a utilizar debido a su practicidad, ya que las ecuaciones de conservación de masa, conservación de energía y momento, siguen el mismo esquema y se pueden representar de formas similares.

Escribiendo la ecuación de continuidad con generación de masa:

$$\frac{\rho_P - \rho_P^0}{\Delta t} \Delta x \Delta y + F_n - F_w + F_e - F_s = S_m \Delta x \Delta y. \quad (28)$$

y también la ecuación de transporte:

$$\frac{\rho_P \phi_P - \rho_P^0 \phi_P^0}{\Delta t} \Delta x \Delta y + J_n - J_w + J_e - J_s = S_\phi \Delta x \Delta y. \quad (29)$$

Multiplicando la Ecuación (28) por  $\phi_P$  y restándosela a la Ecuación (29) se obtiene:

$$\begin{aligned} \rho_P^0 \frac{\phi_P - \phi_P^0}{\Delta t} \Delta x \Delta y + (J_n - F_n \phi_P) - (J_w - F_w \phi_P) + (J_e - F_e \phi_P) - (J_s - F_s \phi_P) \\ = (S_\phi - S_m \phi_P) \Delta x \Delta y. \end{aligned} \quad (30)$$

Cada término  $J - F\phi_P$  se encuentra directamente con la Ecuación (25) o (26) dependiendo del tipo de esquema que se desea utilizar. Se define el coeficiente  $a_E$  tal que:

$$J_e - F_e \phi_P = a_E (\phi_P - \phi_E). \quad (31)$$

Aplicando la Definición (31) a la Ecuación (30), se obtiene la siguiente ecuación general de transporte:

$$a_P \phi_P = a_N \phi_N + a_W \phi_W + a_E \phi_E + a_S \phi_S + b. \quad (32)$$

Donde cada uno de los coeficientes se calcula con los esquemas de diferencias especiales presentados anteriormente:

$$a_N = D_n \cdot \mathcal{A}(|P_n|) + \max\{-F_n, 0\}, \quad (33)$$

$$a_W = D_w \cdot \mathcal{A}(|P_w|) + \max\{F_w, 0\}, \quad (34)$$

$$a_E = D_e \cdot \mathcal{A}(|P_e|) + \max\{-F_e, 0\}, \quad (35)$$

$$a_S = D_s \cdot \mathcal{A}(|P_s|) + \max\{F_s, 0\}, \quad (36)$$

$$a_P = a_N + a_W + a_E + a_S + \left( \frac{\rho_P^0}{\Delta t} - S_P + S_m \right) \Delta x \Delta y, \quad (37)$$

$$b = \left( \frac{\rho_P^0 \phi_P^0}{\Delta t} + S_C \right) \Delta x \Delta y. \quad (38)$$

(Versteeg *et. al* 1995), (S.V. Patankar, 1980).

Las siguientes Tablas (1, 2 y 3), explican con detalle cada uno de los coeficientes **a**, **A(|P|)**, **F** y **D**, para el caso de flujo en una, dos y tres dimensiones.

Tabla 1. Función **A(|P|)** para los diferentes esquemas. **P** es el número de Peclet (S.V. Patankar, 1980).

Esquema	Fórmula para <b>A( P )</b>
Diferencias Centrales	$1 - 0.5 P $
“Upwind”	1
Potencia	$\text{Max}(0, 1 - 0.5 P )$
Exponencial	$\frac{ P }{\exp( P ) - 1}$

Tabla 2. Fórmulas para calcular **F** y **D** en las caras del volumen de control (Versteeg *et al*, 1995).

Cara	w	e	s	n	b	t
<b>F</b>	$(\rho u)_w A_w$	$(\rho u)_e A_e$	$(\rho u)_s A_s$	$(\rho u)_n A_n$	$(\rho u)_b A_b$	$(\rho u)_t A_t$
<b>D</b>	$\frac{\Gamma_w}{dx_{WP}} A_w$	$\frac{\Gamma_e}{dx_{PE}} A_e$	$\frac{\Gamma_s}{dy_{SP}} A_s$	$\frac{\Gamma_n}{dy_{PN}} A_n$	$\frac{\Gamma_b}{dz_{BP}} A_b$	$\frac{\Gamma_t}{dz_{PT}} A_t$

**Tabla 3. Volúmenes y áreas en las caras del volumen de control (Versteeg et al, 1995).**

	<b>1D</b>	<b>2D</b>	<b>3D</b>
$\Delta V$	$\Delta x$	$\Delta x \Delta y$	$\Delta x \Delta y \Delta z$
$A_w=A_e$	1	$\Delta y$	$\Delta y \Delta z$
$A_n=A_s$	-	$\Delta x$	$\Delta x \Delta z$
$A_b=A_t$	-	-	$\Delta x \Delta y$

### 3.5 Cálculo de las propiedades termodinámicas en las intercaras.

Siendo las propiedades de estado condiciones cambiantes a lo largo de la simulación para cada volumen de control, es necesario establecer un criterio o punto en común entre las celdas para poder dar solución al sistema de ecuaciones, estos puntos son las intercaras.

Las propiedades en las intercaras son calculadas mediante interpolaciones directas. Matlab cuenta con subrutinas programadas para realizar diferentes tipos de interpolaciones, tal como los splines o interpolaciones lineales. Dentro del programa de simulación, se puede escoger el tipo de interpolación que se quiera realizar.

### 3.6 Algoritmo SIMPLE

Las ecuaciones (3), (4) y (5), (ecuaciones gobernantes para flujo laminar en dos dimensiones), presentan varias complicaciones importantes: Primero, las ecuaciones contienen términos o cantidades no lineales, como el flujo másico por ejemplo. Segundo, las ecuaciones están intrínsecamente acopladas debido a que la velocidad aparece en cada ecuación de momento y en la de continuidad. Y el último problema es la presión. No existe una ecuación independiente para la presión, sino que aparece mezclada junto a las ecuaciones de momento.

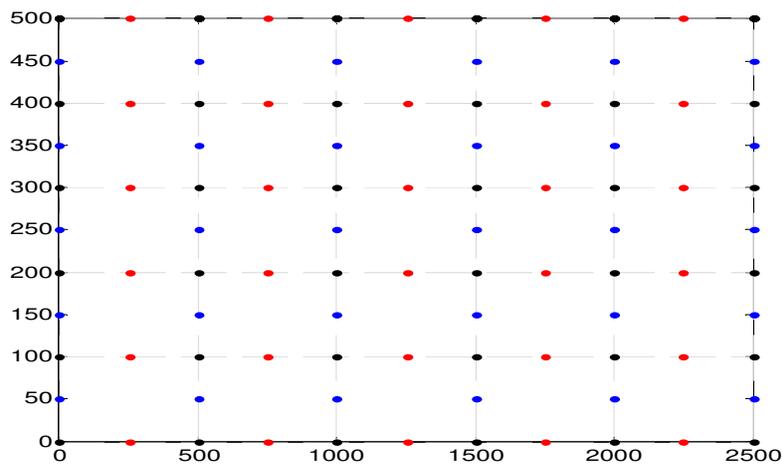
Los problemas asociados a las no linealidades en el conjunto de ecuaciones se pueden resolver mediante el uso de un algoritmo iterativo como SIMPLE desarrollado por Patankar y Spalding en 1972.

El nombre de SIMPLE significa, Semi Implicit Pressure Linked Equations, lo que se traduce a **Ecuaciones Semi Implícitas de Presión Acopladas**.

El algoritmo consiste en suponer velocidades y campos de presiones iniciales y progresivamente ir corrigiendo estas presiones y velocidades hasta alcanzar la convergencia para obtener los campos de presiones y velocidades reales.

Sin embargo Patankar (1980), indica que no pueden representarse las presiones y las velocidades en los mismos nodos ya que la presión no quedaría correctamente representada y aunque las ecuaciones tendrían solución y convergerían, los valores tanto de presiones como de velocidades, estarían incorrectos. Para resolver ese problema se planteó el uso del **mallado escalonado** (Staggered grid, de Harlow y Welch, 1965). En la Figura 4 se muestra este mallado.

La idea es evaluar las variables escalares, como la presión, densidad, temperatura, etc., en los puntos centrales pero calcular la velocidad en las caras de las celdas. En la Figura 5 se indican todas las características de este mallado escalonado.



**Figura 4. Mallado escalonado. Los puntos negros son los nodos centrales, los puntos en azul son las velocidades  $v$ , y los puntos rojos son las velocidades  $u$ . ( $u$  en dirección  $x$  y  $v$  en la dirección  $y$ ).**

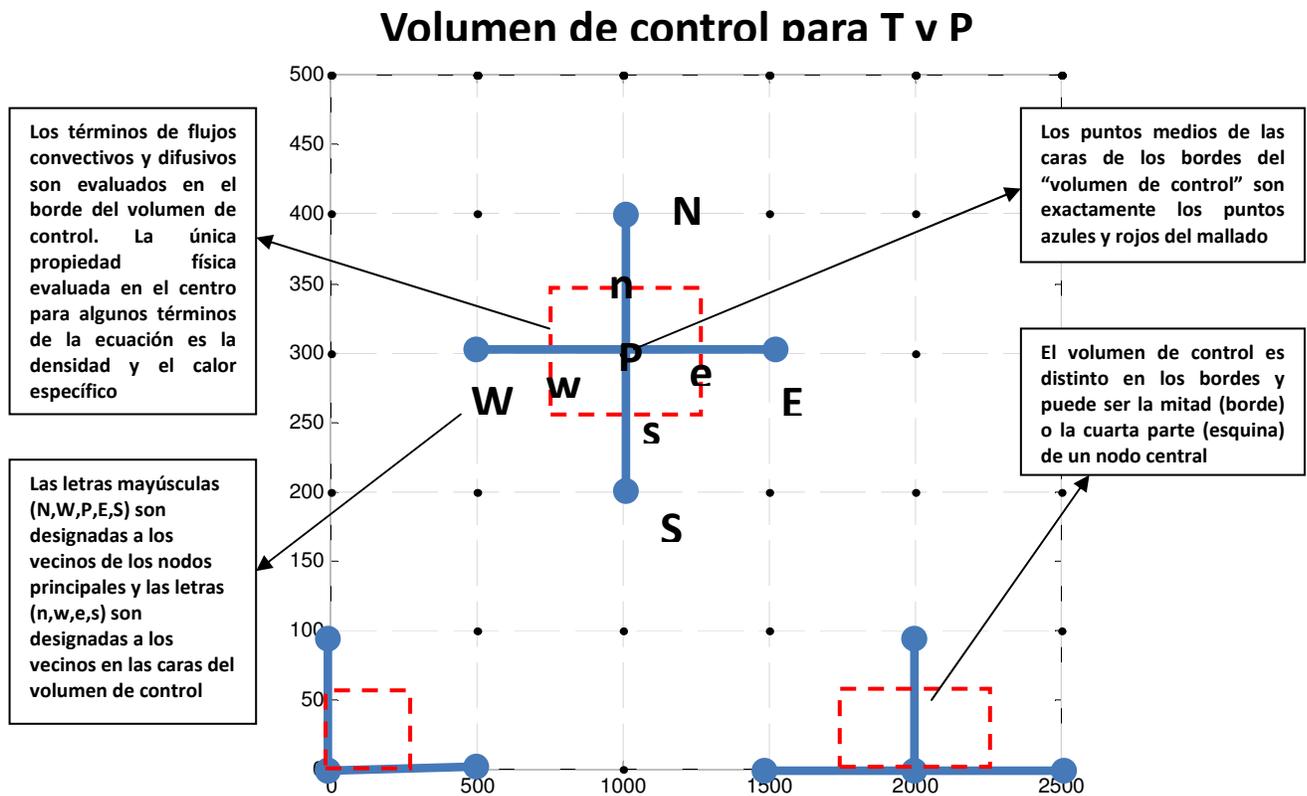


Figura 5. Mallado escalonado, modificado de la versión original de Versteeg *et. al* (1995).

En la Figura6, se muestra un ejemplo de cómo se toman diferentes propiedades y de diferentes matrices para armar una sola matriz escalonada.

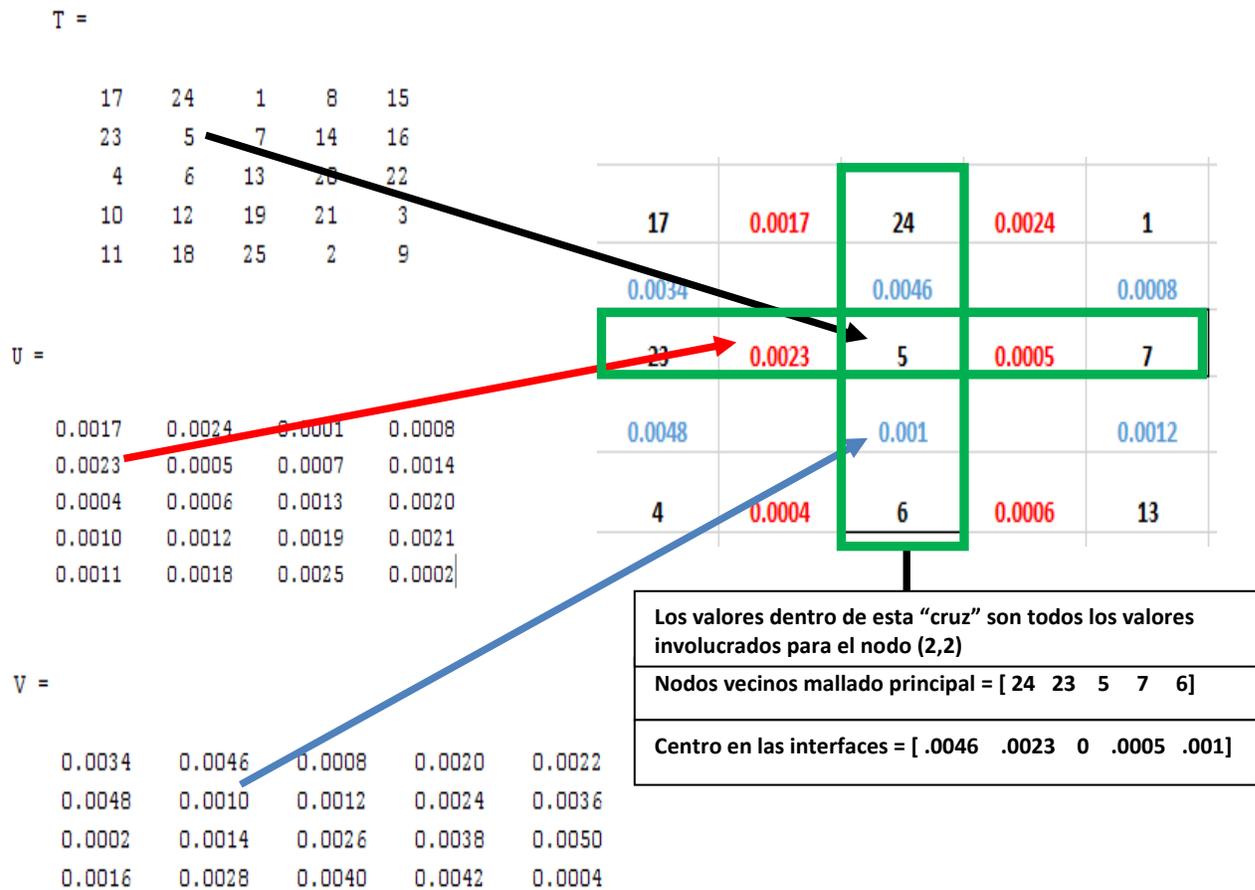


Figura6. Ejemplo del mallasado escalonado para la matriz de Temperatura, de velocidades u, y velocidades v.

La discretización básica de las ecuaciones de momento siguen el mismo esquema explicado anteriormente. Recordando del Capítulo 2, la ecuación de momento en X:

$$\frac{\partial(\rho u)}{\partial t} + \nabla \cdot (\rho \mathbf{u}u) = -\frac{\partial p}{\partial x} + \nabla \cdot (\mu \nabla u) + \rho g_x + \Phi_x. \tag{3}$$

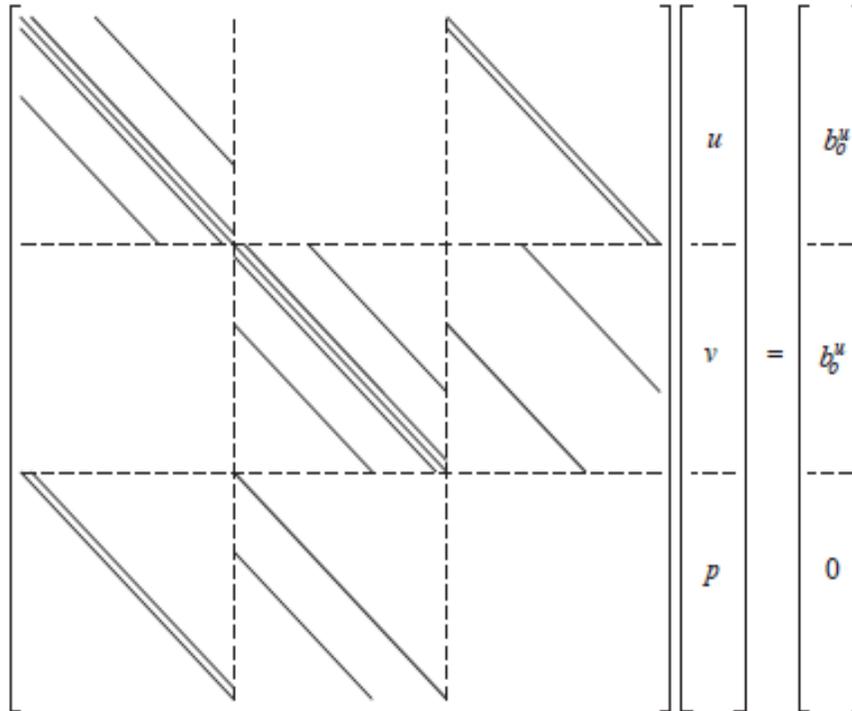
Al aplicarle la forma de discretización que se explicó en la Sección 3.4, resultan las ecuaciones 39 y 40.

$$a_P u_P - a_N u_N - a_W u_W - a_E u_E - a_S u_S = \frac{p_{i-1,j} - p_{i,j}}{\partial x} + b_0. \tag{39}$$

$$a_P v_P - a_N v_N - a_W v_W - a_E v_E - a_S v_S = \frac{p_{i-1,j} - p_{i,j}}{\partial y} + b_0. \tag{40}$$

La Ecuación 39 representa la ecuación del momento en X. Para el momento en Y se hace la misma representación pero reemplazando las velocidades  $u$  por las velocidades  $v$ . Los términos  $b_0$ , representan efectos de gravedad, o efectos de viscosidad.

La Figura7, representa como queda el sistema de ecuaciones acoplados para el mallado escalonado.



**Figura 7. Sistema de ecuaciones para el mallado escalonado.  $P$  representa las presiones,  $u$  representa las velocidades en  $x$ ,  $v$  las velocidades en  $y$ ,  $b_0$  efectos de gravedad o viscosidad. Las líneas en la matriz representan valores que no son cero.**

Tras agrupar los términos de forma correspondiente se obtendría un sistema de ecuaciones de la forma  $Ax=b$  donde,  $x$  sería el vector de velocidades, y presiones.

El algoritmo SIMPLE en consecuencia es como se explica a continuación:

- 1) Asumir un campo de presiones.
- 2) Resolver las ecuaciones discretizadas de momento (39 y 40), con esos valores de presiones para obtener  $u^*$  y  $v^*$ .
- 3) Definir  $p'$  como la diferencia entre el campo de presiones real y el campo de presiones asumido  $p^*$ , de tal forma que  $p=p^*+p'$
- 4) De forma análoga se definen las velocidades  $u'$  y  $v'$ , obteniendo  $u=u^*+u'$ ;  $v=v^*+v'$

- 5) Se resuelven las formulas corregidas para el momento, propuestas por Patankar y Spalding (1980):

$$u'_{ij} = d_{i,j}(p'_{I-1,J} - p'_{I,J}), \quad (41)$$

$$v'_{ij} = d_{i,j}(p'_{I,J-1} - p'_{I,J}). \quad (42)$$

donde  $d_{i,j} = \frac{A_{i,j}}{a_{i,j}}$  y  $d_{i,j} = \frac{A_{i,j}}{a_{i,j}}$ .

- 6) Finalmente se obtienen las velocidades reales mediante 43 y 44

$$u'_{ij} = u^* + d_{i,j}(p'_{I-1,J} - p'_{I,J}), \quad (43)$$

$$v'_{ij} = v^* + d_{i,j}(p'_{I,J-1} - p'_{I,J}). \quad (44)$$

- 7) Repetir el proceso para cada nodo

Los coeficientes mantienen el mismo criterio explicado en la Tabla 2 y se representan de la siguiente manera:

**Tabla4. Coeficiente de discretización para las ecuaciones de momento.**

$a_{i+1,j}$	$a_{i-1,j}$	$a_{i,j+1}$	$a_{i,j-1}$	$b'_{i,j}$
$(\rho dA)_{i+1,j}$	$(\rho dA)_{i,j}$	$(\rho dA)_{i,j+1}$	$(\rho dA)_{i,j}$	$(\rho u^* A)_{i,j} - (\rho u^* A)_{i+1,j} + (\rho v^* A)_{i,j} - (\rho v^* A)_{i,j+1}$

**Nota:** En la sección de apéndices, Figura47, se encuentra el diagrama de flujo del algoritmo SIMPLE.

### 3.7 Convergencia del Algoritmo SIMPLE:

Como se explicó en el Capítulo 1, la convergencia es la propiedad de un método numérico para llegar a una solución la cual aproxime a la solución exacta, es por este motivo que es de gran importancia en los procesos de simulación. El problema radica en el criterio de convergencia a elegir y el tipo de problema que se va a resolver,

mientras más se aproxime el método numérico a la solución exacta más esfuerzo computacional se requerirá.

Explicando esto con un ejemplo práctico: se pueden correr 2 simulaciones y con criterios de convergencia una de 100 iteraciones la otra de 500 para cada etapa de tiempo y que la primera tarde 30 minutos y la segunda 2 horas y media; al comparar los resultados se puede observar que la diferencia es del orden de decimales.

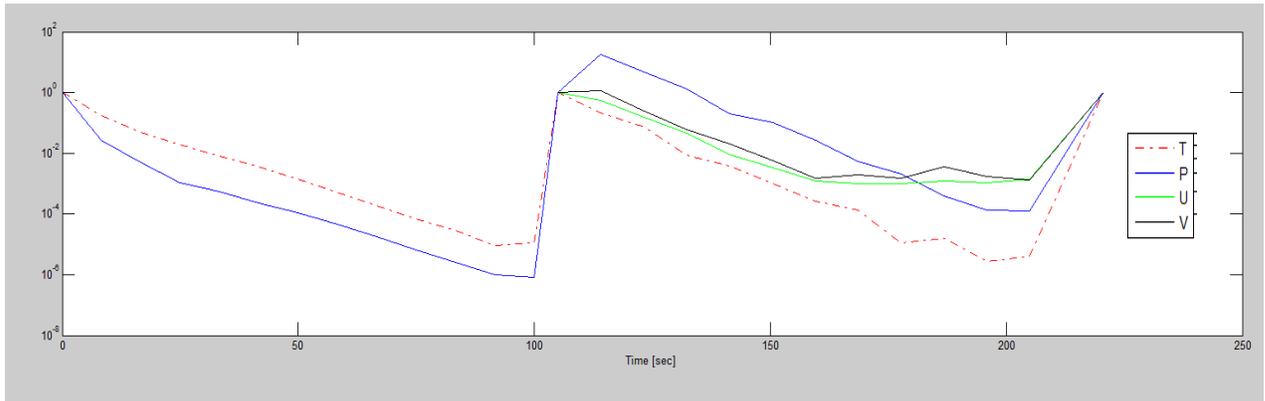
En resumen, puede decirse que el criterio de convergencia puede elegirse en función del grado de exactitud que se necesite para resolver el problema particular.

Los residuales miden que tan bien los valores de presión, velocidad y temperatura de la iteración anterior, satisfacen las ecuaciones basados en los coeficientes de la siguiente iteración. Es decir, los residuales son indicadores claves que permiten determinar si se alcanzará la convergencia al resolver el sistema de ecuaciones. En la Figura8 se muestra su representación gráfica y en la Tabla5 se indican cómo son calculados.

**Tabla. 5. Ecuaciones de residuales.**

Residuales de velocidad en x	Residuales de velocidad en y	Residuales de presión	Residuales de temperatura
$\sigma_u = \frac{ u^* - u }{ u^* }$	$\sigma_v = \frac{ v^* - v }{ v^* }$	$\sigma_p = \frac{ p^* - p }{ v^* }$	$\sigma_T = \frac{ T^* - T }{ T^* }$

En este Capítulo se discretizaron las ecuaciones de conservación y momento, reescribiéndolas en forma de coeficientes para que sea más fácil su manejo y entendimiento. Se analizaron varios esquemas de solución como el de diferencias centradas el cual no es muy recomendado para números de Peclet mayores a 2. Esto se debe a la naturaleza no lineal de las ecuaciones y la gran inestabilidad que involucra la solución del sistema de ecuaciones.



**Figura 8. Representación de los residuales por iteración para varios intervalos de tiempo. El eje Y representa el valor del residual, mientras que el eje X representa el tiempo transcurrido. Cada curva representa un residual, la roja es de temperatura, la azul es de presión, la verde de velocidad horizontal y la negra de velocidad vertical.**

Los otros esquemas pueden ayudar a disminuir la inestabilidad y lograr alcanzar la convergencia, sin embargo, definir bien las condiciones de fronteras, las entradas, salidas, fuentes, y propiedades en sí del yacimiento son claves para poder obtener buenos resultados. Una buena forma de observar si los resultados están siendo congruentes y van por buen camino es a través de los residuales.

## Capítulo 4. Funcionamiento del programa y Problemas

El software desarrollado en este trabajo de tesis se programó con Matlab, versión R2013a, debido a las librerías matemáticas y facilidad del lenguaje de programación, además de la capacidad gráfica que posee.

### 4.1 Estructura del simulador

El simulador de yacimientos geotérmicos cuenta con 3 módulos y cada módulo tiene un conjunto de rutinas y subrutinas las cuales permiten su funcionamiento. En las Figuras 9, 10 y 11, se indica el diagrama de flujo general del programa.

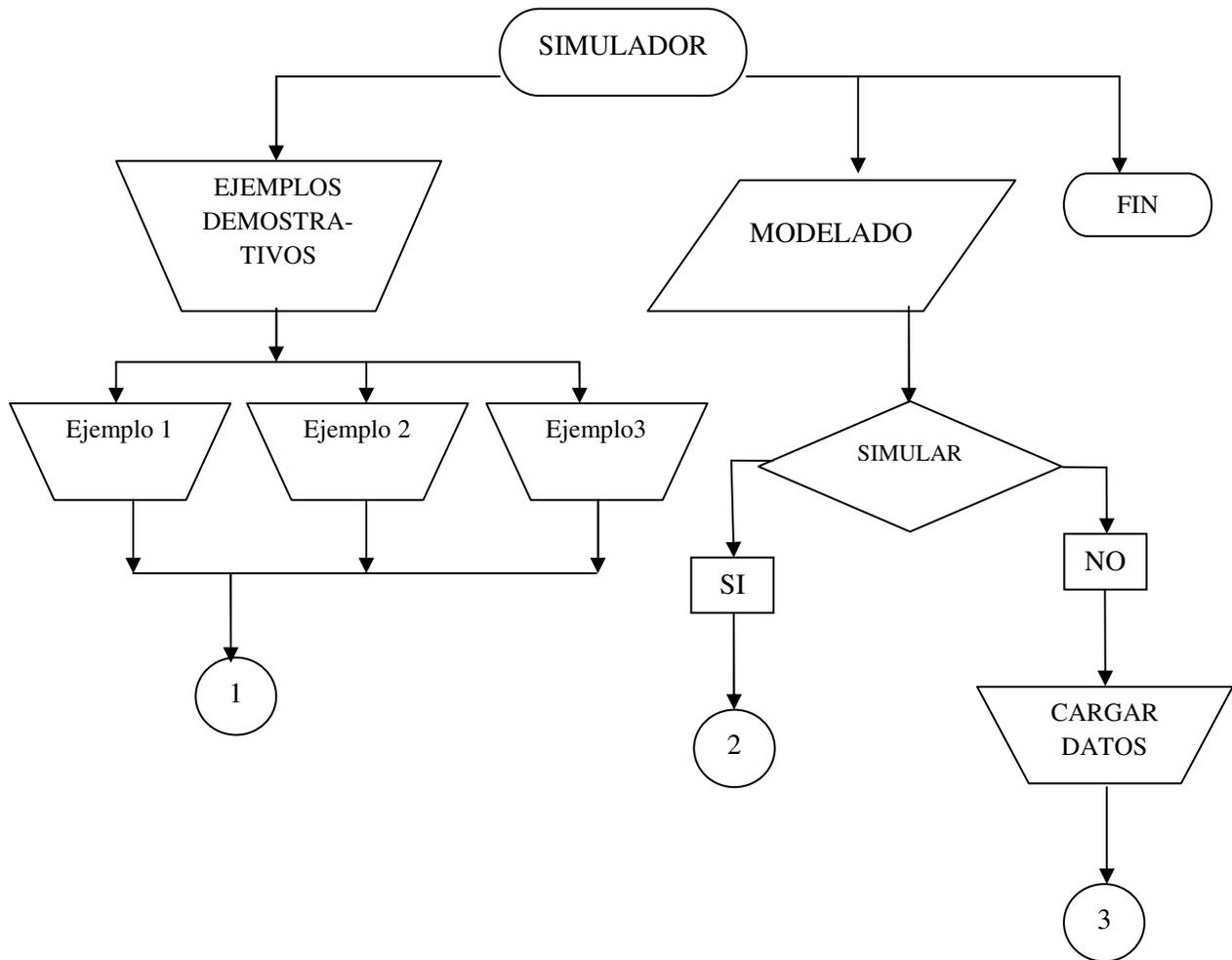


Figura 9. Diagrama de flujo general del simulador.

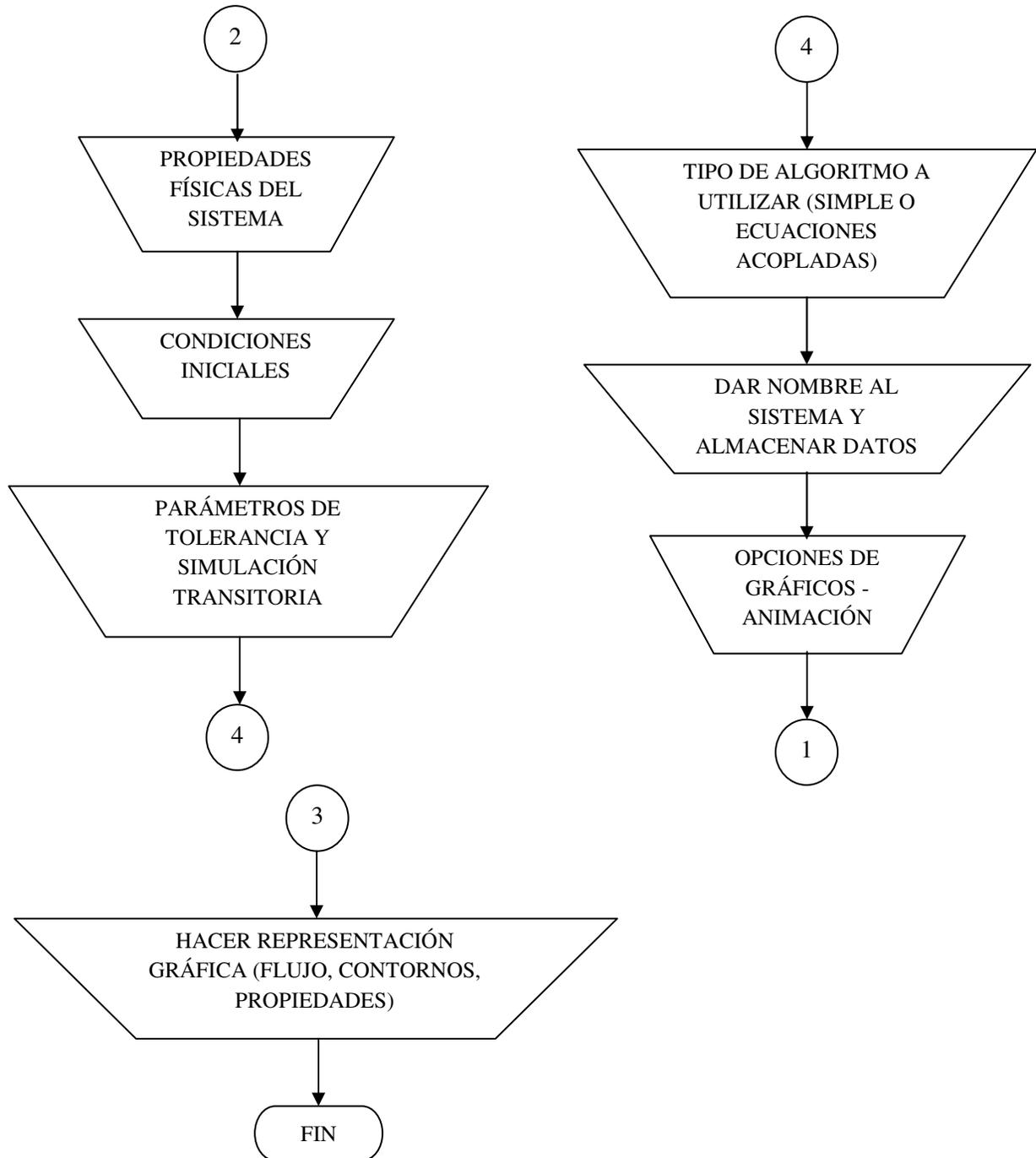


Figura 10. Diagrama de flujo general del simulador.

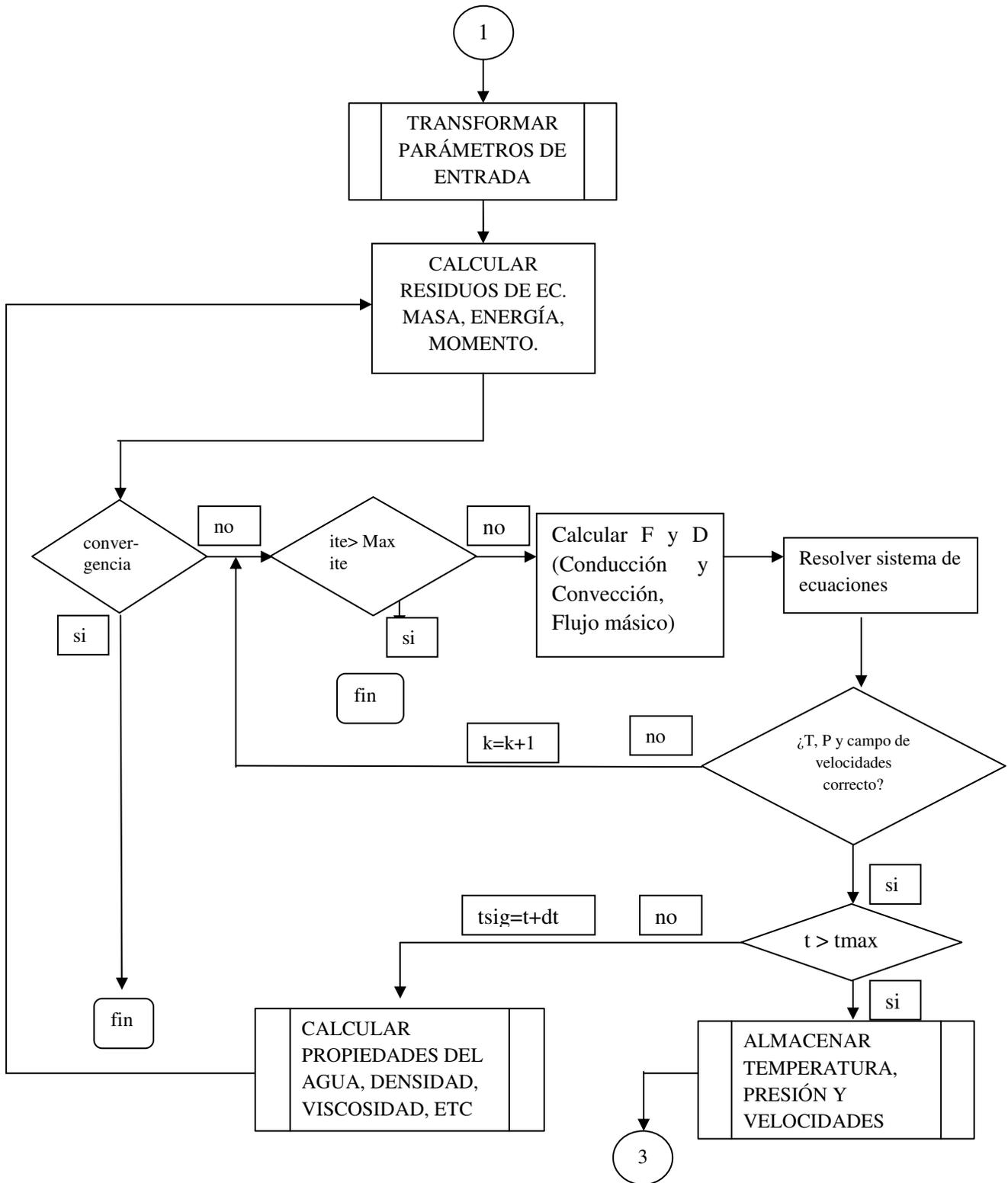


Figura 11. Diagrama de flujo general del simulador. Solución numérica.

### 4.1.1 Módulo principal

Contiene los archivos de arranque o de inicialización del programa, los cuales se encargan de llamar a las demás subrutinas para los cálculos de la simulación.

Es en este módulo donde se colocan las entradas del programa, opciones de gráficos, opciones de interpolación, también se encuentran los archivos encargados de guardar y cargar los datos de simulaciones y de realizar los gráficos.

#### 4.1.1.1 Archivo main.m

El archivo **main.m** contiene todos los parámetros modificables. Mientras no se altere la estructura central del archivo, se puede crear tantos "main.m" con nombres y parámetros distintos como el usuario lo desee. Por ejemplo, cada ejercicio propuesto en este trabajo posee un archivo main, y fueron renombrados como ejemplo 1, ejemplo 2 y ejemplo 3, respectivamente. Los resultados de éstas simulaciones se encuentran en el Capítulo 5.

Este archivo se ubica en el módulo principal, y éste controlará las opciones del programa, es decir, en **main.m** se elige si se graficarán los resultados, que tipos de gráficos, o si se desea almacenar el proyecto, etc.

A continuación se presenta un ejemplo del archivo **main.m**.

```
% constantes fisicas del problema-----
T0 = 20;      % temperatura inicial [°C]
Tt = 20;      % temperatura en el borde superior [°C]
Pt = 2e6;     % presion en el borde superior [Pa]
nx = 20;      % numero de celdas en "x"
ny = 20;      % numero de celdas en "y"
dx = 100;     % espaciamiento en "x" [metros]
dy = 50;      % espaciamiento en "y" [metros]
dz = 500;     % longitud en "z" [metros]

dt = 100;     % espaciamiento de tiempo en [sec] (este valor incrementa en
escala lineal automáticamente)
Q0 = 60e6;    % valor de la fuente de calor en unidades Watts

ind_q0= [9 10 11 ]; % posiciones donde hay fuente de calor (en el fondo del
yacimientto)

kFactor=3.5;  % factor de correccion para la conductividad del yacimiento
%0.6 conduct del agua
```

```

% preparar los datos-----
[U0,V0,P0,T0,constants]=PrepareInputData(T0,Tt,Pt,nx,ny,dx,dy,dz,dt,Q0,ind_q0,
kFactor);

% opciones de calculo-----

options.InterpMethod = 'spline'; % metodo de interpolacion
{'spline'}/'linear'
options.Scheme       = 'upwind'; % esquema de diferencias
{'upwind'}/'centred'/'exp'/'hibrid'/'power'
options.MaxIt       = 500;      % # de iteraciones maxima en el algoritmo
por cada tiempo {500}
options.TolT        = .01;      % tolerancia maxima en T {.00001}
options.TolP        = .1;       % tolerancia maxima en P {.00001}
options.TolU        = .01;      % tolerancia maxima en U {.00001}
options.TolV        = .01;      % tolerancia maxima en V {.00001}
options.TimeMax     = 200*3600*24; % tiempo maximo de simulacion
options.SteadyState = 10;       % factor multiplicado por la tolerancia
% para detener los calculos antes de
% llegar al estado estacionario

% algoritmo CFD a utilizar-----

algorithm           = 'coupled'; % algoritmo de CFD a utilizar
{'coupled'}/'SIMPLE'

% ejecutar simulacion-----

[U,V,P,T,t]=cfdsolve(T0,P0,U0,V0,constants,options,algorithm);

% guardar datos de salida -----

savemat;

% opciones para la animacion -----

PlotOptions.StaticColorbar = false;          % barra de colores estatica
false/true
PlotOptions.StreamPlot     = 'stream'; % plot de linea de flujo
'streamslice'/'quiver'
PlotOptions.FrameSpeed     = .01;  %.5      % numero de segundos que dura un
solo frame
PlotOptions.MaxTime        = 100;        % porcentaje del tiempo maximo
usado para la
% simulacion (en porcentaje de
% 0 a 100)

plotdata(U,V,T,P,t,'stream',constants,PlotOptions)
%tipos de gráficos 'stream', 'countour', 'properties', 'transient'
%estacionario

```

#### 4.1.1.2 Función PrepareInputData()

Lee los datos de entrada escritos en "main.m" y genera las matrices con los datos iniciales necesarios para comenzar con la simulación. Se encuentra ubicado en el

módulo principal y este archivo no debe ser modificado para garantizar el correcto funcionamiento del programa.

#### 4.1.1.3 Funciones `savemat()` / `loadmat()`

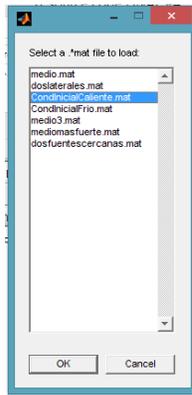


Figura 12. Interfase `loadmat`

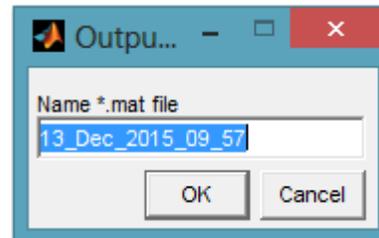


Figura 13. Interfase `savemat`

**savemat:** guarda los datos almacenados en el espacio de trabajo (workspace) de matlab, dentro de la carpeta `.../outputdata/`

El nombre del archivo a guardar es editado por el usuario. Por defecto, este nombre es la fecha y la hora a la cual se generó los datos.

**loadmat:** carga los datos almacenados dentro de la carpeta `.../outputdata/` directo al espacio de trabajo (workspace) de matlab.

Se presenta una lista de los archivos de datos `*.mat` contenidos dentro de esta carpeta, y el usuario escoge el archivo que desea cargar. El archivo `recent.mat`, si es que existe, representa el archivo con los últimos datos generados independientemente si el usuario le asigno o no un nombre para los mismos.

#### 4.1.1.4 Función `plotdata()`

Esta función sirve para generar distintos gráficos y puede utilizarse de diferentes maneras. Se encuentra ubicada en el módulo principal. Y utiliza subrutinas de gráficos como **`gridshow.m`**

(1) Crear una animación con perfil de temperatura y patrón de flujo simultáneamente

**plotdata(U,V,T,P,t, 'stream' ,constants,PlotOptions).**

La estructura `PlotOptions` contiene los siguientes datos y se encuentra incluido en "main.m":

```
PlotOptions.StaticColorbar = false;           % barra de colores estatica
false/true
PlotOptions.StreamPlot     = 'streamslice';  % plot de linea de flujo
'sreamslice'/'quiver'
PlotOptions.FrameSpeed     = .01;   %.5      % numero de segundos que dura un
solo frame
PlotOptions.MaxTime        = 100;        % porcentaje del tiempo maximo
usado para la
% simulacion (en porcentaje de
% 0 a 100)
```

Las demás variables son el resultado de la simulación y pueden ser cargadas directamente con la función `loadmat`.

(2) Generar una ventana de perfiles de presión y temperatura en cualquier instante de tiempo elegido:

**plotdata(U,V,T,P,t, 'contour' ,constants,Time).**

El parámetro **Time** es el tiempo en segundos al cual se quieren realizar los gráficos de contorno. Ejemplo, si se desea graficar la presión y temperatura a los dos días del tiempo de inicio de la simulación, el parámetro **Time** debe ser igual a 172800 (segundos).

(3) Generar una ventana con las magnitudes de presión, temperatura y velocidades en  $X$  y  $Y$ , velocidades  $u$  y  $v$  respectivamente en todo el tiempo de la simulación.

**plotdata(U,V,T,P,t,'transient',constants).**

(4) Generar ventanas con las siguientes propiedades de estado: conductividad térmica, viscosidad y densidad.

**plotdata(U,V,T,P,t,'properties',constants,Time).**

Igual que para el caso (2), el parámetro **Time**, representa el tiempo en segundos a la cual se desean obtener las propiedades de estado.

#### 4.1.1.5 Función `gridshow()`

Se encuentra en el módulo de principal. Es una subrutina necesaria para poder graficar el mallado escalonado.

#### 4.1.1.6 Función `cfdsolve()`

Esta función sirve para resolver el problema DCF a partir de las matrices iniciales  $T_0, P_0, U_0, V_0$  en  $t = 0$ . Retorna las matrices  $(T, P, U, V)$  en todos los instantes de tiempo (es decir que son matrices de tres dimensiones)

**`[U,V,P,T,t]=cfdsolve(T0,P0,U0,V0,constants,options,algorithm)`**

La estructura "options" tiene las siguientes propiedades:

```
options.InterpMethod = 'spline'; % metodo de interpolacion
{'spline'}/'linear'
options.Scheme       = 'upwind'; % esquema de diferencias
{'upwind'}/'centred'/'exp'/'hibrid'/'power'
options.MaxIt       = 500;      % # de iteraciones maxima en el algoritmo
por cada tiempo {500}
options.TolT        = .001;    % tolerancia maxima en T {.00001}
options.TolP        = .01;    % tolerancia maxima en P {.00001}
options.TolU        = .001;    % tolerancia maxima en U {.00001}
options.TolV        = .001;    % tolerancia maxima en V {.00001}
options.TimeMax     = 3600*24*300; % tiempo maximo de simulacion 3600
segundos * 24 horas * 300 días
options.SteadyState = 10;      % factor multiplicado por la tolerancia
% para detener los calculos antes de
% llegar al estado estacionario
```

`options.InterpMethod` = método de interpolación para encontrar los flujos difusivos y másicos en la matriz escalonada. Puede escogerse entre interpolación cúbica, spline e interpolación lineal.

El recomendado es interpolación spline, pues ayuda a evitar oscilaciones indeseables al interpolar entre las intercaras o nodos.

`options.Scheme` = método de diferencias para el término convectivo en las ecuaciones diferenciales de transporte. (**Ver Capítulo 3, Tabla 1**)

options.ToIT = tolerancia de las variables definida como se indicó en el **Capítulo 3 en la Tabla 5**

options.TimeMax = tiempo máximo de simulación. Después de este valor de tiempo, simplemente la simulación se detiene sin llegar al estado estacionario.

Debido a que los tiempos de estabilización en un yacimiento geotérmico son muy grandes, es decir, puede requerir cientos de días para que estabilice y alcance una temperatura y presión estable, se recomienda dejar estos tiempos amplios de orden de  $1 \times 10^7$  segundos, a menos que se tenga otro propósito en la simulación.

options.SteadyState = valor de tolerancia del estado estacionario. Debido a que teóricamente no se puede llegar al estado estacionario computacionalmente al resolver una ecuación diferencial, se define una banda de  $S$  veces la tolerancia:

$$\text{SteadyState} = S \times \frac{\max\{|x_{ij}^{k+1} - x_{ij}^k|\}}{\max\{|x_{ij}^k|\}}$$

#### 4.1.2 Módulo de algoritmos

Este es el módulo central del simulador, y es donde se encuentran todas las funciones que realizan los cálculos para el correcto funcionamiento del programa, estos archivos ya no son modificables y cualquier modificación que se les haga, puede causar que el programa no funcione correctamente. Corresponde a la Figura 11 del diagrama de flujo.

##### 4.1.2.1 Función aneighbor()

Es la función que se encarga de crear los coeficientes  $a_{i,j}$ , descritos en el Capítulo 3, de las ecuaciones de transporte en las intercaras para cada nodo.

##### 4.1.2.2 Función neighboring()

Se encarga de crear vectores con los datos de los nodos vecinos al nodo central, es decir, crea un vector que tiene las temperaturas, otro con las velocidades  $u$  y  $v$ , otro

de presiones, para de esta forma poder usar la matriz escalonada explicada en el Capítulo 3. En la sección de apéndices se muestra un ejemplo del uso de esta función.

#### 4.1.2.3 Función Flux()

Esta función calcula los flujos difusivos y convectivos de la ecuación de transporte de cantidad de movimiento y de energía.

Tiene de entrada las matrices de temperatura, presión, velocidad  $u$  y velocidad  $v$ , y las constantes,  $dt$ ,  $dx$ ,  $dy$ , etc.. Da como salida los términos  $D$  en  $X$  y en  $Y$ ,  $F$  en  $X$  y en  $Y$ , densidad y calor específico.

#### 4.1.2.4 Función SolveSystem()

Se encarga de crear los sistemas de ecuaciones  $Ax=B$  y de resolverlos. Se apoya en las funciones `aneighbor`, `neighboring` y `flux` para poder hacer este proceso.

También calcula los residuos y grafica los resultados de la convergencia de cada iteración.

### 4.1.3 Módulo de propiedades del agua

Éste módulo consta de una serie de funciones que fueron desarrolladas por la Asociación Internacional para las Propiedades del Agua y el Vapor (IAPWS, 2007), para el cálculo de las propiedades de estado del agua líquida y vapor. La función principal de éste módulo es **IAPWS\_IF97.m**. El uso de éste módulo se encuentra en la Figura 11 del diagrama de flujo.

#### 4.1.3.1 Función IAPWS\_IF97()

Permite calcular, densidad, entalpía, viscosidad, calor específico, volumen específico, calidad y conductividad térmica, en función de las variables de estado presión y temperatura.

#### 4.1.3.2 Función ComputeProperties()

Es una función de simplificación creada para calcular en cada nodo o intercara, las propiedades indicadas en la función `IAPWS_IF97`.

## Capítulo 5. Validación y Problemas propuestos

Para comprobar la validez de los resultados obtenidos por éste simulador desarrollado en este trabajo de tesis, se decidió utilizar el modelo de caja propuesto por M.L. Sorey (1978), tal como se muestra en la Figura 14. Todos los lados son impermeables y además los lados verticales están aislados. Una fuente de calor es aplicada en el centro del yacimiento, para asegurar la convergencia el fluido es homogéneo en una sola fase, las dimensiones de la longitud del yacimiento deben ser el doble de las dimensiones de la altura.

Este es un modelo teórico, cuyos valores de fuente de calor y dimensiones totales, pueden ser modificadas, siempre y cuando se respeten las condiciones de flujo laminar, y  $h_x$  debe ser dos veces  $h_z$  ( $h_x=2h_z$ ). Es un modelo básico que ya fue resuelto antes por otros autores como Beck en el 1972 y del cual se posee suficiente información para reproducirlo.

Sorey, esperaba obtener un resultado de dos celdas convectivas tal como se muestra en la Figura 15 y una distribución de temperatura simétrica tal como se muestra en la Figura 16.

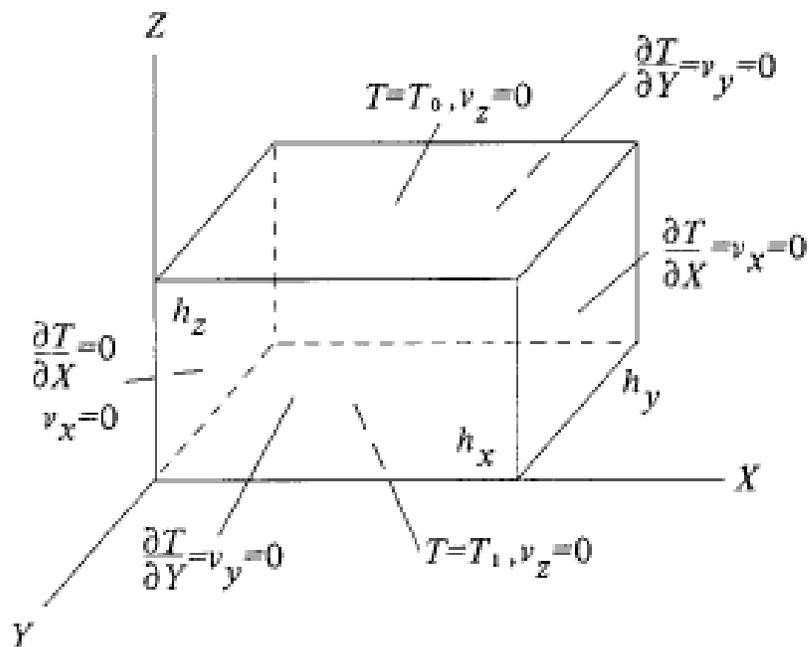


Figura 14. Condiciones de frontera del modelo de Caja propuesto por M.L. Sorey (1978).

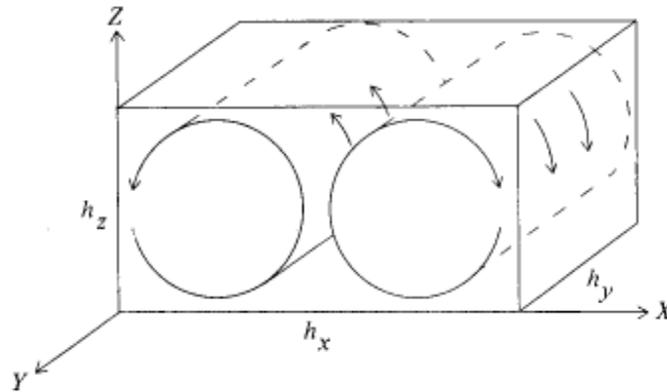


Figura 15. Resultados esperados del modelo de Caja propuesto por M.L. Sorey (1978),  $h_x=2h_z$ .

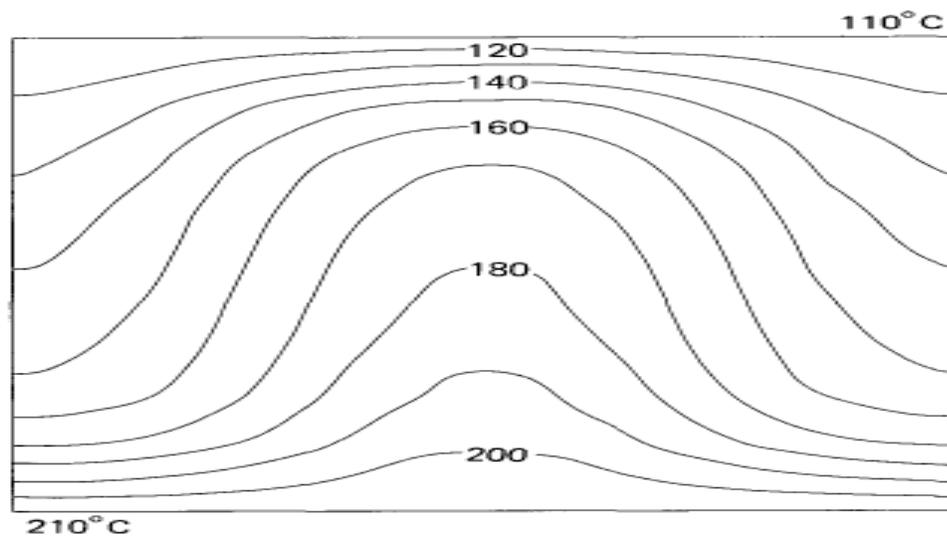


Figura 16. Resultados obtenidos del modelo de Caja propuesto por M.L. Sorey (1978), distribución uniforme de temperaturas, fuente en la mitad de la longitud  $h_x$ .

Los modelos que se presentarán a continuación, fueron corridos en una computadora con procesador Intel Core I7 4700MQ, con 16 GB de memoria RAM DDR-3 y procesador gráfico Nvidia G-Force GTX770M de 3GB DDR-5.

### 5.1 Modelo de caja

Para poder reproducir este ejercicio, primero se debe ajustar los parámetros de entrada. Se creó un sistema de 21 x 21 celdas, con una fuente principal en las celdas 9, 10, 11. Para las demás celdas se crearon pequeñas fuentes de valores cada vez más bajos, de forma tal que se cree el efecto de una fuente de calor de efecto radial pero que actúe a lo largo de toda la base del yacimiento. El  $dx$  elegido fue de 100 metros y el

dy de 50 metros. La fuente de calor es una fuente central de 90 MWatts y cuya presencia disminuye a lo largo del yacimiento para crear un efecto de fuente radial.

En este sistema se considera el efecto de la gravedad y se supone en todo momento que el fluido se mantiene en estado líquido.

Los resultados del simulador de este trabajo de tesis, son como se muestran en las Figuras 17, 18, 19

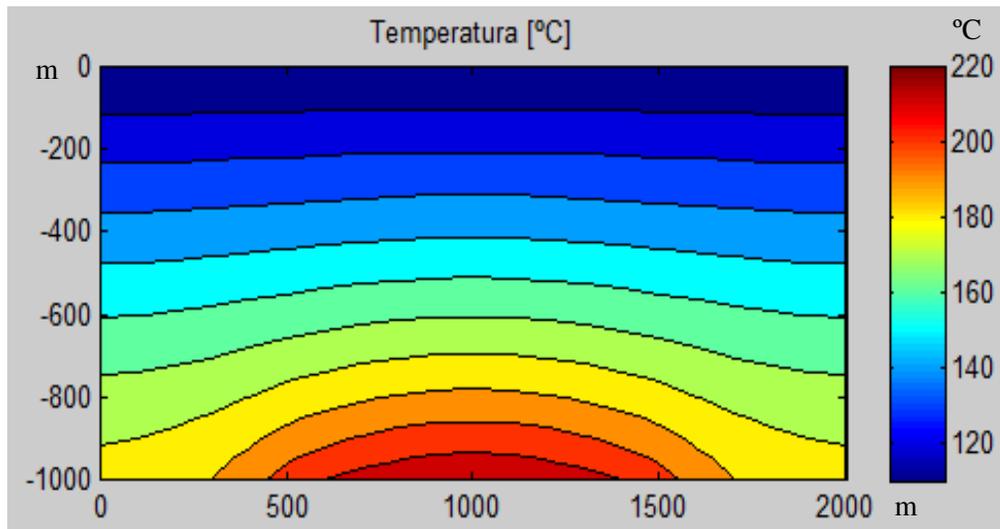


Figura 17. Isothermas del modelo simulado. Temperatura en °C.

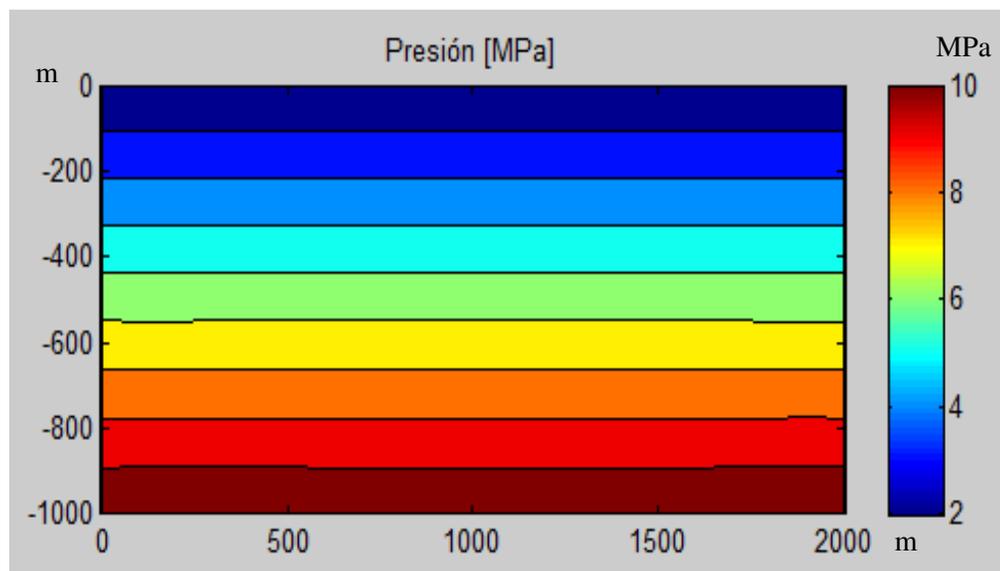
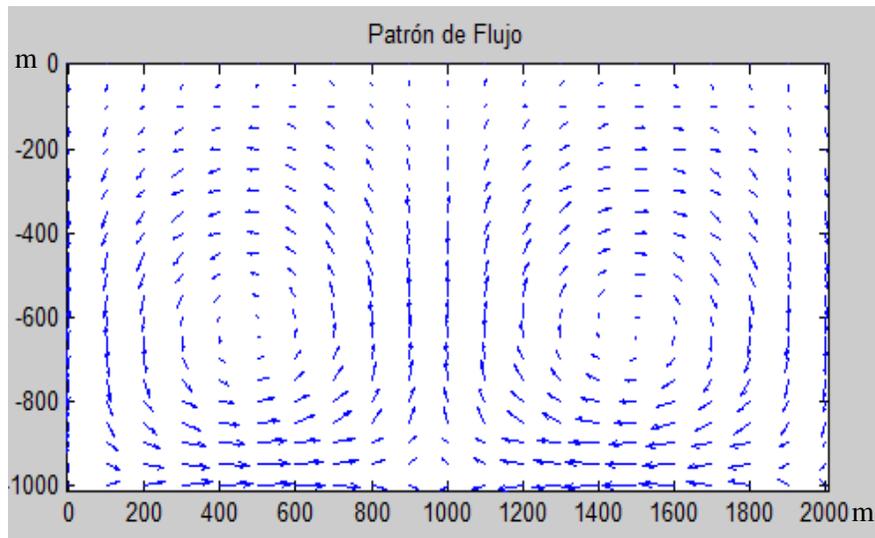
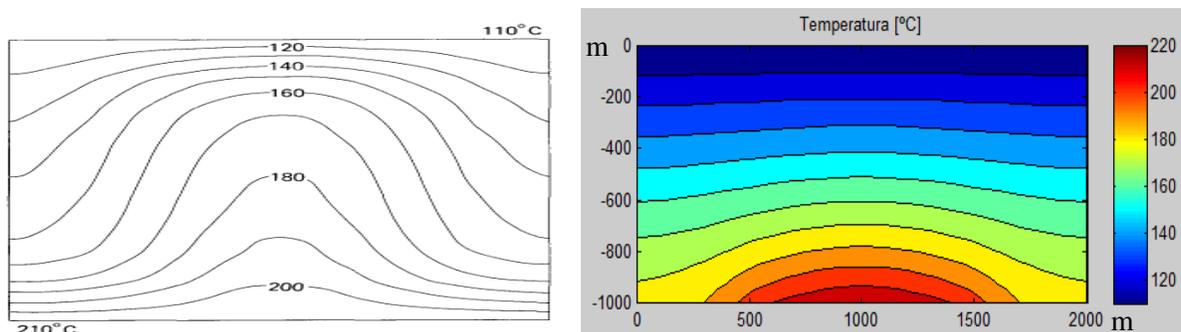


Figura 18. Isobaras del modelo simulado. Presión en MPa.



**Figura 19. Patrón de flujo del modelo simulado.**

En los resultados obtenidos, se puede apreciar cierta similitud con los resultados expuestos por Sorey (1978), tal como se indica en la Figura 20. Sin embargo hay que resaltar que los métodos utilizados por Sorey son diferentes a los utilizados en este trabajo de tesis.



**Figura 20. Comparación del modelo de Sorey contra el modelo generado por el simulador de esta tesis.**

Sorey utiliza Diferencias Finitas Integradas para el desarrollo de las ecuaciones de masa y energía, también utiliza otros métodos de interpolación para poder general las isothermas que no especifica en su publicación para poder generar las isothermas. En este trabajo se utilizan los métodos de interpolación incluidos en las librerías de Matlab y las ecuaciones fueron discretizadas mediante el método de volumen finito explicado en el Capítulo 2. Se puede apreciar que en la interpolación realizada por Sorey, las isothermas se encuentran muy curvadas, esto puede deberse al método de interpolación o la discretización en sí, adicionalmente, el método que utiliza Sorey para resolver las

ecuaciones de masa y energía no es un método de ecuaciones acopladas, es un método donde primero se resuelve la ecuación de masa para  $t_0+\Delta t/2$ , pero la ecuación de energía se resuelve para  $t_0+\Delta t$ , luego se vuelve a resolver la ecuación de masa para el tiempo  $t_0+3\Delta t/2$ , y la de energía para  $t_0+2\Delta t$  y así sucesivamente. M.L. Sorey (1978). Es importante mencionar que la solución obtenida presenta la simetría y patrón de flujo esperados, además de satisfacer las condiciones de frontera del caso sintético.

## **5.2 Modelo de caja (Problema 2).**

A continuación se muestra otro ejercicio de modelo de caja. Las condiciones de frontera son como se indicaron en la Figura 14. Los datos de entrada se muestran en la Tabla 6. La fuente que se utiliza en el ejercicio es del tipo radial incidiendo en los 3 nodos centrales con mayor intensidad.

Los resultados de la simulación se aprecian en las siguientes secuencias de imágenes para diferentes tiempos (Figuras 21-26). Para cada instante de tiempo se realiza una comprobación de convergencia la cual está representada en la Figura 27:

Tabla 6. Entradas para el problema 2.

<b>PROPIEDADES DEL MEDIO</b>		
Densidad (Kg/m <sup>3</sup> )		2600
Porosidad (fracción)		0.1
Permeabilidad (Darcy)		1x10 <sup>-3</sup>
Conductividad Térmica (W/m °C)		3.5
<b>CONDICIONES INICIALES DEL FLUIDO</b>		
	Temperatura ° C	Presión Mpa
Tope	30	2x10 <sup>6</sup>
Fondo	200 (Isotérmico)	controlada por gradiente hidrostático
Inicial en el yacimiento	10	controlada por gradiente hidrostático
<b>ESTADO TRANSITORIO Y TOLERANCIAS</b>		
Tiempo inicial (s)		0
Tiempo final (s)		54432000 (630 días)
Incremento del paso del tiempo		100
Tolerancia de Temperatura		0.01
Tolerancia de Presión		0.1
Tolerancia de velocidad U		0.001
Tolerancia de velocidad V		0.001
<b>FACTORES EXTERNOS</b>		
Fuente de calor (Watts)		70x10 <sup>6</sup> , 90x10 <sup>6</sup> , 90x10 <sup>6</sup> , 90x10 <sup>6</sup> , 70x10 <sup>6</sup>
Nodos de inyección		4, 5, 6, 7, 8
<b>CONDICIONES DE LA SIMULACIÓN</b>		
Tipo de algoritmo	acoplado	(ecuaciones acopladas y resueltas de forma simultánea)
Máximo de iteraciones por paso del tiempo		500
Frontera inferior Isotérmica		Si
Discretización en X		200
Discretización en Y		100
Discretización en Z		500
Cantidad de nodos en X		11
Cantidad de nodos en Y		11
Método de interpolación		Spline
Esquema de diferencias		Upwind

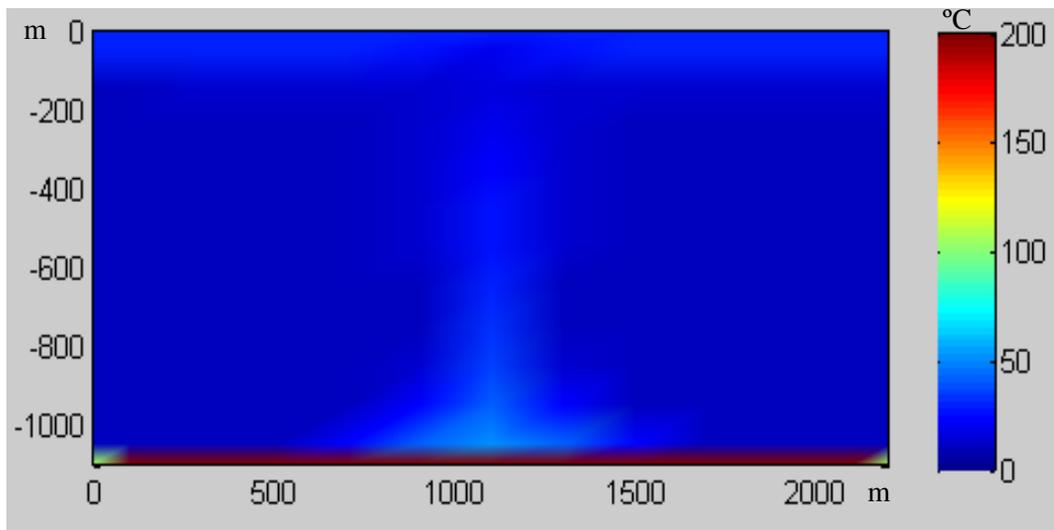


Figura 21. Condiciones de flujo y temperatura iniciales cuando inicia a actuar la fuente de calor.

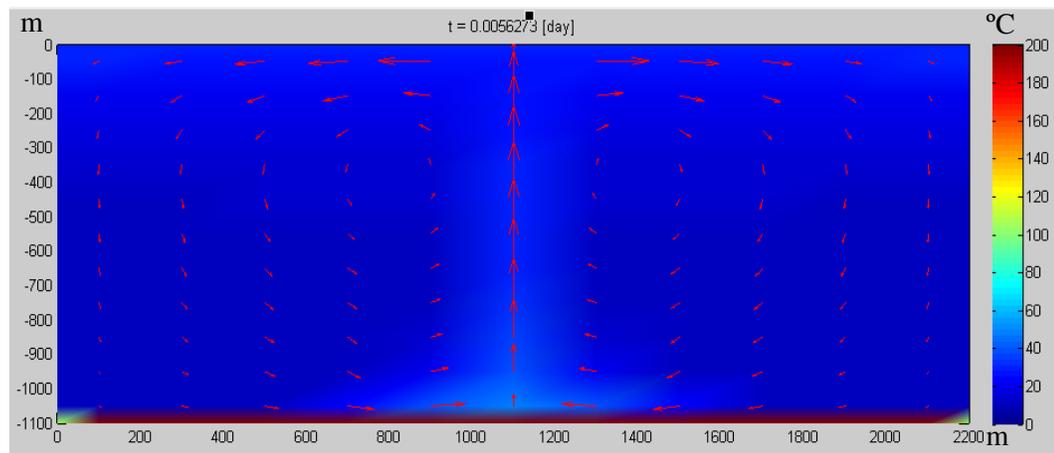


Figura 22. Condiciones de flujo y temperatura a los 0.0056 días.

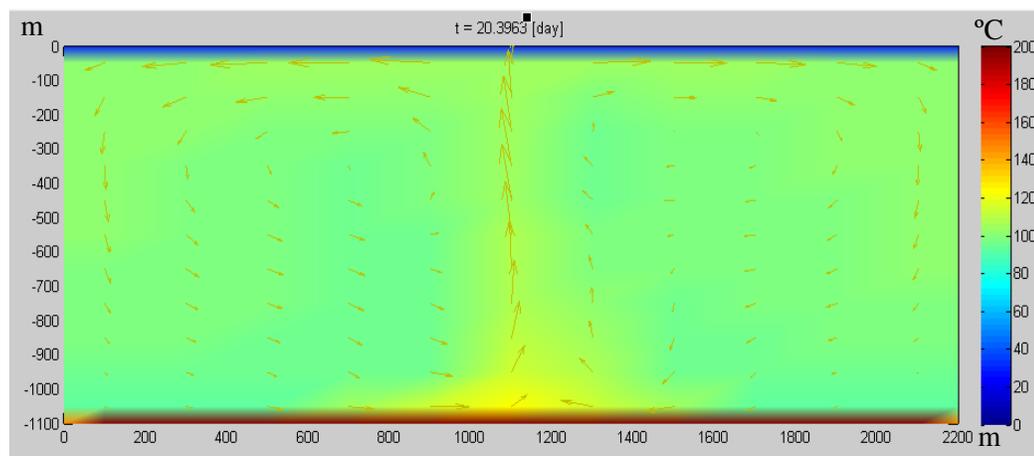


Figura 23. Condiciones de flujo y temperatura a los 20.3963 días.

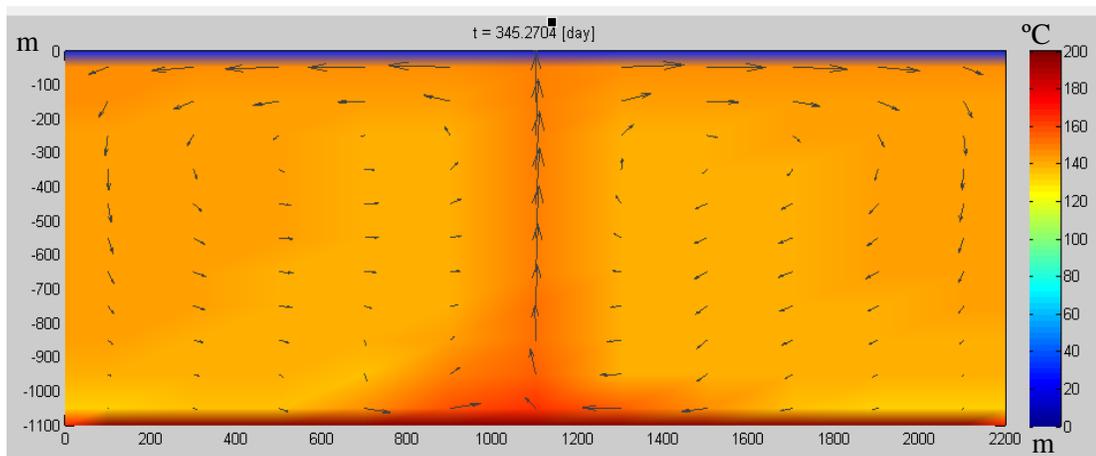


Figura 24. Condiciones de flujo y temperatura a los 345.2704 días.

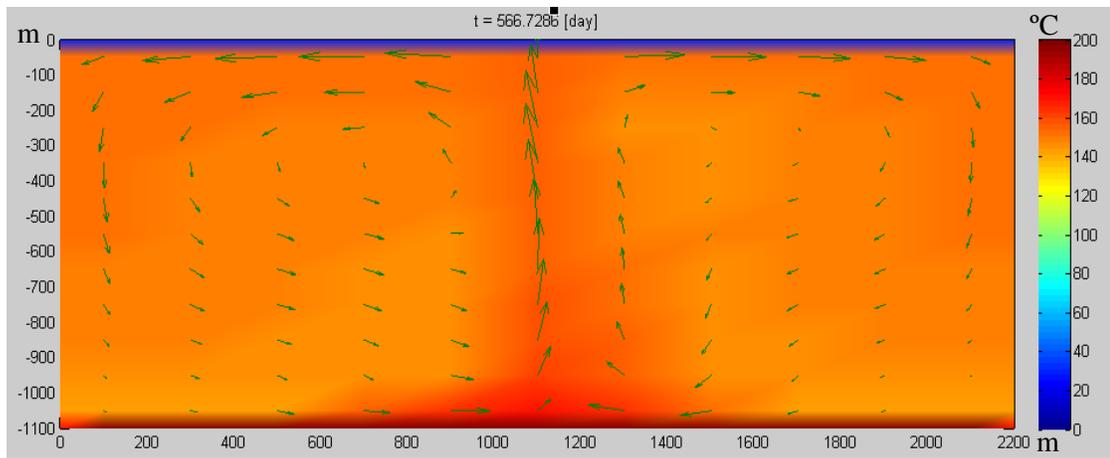


Figura 25. Condiciones de flujo y temperatura a los 566.7286 días.

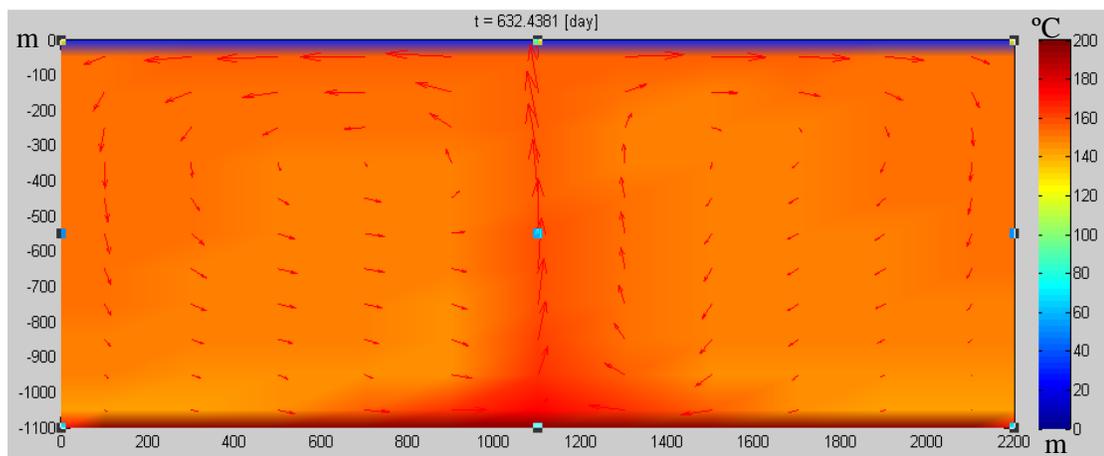
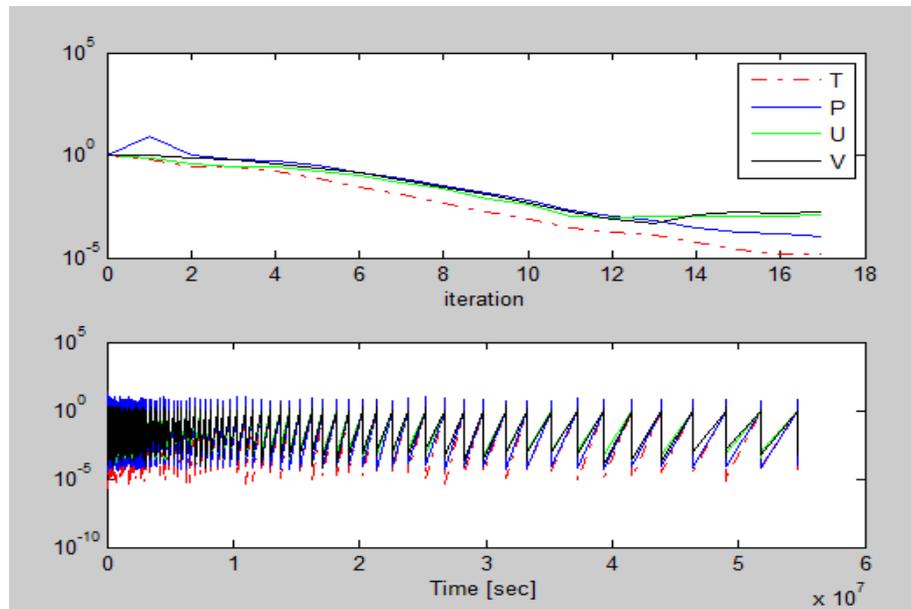
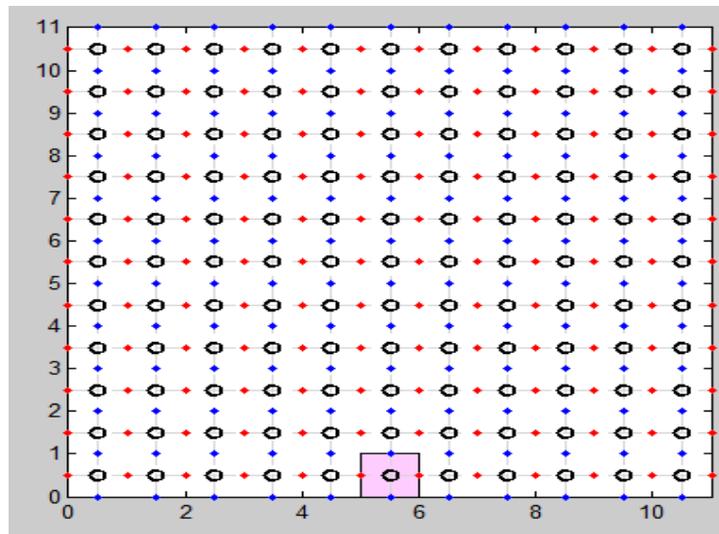


Figura 26. Condiciones de flujo y temperatura a los 632.4381 días.



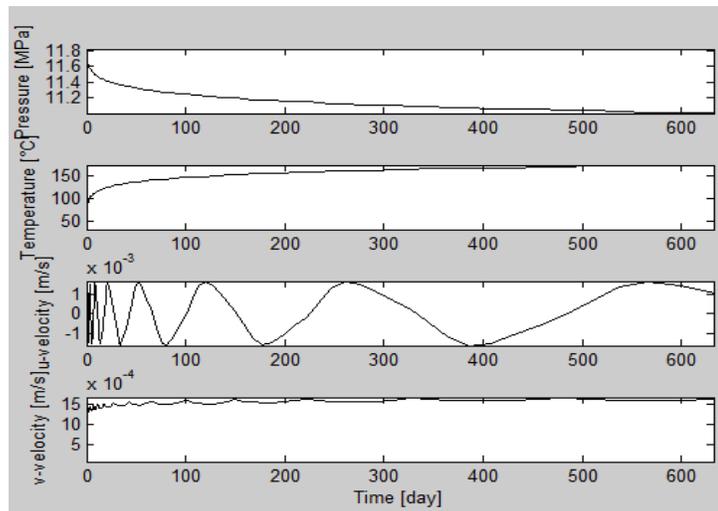
**Figura 27. Gráfico de convergencia.** La parte superior indica la cantidad de iteraciones necesarias en ese tiempo para alcanzar la convergencia. La parte inferior agrupa todas las convergencias alcanzadas a lo largo de la simulación.

La siguiente imagen (Figura28), representa el mallado escalonado durante la simulación y este permite elegir un punto en él para poder observar cómo varían las velocidades, presión y temperatura en ése punto; como se indica en la Figura29.



**Figura 28. Mallado escalonado (el recuadro en color rosa indica el punto seleccionado para representarlo).**

Se observa un comportamiento muy variable en la velocidades, principalmente durante el principio de la simulación; esto puede deberse a la influencias de la fuente, la presencia de 90 MWatts en esa celda genera una turbulencia, pero a medida que pasa el tiempo tiende a estabilizarse.



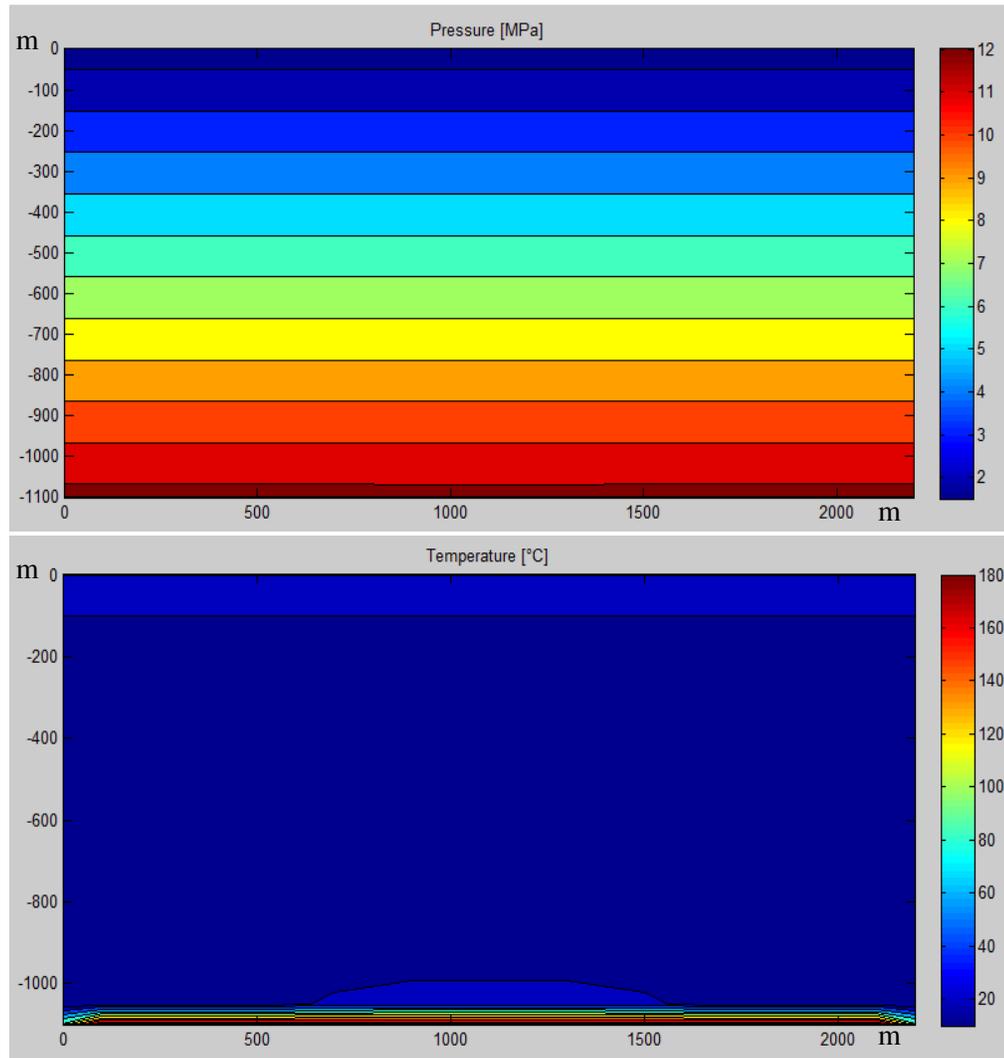
**Figura 29.** Se presenta la presión, temperatura y velocidades  $u$  y  $v$  durante los 630 días de la simulación.

Respecto a la presión se observa que disminuye hasta nuevamente estabilizarse y eso se debe a la variación de la densidad del fluido ya que principalmente la presión está controlada por un gradiente hidrostático.

En la temperatura se ve la tendencia a la alza partiendo de una temperatura de fondo hasta hacerse asintótico indicando la máxima temperatura que podría llegar a alcanzarse en esa zona con una fuente de calor que emite 90 MWatts.

Mientras más tiempo pase en la simulación el yacimiento tiende a estabilizarse, en las Figuras 21 a la 27 se puede ver pequeños cambios en los patrones de flujo pero siempre con la tendencia a estabilizar la temperatura global del yacimiento.

El simulador propuesto en esta tesis también permite obtener los contornos de presión y temperatura para diferentes instantes de tiempo tal como se muestran en las Figuras 30-35.



**Figura 30. Contornos de presión y temperatura al inicio de la simulación.**

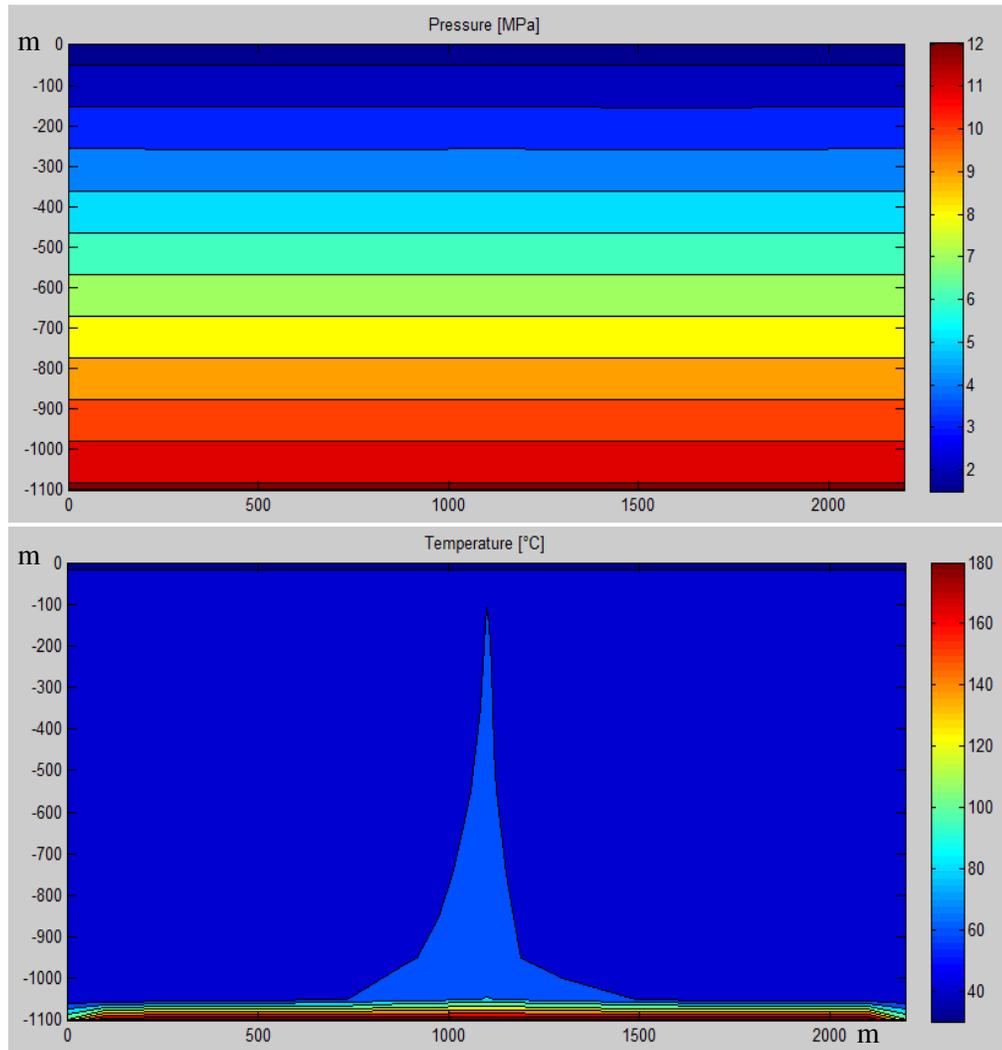


Figura 31. Contornos de presión y temperatura a los 2 días.

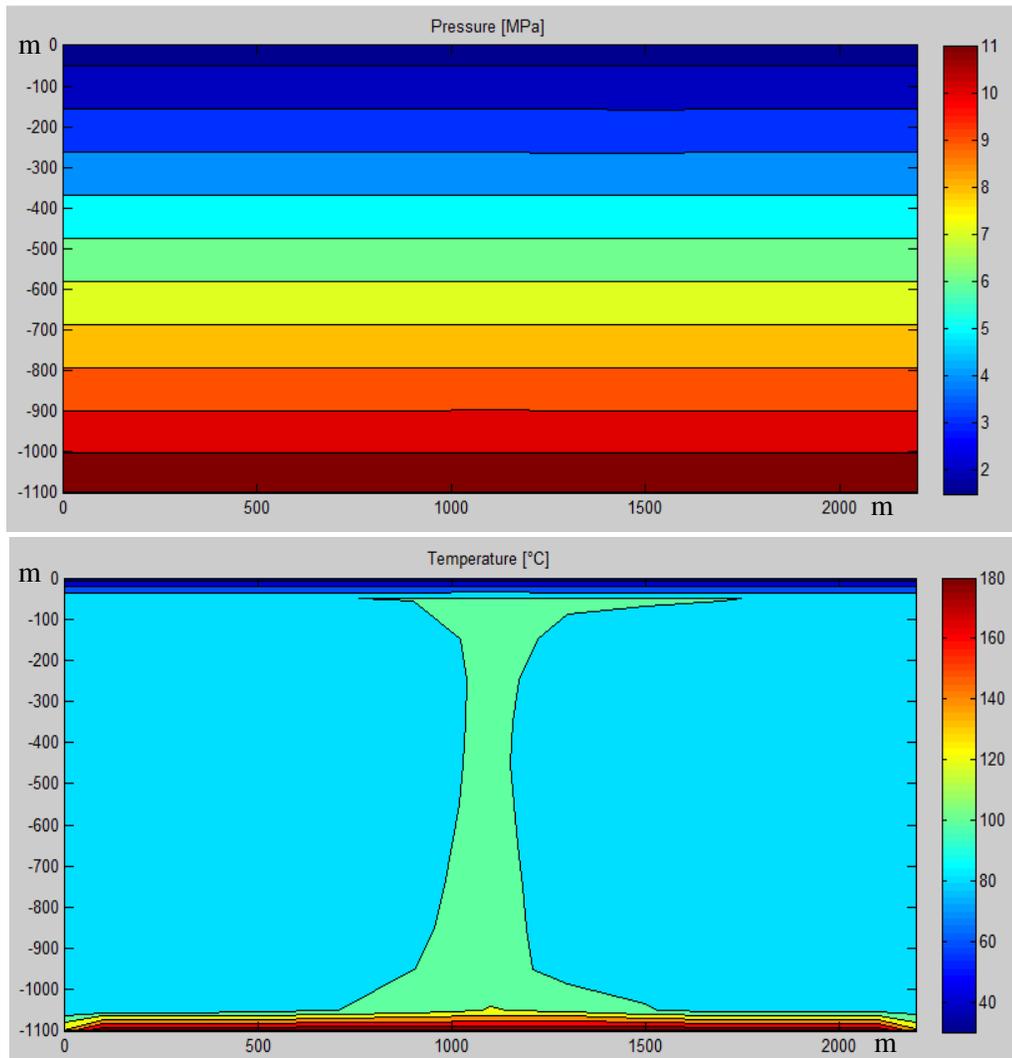


Figura 32. Contornos de presión y temperatura a los 20 días.

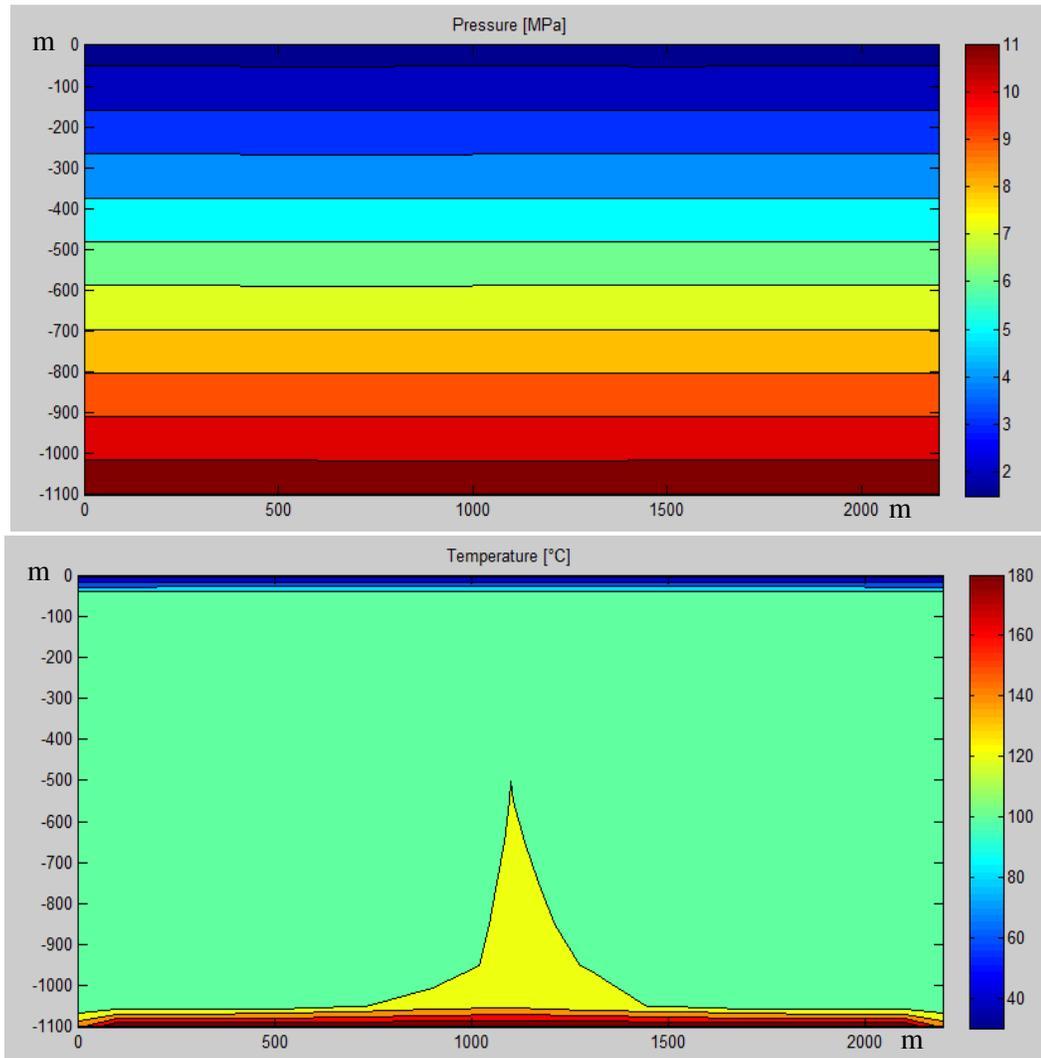


Figura 33. Contornos de presión y temperatura a los 60 días.

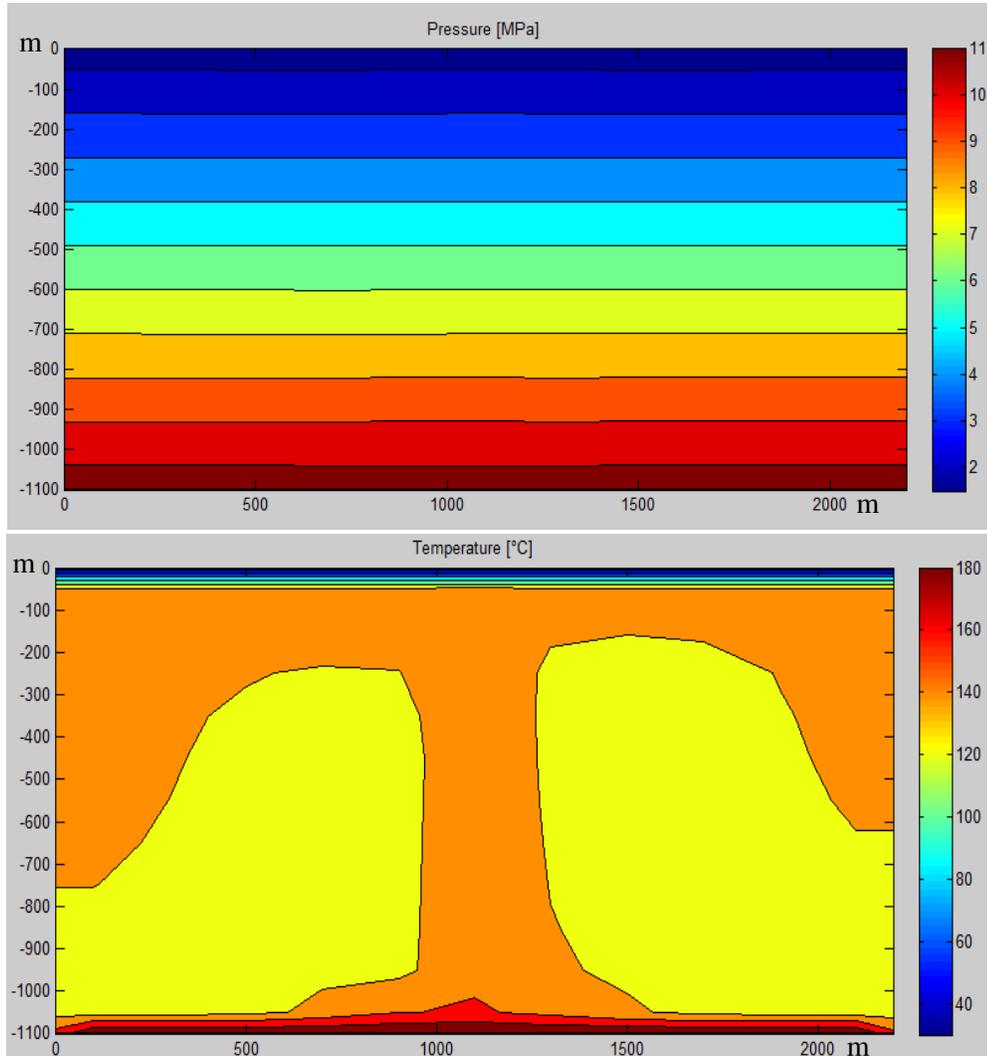


Figura 34. Contornos de presión y temperatura a los 345 días.

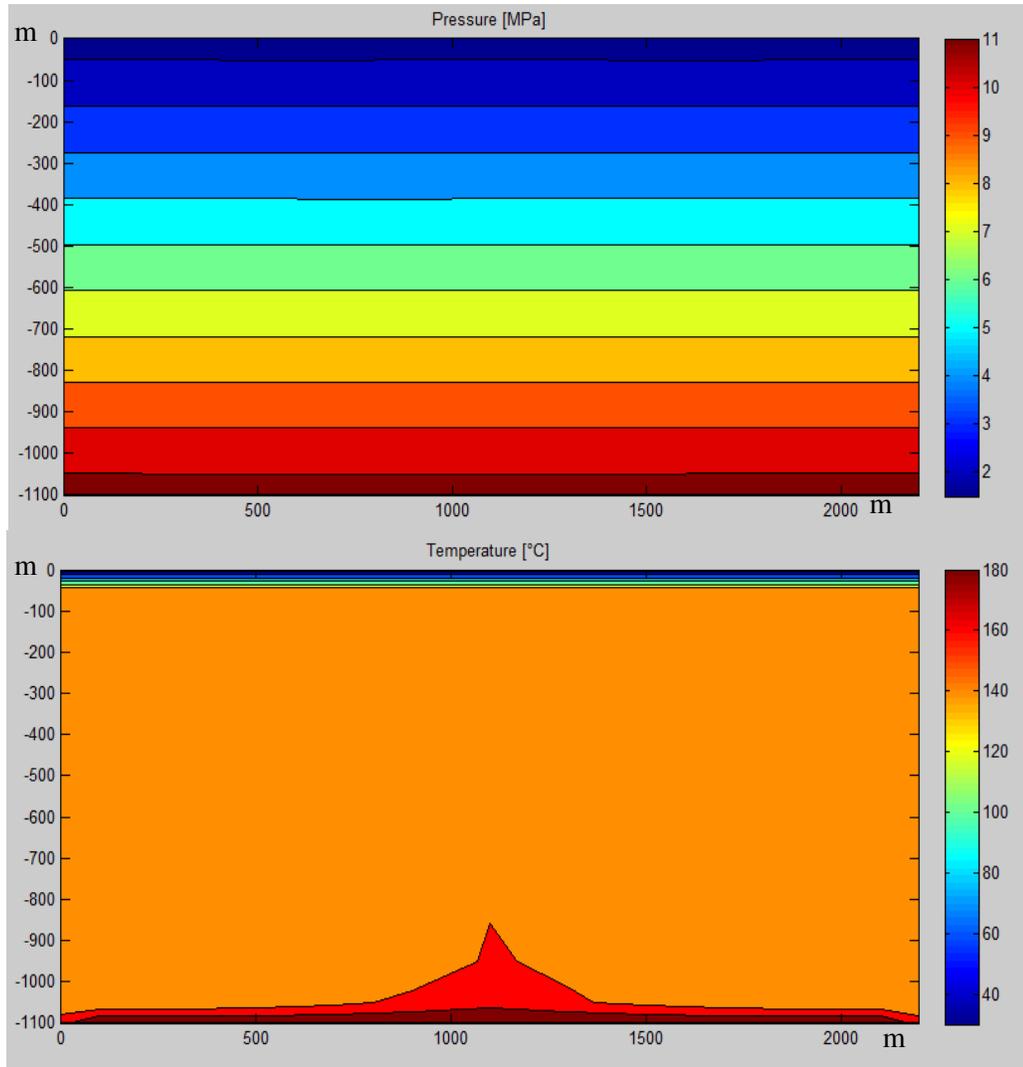


Figura 35. Contornos de presión y temperatura al final de la simulación.

Recordando el **Número de Peclet** del Capítulo dos  $Pe = \frac{F}{D} = \frac{\rho u dx}{\Gamma}$ ; en este problema son superiores a 2; están en el orden de 4.50 a 32. Y varían dependiendo de la densidad, la velocidad y la capacidad calorífica del fluido para cada celda. Por esto podemos concluir que el fenómeno de transferencia de calor está dominado por la convección, y que la conducción casi no influye. Es decir, la roca en sí misma funciona como aislante y por lo tanto, si el yacimiento no estuviese saturado 100% por agua, el yacimiento no se calentaría a estas temperaturas.

### 5.3 Modelo de caja fuente de calor en una esquina (Problema 3)

Este problema corresponde a un yacimiento cuya fuente de calor está ubicada en una esquina, de forma tal que solo debería apreciarse una sola celda convectiva. También se esperaría que no todo el yacimiento logre calentarse debido a las condiciones de permeabilidad y a las dimensiones del yacimiento.

Para cada instante de tiempo se realiza una comprobación de convergencia el cual está representado en la Figura 36. Los resultados de la simulación se aprecian en las siguientes secuencias de imágenes para diferentes tiempos (Figuras 37-42):

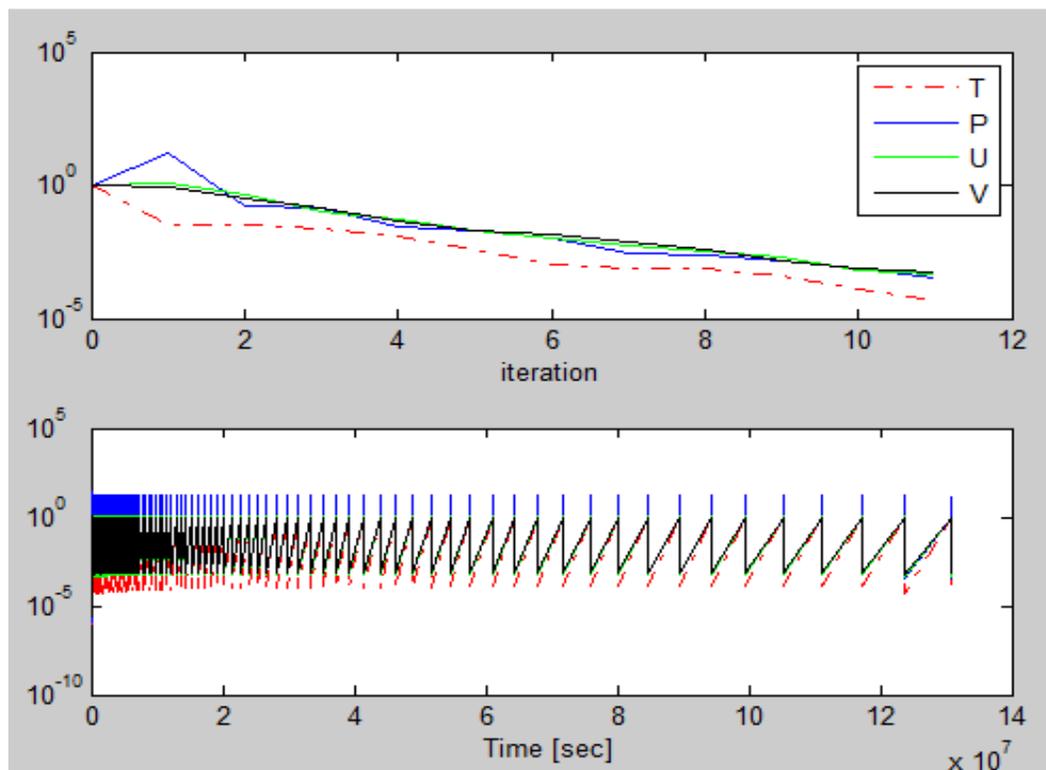


Figura 36. Convergencia de las ecuaciones.

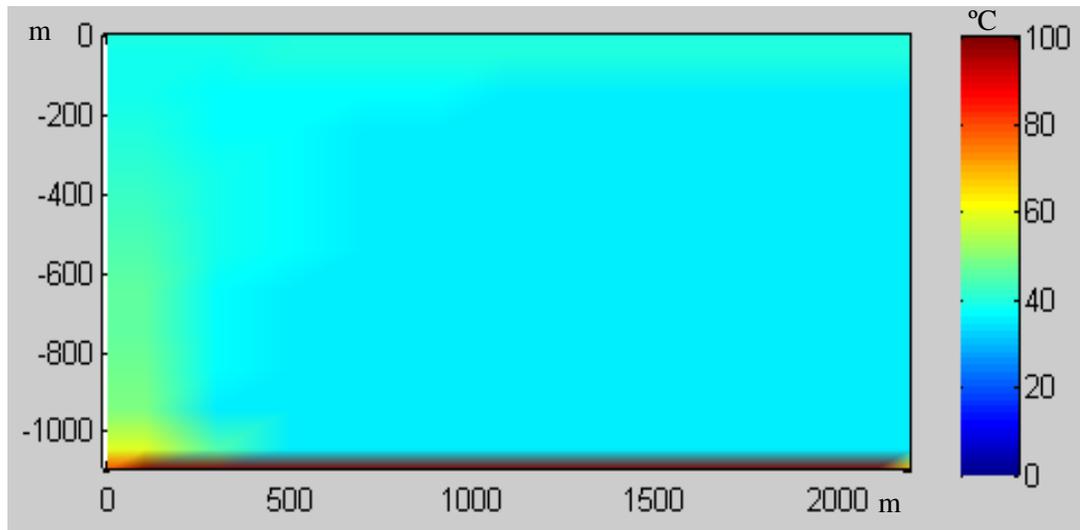


Figura 37. Condiciones de flujo y temperatura al inicio de la simulación.

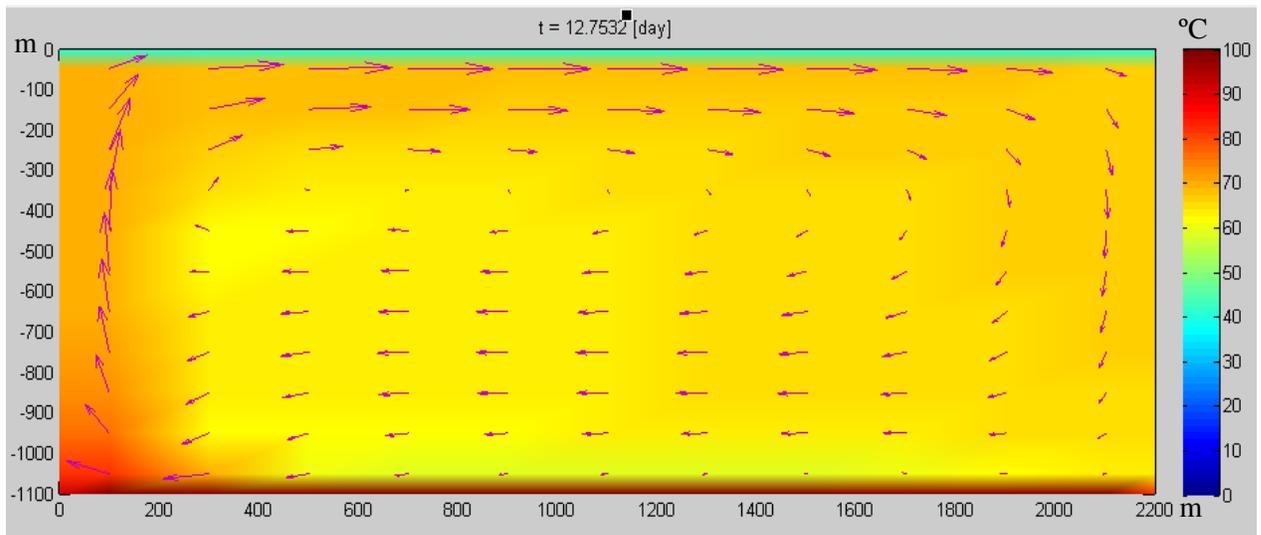


Figura 38. Condiciones de flujo y temperatura a los 12.75 días.

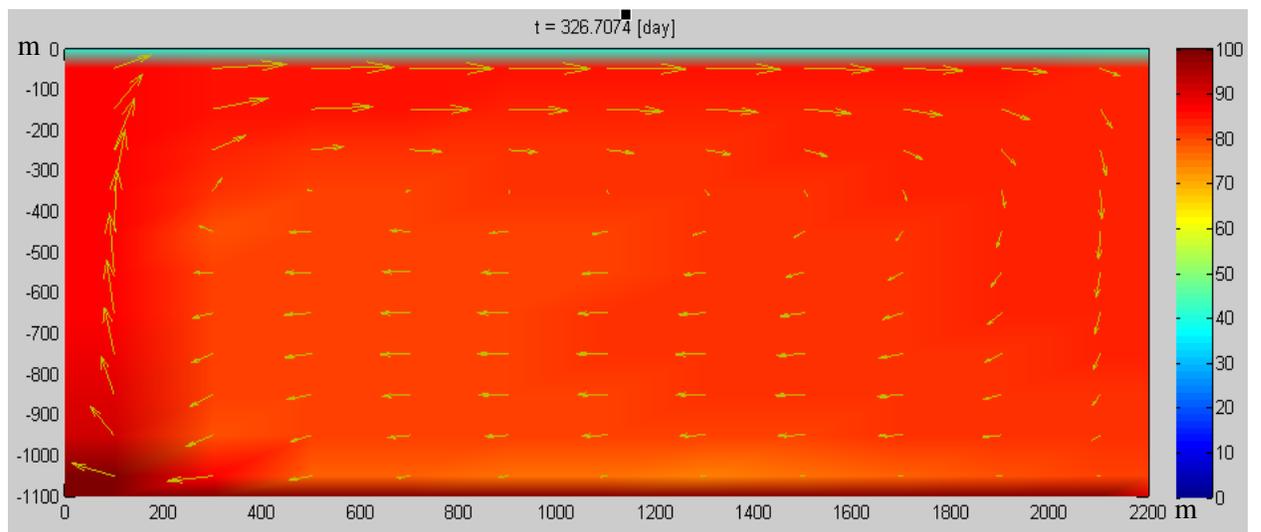


Figura 39. Condiciones de flujo y temperatura a los 326.7074 días.

Tabla7. Entradas para el problema 3.

<b>PROPIEDADES DEL MEDIO</b>		
Densidad (Kg/m <sup>3</sup> )		2600
Porosidad (fracción)		0.1
Permeabilidad (Darcy)		1x10 <sup>-3</sup>
Conductividad Térmica (W/m °C)		3.5
<b>CONDICIONES INICIALES DEL FLUIDO</b>		
	Temperatura ° C	Presión Mpa
Tope	40	3x10 <sup>6</sup>
Fondo	100 (isotérmico)	controlada por gradiente hidrostático
Inicial en el yacimiento	35	controlada por gradiente hidrostático
<b>ESTADO TRANSITORIO Y TOLERANCIAS</b>		
Tiempo inicial (s)		0
Tiempo final (s)		132451200 (1533 días, o 4.2 años)
Incremento del paso del tiempo		100
Tolerancia de Temperatura		0.01
Tolerancia de Presión		0.1
Tolerancia de velocidad U		0.001
Tolerancia de velocidad V		0.001
<b>FACTORES EXTERNOS</b>		
Fuente de calor (Watts)		90x106, 60x106
Nodos de inyección		1, 2
<b>CONDICIONES DE LA SIMULACIÓN</b>		
Tipo de algoritmo	acopladas	(ecuaciones acopladas y resueltas de forma simultánea)
Maximo de iteraciones por paso del tiempo		200
Frontera inferior Isotérmica		Si
Discretización en X		200
Discretización en Y		100
Discretización en Z		500
Cantidad de nodos en X		11
Cantidad de nodos en Y		11
Método de interpolación		Spline
Esquema de diferencias		Upwind

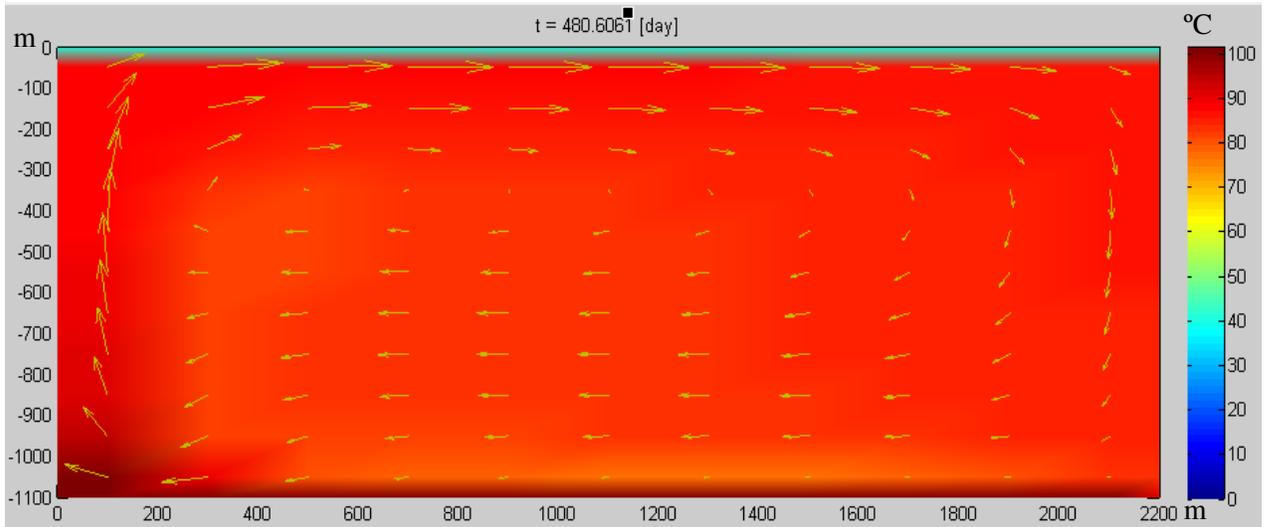


Figura 40. Condiciones de flujo y temperatura a los 480.60 días.

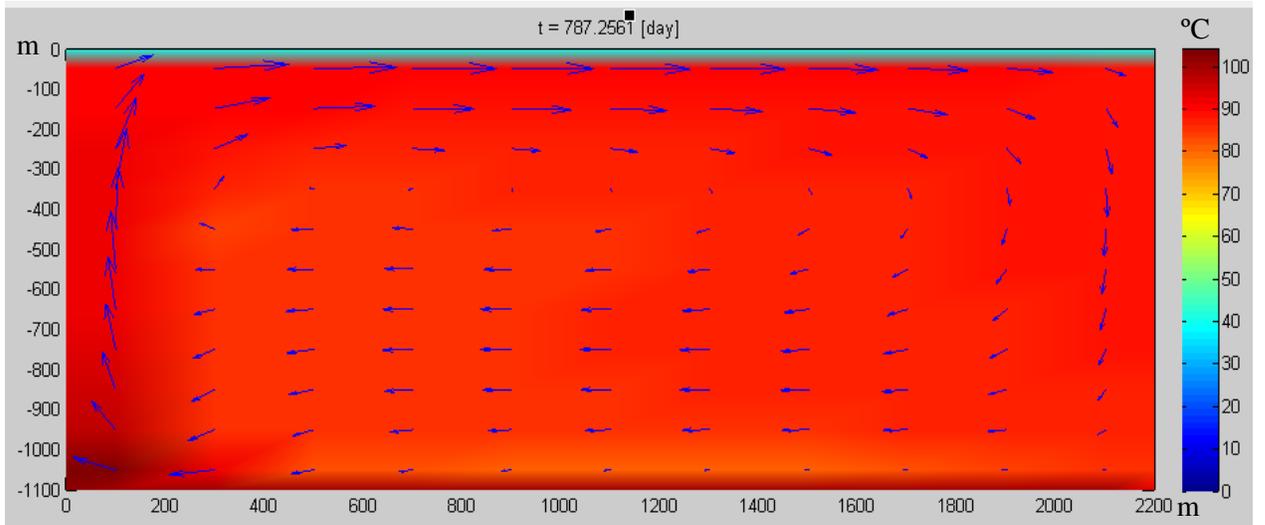


Figura 41. Condiciones de flujo y temperatura a los 787.25 días.

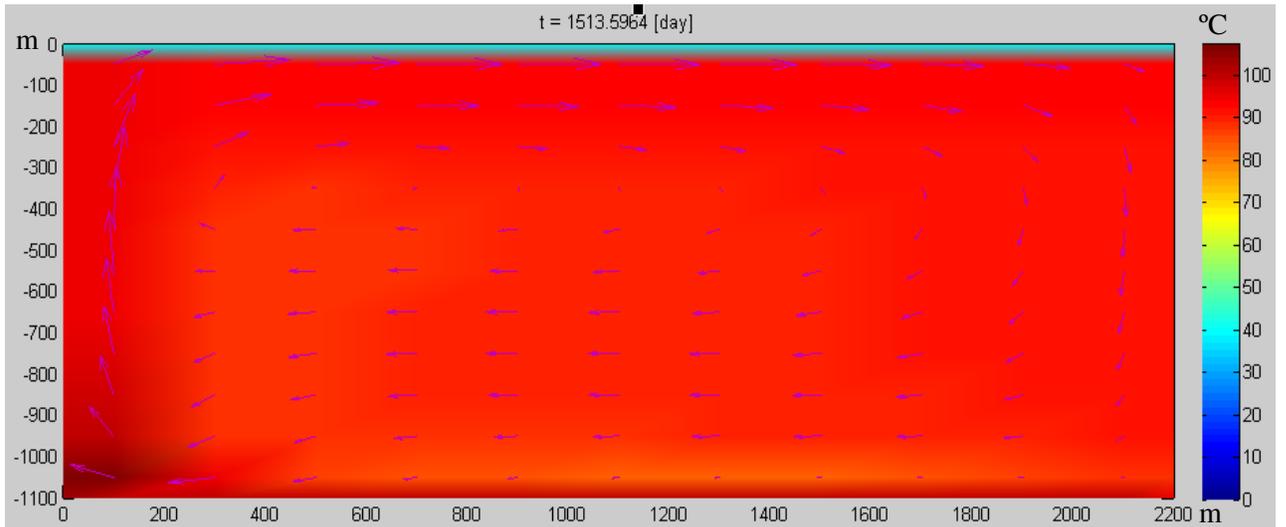
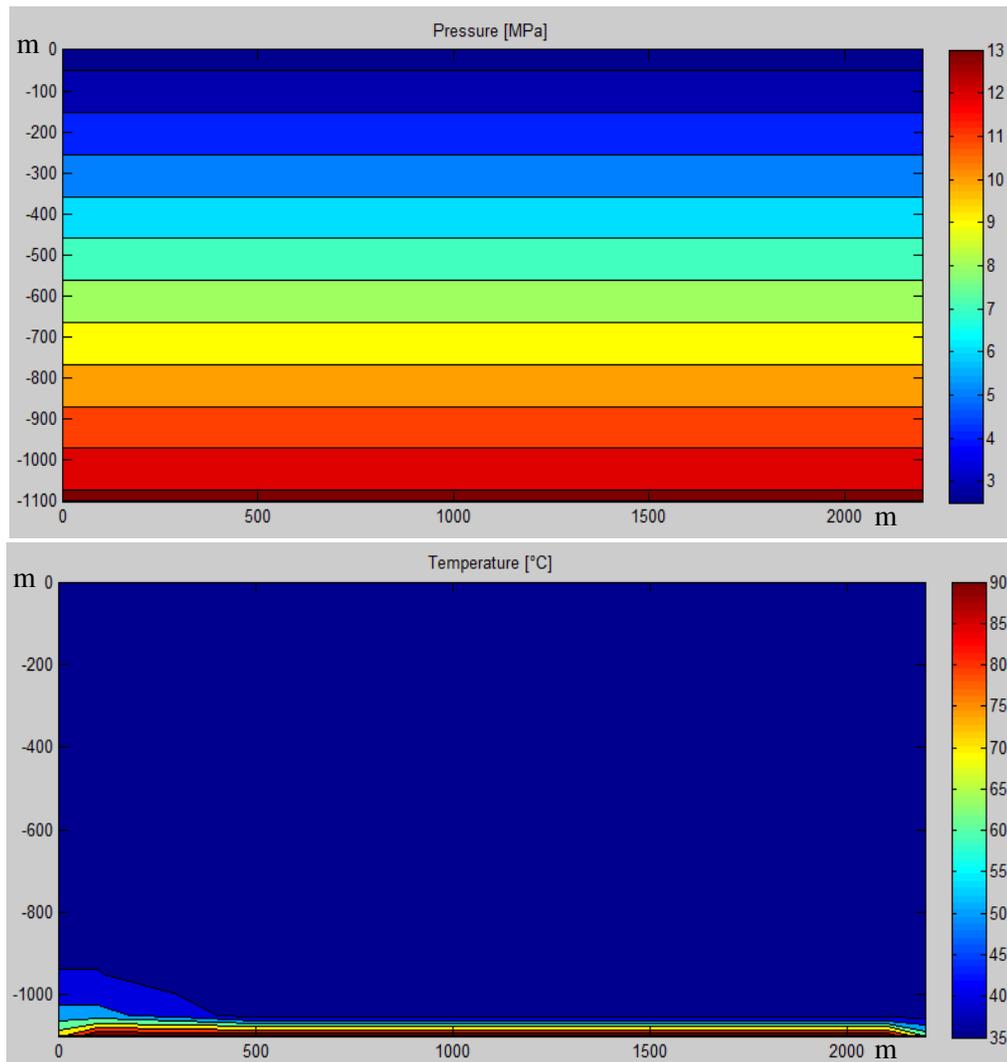


Figura 42. Condiciones de flujo y temperatura a los 1513.59 días.

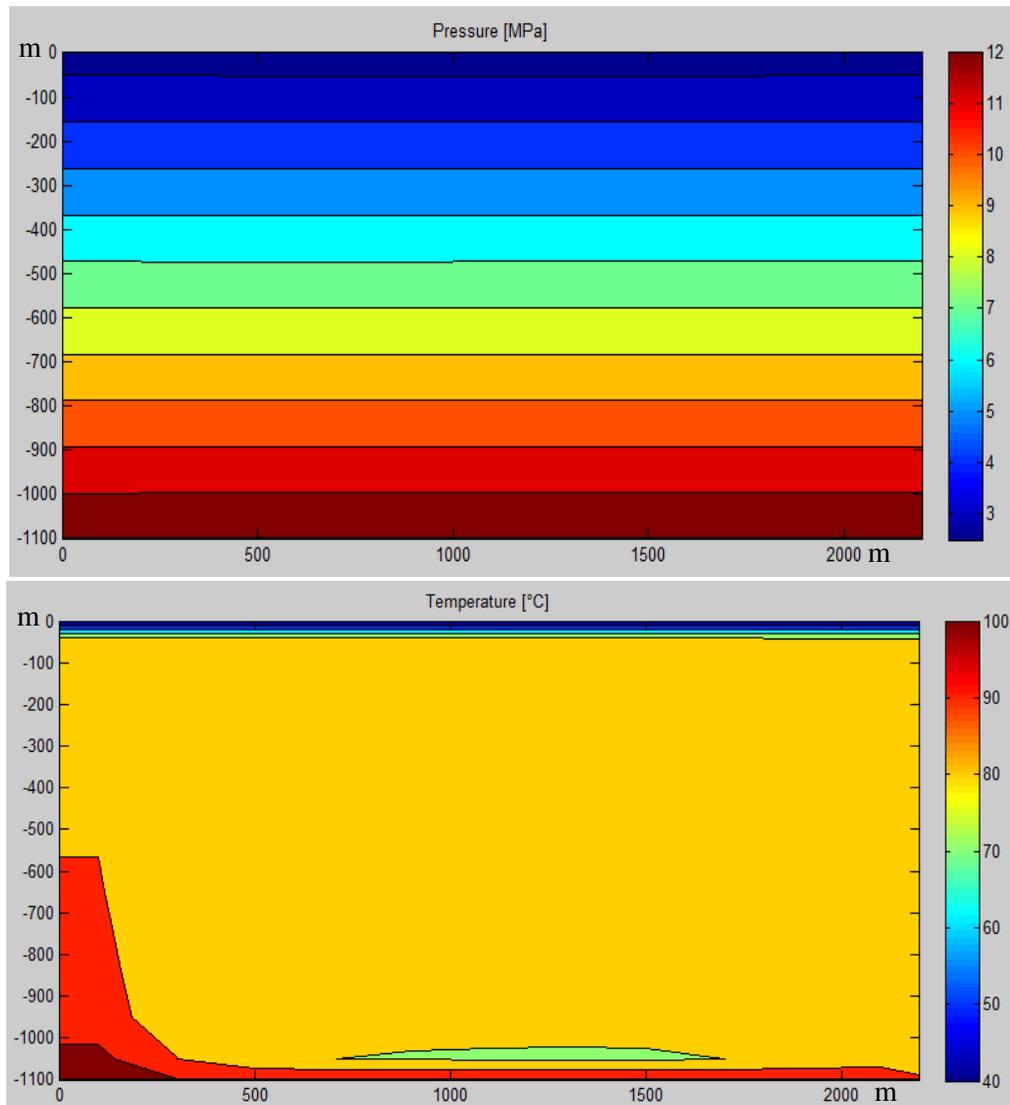
En este modelo se esperaba obtener justamente una sola celda; también se esperaba obtener un contorno de temperatura que represente la mitad del resultado del primer problema ya que básicamente el problema es el mismo pero partido a la mitad, ya que la fuente se encuentra en una esquina del yacimiento y las propiedades de no flujo en las fronteras no han cambiado.

Los modelos del contorno de presión y de temperatura para dos tiempos diferentes, se muestran en las gráficas 43 y 44.

Éstos gráficos tienen la particularidad de que ayudan a visualizar de mejor manera las isotermas y adicionalmente permite ver las isobaras.



**Figura 43. Contornos de presión y temperatura al inicio de la simulación.**



**Figura 44. Contornos de presión y temperatura a los 1513.59 días.**

Para estos niveles de presión y de temperatura se considera el yacimiento monofásico en estado líquido.

Comparando los gráficos del contorno de presión inicial (Figura 43) y final (Figura 44) se puede apreciar que la presión en el fondo del yacimiento decae de 13 MPa a 12 MPa, esto se debe a que la presión en los líquidos se encuentra dominada mayormente por el gradiente hidrostático. Recordemos que los líquidos son prácticamente incompresibles, y que no reaccionan como los gases. Los gases pueden comprimirse de tal forma que al hacerlo generan incrementos de presión.

En el gráfico de contornos de temperaturas al inicio de la simulación (Figura 43) se aprecia el efecto de la fuente, con una cama isotérmica, luego al final de la simulación (Figura 44) vemos la expansión y distribución de la temperatura, donde alcanza 100 grados centígrados desde los 1100 metros hasta casi los 1000 metros de profundidad y alrededor de 90 °C desde los 1000 metros hasta los 670 metros de profundidad. El resto del yacimiento se mantiene aproximadamente en 80 °C. También se aprecia una pequeña cama de fluido que está a un aproximado de 70 °C entre los 1050 y 1000 metros de profundidad, la cual debe ser consistente con una porción de fluido que se enfrió y descendió para volver a calentarse y así repetir el ciclo de convección.

En este Capítulo hemos comprobado el funcionamiento del simulador, también hemos realizado 2 ejercicios adicionales como ejemplos y se discutieron los resultados de cada caso.

Es importante recordar que debido a la no linealidad de las ecuaciones, tal como se describió en el Capítulo 2 y 3, el sistema de ecuaciones es altamente inestable y que para poder resolver este problema se recomienda utilizar el método “upwind”, también conocido como diferencias centrales hacia adelante. Para ayudar a la convergencia y a la estabilidad, hay que elegir los parámetros de entrada correctamente, y es crítico definir bien las condiciones de frontera para cada problema.

Un yacimiento geotérmico funciona como una lámpara de lava, la misma está compuesta por agua y cera, donde la cera al ser calentada se hace menos densa que el agua y al enfriarse cae porque se hace más densa, cumpliendo así con un ciclo de convección. Comparando el problema 3 con el caso de la lámpara, se entiende claramente porqué se encuentra esa pequeña capa de agua, que está a menor temperatura que el resto del yacimiento, en el fondo del mismo. En las Figuras 45 y 46 se muestra un ejemplo de las lámparas de lava.



Figura 45. Lámparas de lava.

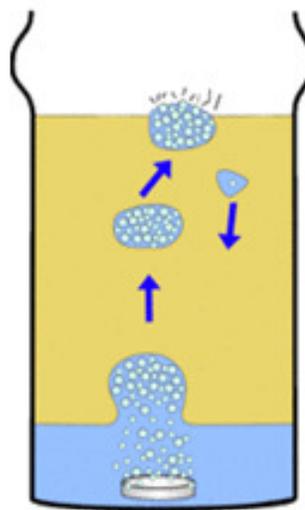


Figura 46. Representación del funcionamiento de las lámparas de lava.

## Capítulo 6. Conclusiones

---

Es una realidad que los combustibles fósiles se agotan y representan una amenaza ecológica, también que la energía nuclear es muy costosa y es peligrosa, por lo que la ciencia ha tenido que evolucionar para dar paso a nuevas formas de generación de energía o de aprovechamiento de las energías que ya están disponibles en el planeta, tal es el caso de la energía geotérmica; la cual es proveniente del interior de la Tierra y es una energía que está disponible para aprovecharla.

Sin embargo, hay que estudiarla y conocerla para poder extraer ésta energía eficientemente, evitando así el agotamiento prematuro de los yacimientos geotérmicos.

De esta forma podríamos decir que la geotermia es la ciencia que estudia las condiciones térmicas de la Tierra, y la energía geotérmica es la energía que puede ser obtenida aprovechando el calor proveniente de la Tierra.

Los estudios de la geotermia son complejos debido a la gran cantidad de factores que influyen en el comportamiento de los yacimientos geotérmicos, y la dificultad que presenta conocerlos o definirlos correctamente.

De los diferentes estudios que se han realizado, algunos tratan de generar modelos matemáticos que representen el sistema físico de los yacimientos geotérmicos, esto con el fin de poder predecir el comportamiento de los mismos y poder aprovechar mejor esta fuente de energía. Sin embargo, la complejidad de las ecuaciones matemáticas que describen estos procesos físicos, ha llevado a generar diferentes modelos numéricos.

El modelo numérico de este trabajo, parte de las ecuaciones generales de conservación de la masa, de la energía y de momento, las cuales fueron simplificadas siguiendo esquemas que ya fueron previamente comprobados como por ejemplo la ley de flujo de Darcy para yacimientos porosos.

En el conjunto de ecuaciones simplificadas, se asume que las propiedades de la roca yacimiento son constantes en todas las direcciones y que no varían en el tiempo, algunas de estas propiedades son por ejemplo: la porosidad, la permeabilidad, la conductividad térmica de la roca. También se asume que el yacimiento está totalmente saturado por agua pura en estado líquido y cuyas propiedades, como por ejemplo la densidad, viscosidad y capacidad calorífica, son variables en todas las direcciones y en el tiempo.

Las ecuaciones desarrolladas en este trabajo no contemplan la fase de vapor de agua, por lo que no se obtendrían resultados correctos para flujos bifásicos o monofásicos de vapor.

Adicionalmente en este trabajo sólo se contempla convección natural, la convección forzada es otro tema de estudio, ya que altera las ecuaciones de conservación de masa y energía, es decir que altera los patrones de flujo y de temperatura del yacimiento geotérmico. Definitivamente debido a la complejidad de las ecuaciones matemáticas que representan el sistema físico en estudio, se requiere el uso de soluciones numéricas mediante cálculos computacionales para poder predecir el comportamiento de los yacimientos geotérmicos.

En el primer ejemplo que se encuentran en el Capítulo 5 de este trabajo, se realiza una validación del simulador. Se compara el resultado obtenido en esta tesis, con un resultado propuesto por Sorey en 1978 y se observa la similitud entre ambos.

El yacimiento de este ejercicio, posee una fuente de calor en el centro que se va atenuando hacia los extremos para crear el efecto de una fuente radial. Las condiciones de frontera son de no flujo, y las paredes del yacimiento se consideran adiabáticas, y de temperatura constante.

Los resultados obtenidos fueron congruentes con lo esperado del modelo teórico, generando dos celdas de convección a ambos lados de la fuente de calor central.

En el segundo ejemplo, se realiza un ejercicio similar al primero, pero con otros valores de fuente de calor, tal como se indican en la Tabla 5.1 del Capítulo 5, y con la variante de que la temperatura de fondo en este yacimiento es isotérmica y de 200 °C.

Igualmente, se observan dos grandes celdas de convección a lo largo de 632.43 días que es el tiempo que tarda en estabilizar el yacimiento. Es normal que los tiempos de estabilización de un yacimiento sean tan grandes, incluso pueden tardar más de 2 años en estabilizarse, esto debido a las altas temperaturas que influyen en el sistema.

Un buen ejemplo comparativo de cómo funcionan las celdas de convección puede verse simplemente con una lámpara de lava. Estas lámparas poseen agua y cera. La cera al estar a temperatura ambiente es más densa que el agua por lo tanto se hunde, pero al calentarse el sistema, la cera con el calor se hace menos densa que el agua y por ende termina subiendo al tope de la lámpara, al enfriarse vuelve a caer y se repite el ciclo una y otra vez.

Para darnos idea de las diferencias de magnitudes entre una lámpara de lava y un yacimiento geotérmico. Una lámpara de lava tiene un generador de calor de 40 Watts y tarda aproximadamente 45 minutos en calentar un litro de agua a 60° C con un área de contacto aproximado de 0.007 metros cuadrados. El yacimiento geotérmico del segundo problema propuesto, tiene una fuente de calor de 90 millones de Watts actuando en tres celdas de 100 mil metros cuadrados cada una. Se esperaría que tarde años en calentar al yacimiento.

Finalmente tras realizar este proyecto y realizar varias simulaciones se puede observar lo siguiente:

El medio en sí funciona como aislante térmico, es decir, que su conducción es básicamente despreciable y que quien realmente transporta el calor es el término convectivo. Retomando el ejemplo de la lámpara de lava, si la misma estuviese llena con arena y 100 por ciento saturada de agua, tardaría más de 45 minutos en calentarse a los 60 °C, debido a que el calor básicamente es llevado por el agua dentro de la lámpara y ahora ésta agua tendría movilidad limitada porque está siendo limitada en su

movimiento por la permeabilidad de la roca, ahora bien, si la lámpara estuviese llena de arena sin agua, entonces seguro el fondo de la lámpara estaría a 60 °C pero el tope de la misma estaría a temperatura ambiente, cumpliendo que la conducción no es tan eficiente como la convección.

Matemáticamente hablando esto se demuestra con altos valores en el número de Peclet, los cuales definen que el flujo de calor esté dominado en sí por la convección.

Recordando, los valores del el **Número de Peclet**, definido en el segundo Capítulo como  $Pe = \frac{F}{D} = \frac{\rho u dx}{\Gamma}$ ; en el segundo problema, son superiores a 2.0 y están en el orden de 4.50 a 32, variando dependiendo de la densidad, la velocidad y la capacidad calorífica del fluido para cada celda.

Los factores que dominan la convección son la viscosidad y la permeabilidad del medio. La porosidad no afecta de manera apreciable el flujo de calor, lo que realmente importa son los poros interconectados entre sí y la capacidad que tiene el fluido para pasar por este espacio.

La solución de las ecuaciones de conservación es altamente inestable, por lo que no siempre se puede conseguir la convergencia de los problemas.

Ajustar correctamente los parámetros de entrada, es decir, colocar mayor o menor cantidad de celdas con espaciamientos adecuados, programar simulaciones con un espaciamiento en el tiempo correcto, entre otros factores como condicionar bien las fronteras, es esencial para conseguir la estabilidad y convergencia de los problemas.

También es importante seleccionar correctamente los criterios de convergencia. Una alternativa para mejorar el algoritmo, es condicionarlo con un método de estabilización que permita obtener convergencia en la solución y mejore los resultados de la estimación de las variables de interés.

## Lista de referencias bibliográficas

---

- Bird R.B., W.E. Stewart, E.N. Lightfoot., (2002). *Transport Phenomena*. (2<sup>a</sup>. ed.). John Wiley & Sons, Inc., USA
- Boltzmann, L., (1986). *Escritos de mecánica y termodinámica*. (5<sup>a</sup>. ed.) Alianza Editorial
- Callen, H. B., (1985). *Thermodynamics an introduction to thermostatics*. (3<sup>a</sup>. ed.). John Wiley & Sons
- Chekhonin E., Parshin A., Pissarenko D., Romushkevich R., *et. al.* (2013). Cuando las rocas se calientan: las propiedades térmicas de las rocas yacimiento. *Oilfield review* 24(3)20-27 Recuperado 2015 de [http://www.slb.com/~/media/Files/resources/oilfield\\_review/spanish12/au12/composite.pdf](http://www.slb.com/~/media/Files/resources/oilfield_review/spanish12/au12/composite.pdf)
- Cruz Chávez, M. A. (1998) Desarrollo del simulador numérico "geo" para el transporte de masa y energía en yacimientos geotérmicos, utilizando la técnica de diferencias finitas integradas. Tesis de maestría. Instituto Tecnológico de Monterrey.ITESM.
- Hougen, Olaf A., Kenneth. M. Watson, R. A. Ragatz., (1982). *Principios de los procesos químicos*. Termodinámica. Editorial Reverté, Barcelona.
- Jiménez Morales, F.; Lemos Fernández M. C., (2001). *Termodinámica: una guía de clase*. (vol. 53) Universidad de Sevilla.
- Kretzshmar Hans-Joachim, Ines Stoecker and M. Kunick, (2007). Revised Release on the IAPWS Industrial Formulation 1997 for the Thermodynamic Properties of Water and Steam. Lucerne, Switzerland, Recuperado en: <http://www.iapws.org>
- Müller E., (2002), *Termodinámica Básica*. (2<sup>a</sup>. ed.). Consultora Kemiteknik C.A. Caracas, Venezuela
- Patankar, S. (1980). *Numerical heat transfer and fluid flow*. CRC press.
- Patankar, S. V., and Spalding, D. B. (1972). A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International journal of heat and mass transfer*, 15(10), 1787-1806.

- Pollack, H. N., Hurter, S. J., and Johnson, J. R. (1993). Heat flow from the Earth's interior: analysis of the global data set. *Reviews of Geophysics*, 31(3), 267-280.
- Reif, F., and Scott, H. L. (1998). Fundamentals of Statistical and Thermal Physics. *American Journal of Physics*, 66(2), 164-167.
- Rinehart, J. S. (1980). *Geysers and geothermal energy* (Vol. 223). New York: Springer-Verlag.
- Tarbutck, E. J., and Lutgens, F. K. (2005). Ciencias de la Tierra, una introducción a la geología física. ( 8ª ed.). Madrid, España: Editorial Prentice Hall.
- Versteeg, H.K.,W. Malalasekera., (1995). An Introduction to computational fluid dynamics: the finite volume method. Longman Scientific & Technical, London.

## APÉNDICE A

### NOMENCLATURA

PARÁMETROS		UNIDADES
$A$	Área de la intercara	$m^2$
$C_p$	Calor específico	$J/kg\cdot C$
$d$	Distancia entre los nodos de dos elementos	$M$
$D$	Difusión	$Kg/m^2\cdot s$
$dt$	Variación del tiempo	$s$
$F$	Flujo másico	$Kg/m^2\cdot s$
$g$	Vector de aceleración gravitacional	$m/s^2$
$k$	Permeabilidad	darcy
$K$	Conductividad térmica	$W/m^{\circ}C$
$n_x, n_y$	Número de celdas en X o en Y	
$P$	Presión	Pascales
$Pe$	Número de Peclet	-
$Q$	Generación de calor	Watts
$t$	Tiempo	$S$
$T$	Temperatura	$^{\circ}C$
$V$	Volumen	$m^3$
$\rho$	Densidad del fluido	$kg/m^3$
$\phi$	Porosidad (fracción)	Adimensional
$\mu$	Viscosidad	Pa-s

## APÉNDICE B

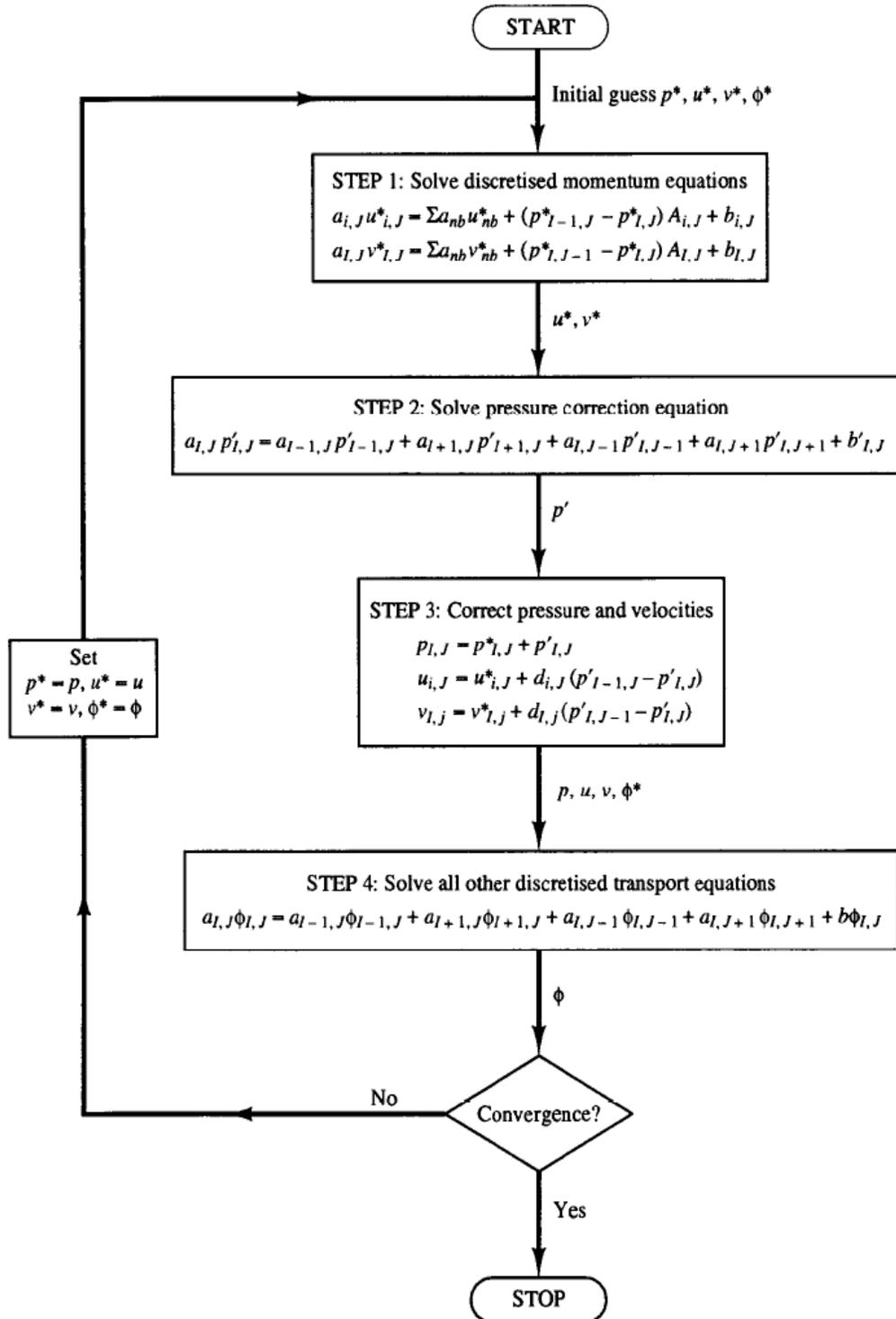


Figura47. Diagrama de flujo del método simple Versteeg &amp; Malalasekera, 1995.

## APÉNDICE C

### Código del simulador

<b>CENTRO DE INVESTIGACIÓN CIENTÍFICA Y EDUCACIÓN SUPERIOR DE ENSENADA, BAJA CALIFORNIA (CICESE) MARZO DE 2015</b>	
<b>PROGRAMA</b>	: simulador
<b>DESCRIPCION</b>	: Simulador de yacimientos geotérmicos monofásico en dos dimensiones
<b>FLUIDO</b>	: Agua pura
<b>ESTADO</b>	: Estacionario - Transitorio
<b>ELABORADO POR</b>	: Ing. Fernando José Bello Fuentes
<b>TESIS</b>	: Para la obtención del grado de maestro en ciencias de la Tierra, geofísica aplicada.
<b>ASESOR</b>	: Dr. Rogelio Vázquez González

#### Módulo Principal:

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%FUNCIÓN QUE PREPARA EL SET DE DATOS A UTILIZAR%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```
function
```

```
[U,V,P,T,constants]=PrepareInputData(T0,Tb,Tt,Pt,nx,ny,dx,dy,dz,dt,q0,iq0,kFactor,Per,IsotBound)
```

```
addpath([cd '/waterprop']); % agregar el directorio waterprop
```

```
addpath([cd '/Estacionario']); % agregar el directorio /estacionario
```

```
g=9.8;
```

```
constants.g =g;
```

```
constants.dx=dx;
```

```
constants.dy=dy;
```

```
constants.dz=dz;
```

```
constants.dt=dt;
```

```
constants.nx=nx;
```

```
constants.ny=ny;
```

```
constants.q0=zeros(ny,nx);
```

```
constants.q0(end,iq0)=q0;
```

```
constants.kFactor=kFactor;
```

```
constants.Per = Per;
```

```
constants.Tb=Tb;
```

```
constants.Tt=Tt;
```

```
constants.Pt=Pt;
```

```
constants.IsotBound=IsotBound;
```

```
[xm,ym,xu,yu,xv,yv] = gridshow([nx ny],[dx dy]);
```

```
constants.x=[xu(1) xm xu(end)];
constants.y=[yv(1) ym yv(end)];
```

```
U=zeros(ny,nx+1);
V=U';
```

```
T=ones(ny,nx)*T0;
T(1,:)=Tt;
```

```
y=0:dy:(ny-1)*dy;
rho=1000;
P=meshgrid(y)'*g.*rho+Pt;
```

```
[Cp,rho,mu,k]=ComputeProperties(T,P);
P=meshgrid(y)'*g.*rho+Pt;
```

```
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% FUNCIÓN QUE SE ENCARGA DE LLAMAR A LOS ALGORITMOS PARA ARMAR EL      %%%
%%% SISTEMA DE ECUACIONES. TAMBIÉN SE ENCARGA DE CHECAR LOS CRITERIOS DE%%%
%%% CONVERGENCIA E IMPRIMIR LOS RESIDUOS EN LA PANTALLA PRINCIPAL      %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function [U,V,P,T,t]=cfdsolve(T0,P0,U0,V0,constants,options,algorithm)
if nargin<7, algorithm = 'coupled'; end% algoritmo por defecto
residual.rP=[]; residual.rT=[]; residual.rU=[]; residual.rV=[];
residual.rb=[];
residual.rt=[];
dt=constants.dt;
k=1; cond = true;
U(:,:,k)=U0; V(:,:,k)=V0; T(:,:,k)=T0; P(:,:,k)=P0;
while cond
t(k)=(k-1)*dt;

%dt=dt*1.09^(1/(2^.2));
dt=dt*1.05;
    residual.t=t(k);
clc;fprintf('Tiempo t = %.2f\n',t(k));

[U1,V1,P1,T1,residual]=cfdalgorithm(T0,P0,U0,V0,constants,options,algorithm,
residual); %ejecutar algoritmo

    U(:,:,k)=U1; V(:,:,k)=V1; T(:,:,k)=T1; P(:,:,k)=P1; % introducir todo en
las matrices

DT=max(max(abs((T0-T1))))>options.SteadyState*options.TolT; % verificar
estado est para T
    DP=max(max(abs((P0-P1))))>options.SteadyState*options.TolP; % verificar
estado est para P
    DU=max(max(abs((U0-U1))))>options.SteadyState*options.TolU; % verificar
estado est para "u"
    DV=max(max(abs((V0-V1))))>options.SteadyState*options.TolV;
cond = DT || DP || DU || DV; % verificar todas los estados estacionarios
cond = cond && t(k)<=options.TimeMax; % ademas si t es menor a tmax, continuar
la simulacion
```

```

    T0=T1; P0=P1; U0=U1; V0=V1; % redefinir valores iniciales
    k=k+1; % iterar el proximo tiempo
end
end

function
[U1,V1,P1,T1,residual]=cfdalgorithm(T0,P0,U0,V0,constants,options,algorithm,residual)
addpath([cd '/algorithm']); % agregar el directorio /algorithm
switch algorithm
case'SIMPLE'
    [U1,V1,P1,T1,residual]=
SIMPLE(T0,P0,U0,V0,T0,P0,U0,V0,constants,options,residual);
case'coupled'

[U1,V1,P1,T1,residual]=SolveSystem(T0,P0,U0,V0,T0,P0,U0,V0,constants,options,r
esidual);
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% FUNCIÓN QUE SE ENCARGA DE ELEGIR LOS NODOS VECINOS DE LAS MATRICES   %%%
%%% TAMBIÉN RESUELVE SEGÚN EL ESQUEMA DE SOLUCIÓN DE LAS ECUACIONES     %%%
%%% ELEGIDO. DIFERENCIAS CENTRADAS, UPWIND, HÍBRIDO, EXPONENCIAL O      %%%
%%% POTENCIA                                                              %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function a = aneighbor(F,D,DtTerm,scheme)
if nargin<4, scheme = 'upwind'; end% por defecto escojemos upwind
a=zeros(1,5);
if all(D==0) % es para usarlo en la ecuacion de continuidad
    a = F;
else% es para usarlo en las ecuaciones de transporte
    P = F./D; % numero de peclet
    sg = [-1 1 0 -1 1]; % signos de [N W P E S]
    for i = 1:5
    if i~=3
    if ~isnan(P(i))
        a(i) = D(i)*TransportScheme(abs(P(i)),scheme) +
max([sg(i)*F(i) 0]);
    end
    end
    end
a(3) = sum(a) + DtTerm;
end
end

function A = TransportScheme(P,scheme)
switch scheme
case'centred'% esquema centrado, solo sirve para dx pequeños
A = 1-abs(P)/2;
case'upwind'% esquema upwind
A = 1;
case'hibrido'% esquema centrado y upwind dependiendo de cual es el maximo
    A = max([1-abs(P)/2 0]);
case'exp'% esquema funcion exponencial, este es el mas exacto
if P ~=0
    A = abs(P)/(exp(abs(P))-1);

```

```

else
    A = TransportScheme(P, 'upwind');
end
case 'power'% esquema de la potencia
A = max([(1-abs(P)/10)^5  0]);
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% FUNCION QUE CALCULA LOS FLUJOS DIFUSIVOS Y CONVECTIVOS          %%%
%%% DE LA ECUACION DE TRANSPORTE DE CANTIDAD DE MOVIMIENTO Y DE ENERGIA %%%
%%% T,P,U,V = MATRICES                                             %%%
%%% CONJUNTO DE CONSTANTES                                        %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [Duu,Dvv,DTu,DTv,Fum,Fvm,Fu,Fv,Ax,Ay,Rho,Cp] =
flux(T,P,U,V,constants)
addpath([cd '/waterprop']); % agregar el directorio waterprop

% calcular propiedades en los nodos principales
[Cp,rho,mu,k]=ComputeProperties(T,P);

k = k*constants.kFactor;

% encontrar el numero de subintervalos a partir de las dimensiones de la
% matriz principal

[ny,nx]=size(T);

% extraer diferenciales
dx = constants.dx; dy = constants.dy; dz = constants.dz;

% interpolar propiedades para la matriz del mallado de U
Cpu = U*0; rhou = U*0; muu = U*0; ku = U*0;

for j = 1:nx-1
    Cpu(:,j+1) =(Cp(:,j) + Cp(:,j+1))/2;
    rhou(:,j+1)=(rho(:,j) + rho(:,j+1))/2;
    muu(:,j+1)=(mu(:,j) + mu(:,j+1))/2;
    ku(:,j+1)=2*(k(:,j).*k(:,j+1))./(k(:,j)+k(:,j+1));
end
Cpu(:,1)=Cp(:,1);    Cpu(:,end)=Cp(:,end);
rhou(:,1)=rho(:,1); rhou(:,end)=rho(:,end);
muu(:,1)=mu(:,1);   muu(:,end)=mu(:,end);
ku(:,1)=k(:,1);    ku(:,end)=k(:,end);

% interpolar propiedades para la matriz del mallado de V
Cpv = V*0; rhov = V*0; muv = V*0; kv = V*0;
for i = 1:ny-1
    Cpv(i+1,:) =(Cp(i,:) + Cp(i+1,:))/2;
    rhov(i+1,:)=(rho(i,:) + rho(i+1,:))/2;
    muv(i+1,:)=(mu(i,:) + mu(i+1,:))/2;
    kv(i+1,:)=2*(k(i,).*k(i+1,))./(k(i,)+k(i+1,));
end
Cpv(1,:)=Cp(1,:);    Cpv(end,:)=Cp(end,:);

```

```

rhov(1,:)=rho(1,:);   rhov(end,:)=rho(end,:);
muv(1,:)=mu(1,:);   muv(end,:)=mu(end,:);
kv(1,:)=k(1,:);     kv(end,:)=k(end,:);

% areas transversales de flujo
Ax = dx*dz;   Ay = dy*dz;

% calcular flujos máxicos en todos los nodos de U y V

Fum = rhou.*Ay;   Fvm = rhov.*Ax;
Fu = rhou.*U.*Ay;   Fv = rhov.*V.*Ax;

% calcular los flujos difusivos de cantidad de movimiento y de energia en
% los mallados de U y V
Duu = muu.*Ay/dx;   Dvv = muv.*Ax/dy;
DTu = (ku./Cpu).*Ay/dx;   DTv = (kv./Cpv).*Ax/dy;

Rho.T =rho; Rho.U =rhou; Rho.V =rhov;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% ESTA FUNCION TOMA LOS PUNTOS CARDINALES EN UN NODO ARBITRARIO (I,J) %%%
%%% DE UNA MATRIZ A. LA SALIDA ES UN VECTOR DE DIMENSION 5 QUE CONTIENE %%%
%%% A (I,J)Y SUS PUNTOS VECINOS. TAMBIEN ESTA FUNCION GUARDA LAS %%%
%%% POSICIONES ABSOLUTAS DE ESOS NODOS EN LA VARIABLES 'POS' %%%
%%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%%
%%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%%
%%%           i-1,j %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%%
%%%           | %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%%
%%%           | %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%%
%%%   i,j-1 -----i,j-----i,j+1 %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%%
%%%           | %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%%
%%%           | %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%%
%%%           i,j+1 %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [nb,uvnb,C,C2] = neighboring(M,U,V,i,j,intm)
if nargin<6, intm='linear'; end

[n,m]= size(M);
I=eye(5);

r = reshape(1:1:n*m,n,m); % coonvertir matriz a vector 1D

if nargin<3, [i,j] = find(r == i); end

if i>1 && i<n && j>1 && j<m
nb = neighbor(M,i,j);
C = diag(neighbor(r,i,j))*I;

else
MA=zeros(n+2,j+m); Mr=zeros(n+2,j+m);
MA(2:n+1,2:m+1)=M; Mr(2:n+1,2:m+1)=r;
nb = neighbor(MA,i+1,j+1);

```

```

    C = diag(neighbor(Mr,i+1,j+1))*I;
end

C=diag(C)'; % vector de posiciones cardinales
% si alguna de las componentes de C es cero quiere decir que no hay vecino
% cardinal para este nodo

% si la matriz de las variables coincide con el mallado de U
if all(size(M)==size(U))
if (C(2)~=0 && C(4)~=0)
    unb=[ interp1(1:m,U(i,:),j-1/2,intm)
interp1(1:m,U(i,:),j+1/2,intm)];
elseif (C(2)==0 && C(4)~=0)
    unb=[ 0 interp1(1:m,U(i,:),j+1/2,intm)];
elseif (C(2)~=0 && C(4)==0)
    unb=[ interp1(1:m,U(i,:),j-1/2,intm) 0];
end

if (C(1)~=0 && C(5)~=0)
    vnb=[ interp1(1:m-1,V(i,:),j-1/2,intm) interp1(1:m-1,V(i+1,:),j-
1/2,intm)];
elseif (C(1)==0 && C(5)~=0)
    vnb=[ 0 interp1(1:m-1,V(i+1,:),j-1/2,intm)];
elseif (C(1)~=0 && C(5)==0)
    vnb=[ interp1(1:m-1,V(i,:),j-1/2,intm) 0];
end

rM=reshape(1:n*(m-1),n,m-1);
C2=[rM(i,j-1) rM(i,j)];
% si la matriz de las variables coincide con el mallado de V
elseif all(size(M)==size(V))

if (C(2)~=0 && C(4)~=0)
    unb=[ interp1(1:n-1,U(:,j),i-1/2,intm) interp1(1:n-1,U(:,j+1),i-
1/2,intm)];
elseif (C(2)==0 && C(4)~=0)
    unb=[ 0 interp1(1:n-1,U(:,j+1),i-1/2,intm)];
elseif (C(2)~=0 && C(4)==0)
    unb=[ interp1(1:n-1,U(:,j),i-1/2,intm) 0];
end

if (C(1)~=0 && C(5)~=0)
    vnb=[ interp1(1:n,V(:,j),i-1/2,intm)
interp1(1:n,V(:,j),i+1/2,intm)];
elseif (C(1)==0 && C(5)~=0)
    vnb=[ 0 interp1(1:n,V(:,j),i+1/2,intm) ];
elseif (C(1)~=0 && C(5)==0)
    vnb=[ interp1(1:n,V(:,j),i-1/2,intm) 0];
end

rM=reshape(1:(n-1)*m,n-1,m);
C2=[rM(i-1,j) rM(i,j)];

else% si la matriz de las variables coincide con el mallado principal
unb=[ U(i,j) U(i,j+1) ];
vnb=[ V(i,j) V(i+1,j) ];

ru=reshape(1:n*(m+1),n,m+1);

```

```

rv=reshape(1:(n+1)*m,n+1,m);
    C2 = [rv(i,j) ru(i,j) 0 ru(i,j+1) rv(i+1,j) ];
    C2 = C2.*(C>0);
end
uvnb = [vnb(1) unb(1) 0 unb(2) vnb(2)].*(C>0);
end
function X = neighbor(A,i,j)
X = [ A(i-1,j)  A(i,j-1)  A(i,j)  A(i,j+1)  A(i+1,j) ];
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% FUNCIÓN QUE REALIZA LA RESOLUCIÓN DEL SISTEMA DE ECUACIONES POR EL  %%
%% MÉTODO SIMPLE (SEMI IMPLICIT PRESSURE LINKED EQUATIONS)          %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function
[U1,V1,P1,T1,residual]=SIMPLE(Tguess,Pguess,Uguess,Vguess,T0,P0,U0,V0,constant
s,options,residual)
if nargin < 10
intm = 'spline';  scheme='upwind';
    MaxIt=500; TolT=.00001; TolP=.00001; TolU=.00001; TolV=.00001;
else
intm=options.InterpMethod; scheme=options.Scheme;
    MaxIt=options.MaxIt;
    TolT=options.TolT; TolP=options.TolP;
    TolU=options.TolU; TolV=options.TolV;
end
DimT=size(Tguess);  DimU =size(Uguess);  DimV =size(Vguess);
dt=constants.dt;    dx=constants.dx;    dy=constants.dy; dz=constants.dz;
Tt=constants.Tt;    Tb=constants.Tb;
q0 = constants.q0;  g = -constants.g ;

[Duu,Dvv,DTu,DTv,Fum,Fvm,Fu,Fv,Ax,Ay,rho0,Cp]= flux(T0,P0,U0,V0,constants);

It=0; cond = true;

while cond && It<MaxIt

[Duu,Dvv,DTu,DTv,Fum,Fvm,Fu,Fv,Ax,Ay,rho,Cp] =
flux(Tguess,Pguess,Uguess,Vguess,constants);

%% -----resolver Ustar-----
du=Uguess*0;
% Resolver Ustar para todos los nodos, congelar V,P,T

N=DimU(1)*DimU(2); % numero total de nodos
n=1; A=zeros(N,N); bU=zeros(N,1); % inicializar variables
for j=1:DimU(2) %recorrer columnas o la variable "x"
for i=1:DimU(1) %recorrer filas o la variable "y"
if j==1 || j==DimU(2) % condiciones de borde para "U" en las paredes
izquierda y derecha
    A(n,n)= 1;
bU(n) = 0; % la velocidad perpendicular vale cero
else
    [Unb,Fnb ] = neighboring(Uguess,Fu ,Fv ,i,j,intm); % calcular
flujos masicos vecinos

```

```

[Unb,Dnb,nb] = neighboring(Uguess,Duu,Dvv,i,j,intm); % calcular flujos
difusivos vecinos
DtTerm = (rho0.U(i,j)/dt)*dx*dy*dz; % calcular termino estado
no-estacionario
anb = aneighbor(Fnb,Dnb,DtTerm,scheme); % calcular coeficientes vecinos a =
f(D,F)
A=incm(A,nb,anb,n); % insertar estos coeficientes en la
matriz "A"
% calcular el termino independiente "b"
bU(n) = (Pguess(i,j-1) - Pguess(i,j))*Ay + ( rho0.U(i,j)*U0(i,j)/dt
)*dx*dy*dz;
du(i,j) = Ay./anb(3); % calcular coeficientes para la ecuacion de continuidad
end
n=n+1; % siguiente nodo
end
end
Ustar=A\bU; % resolver "U*"
Ustar=reshape(Ustar,DimU(1),DimU(2)); % redimensionar

%% -----resolver Vstar-----
dv=Vguess*0;
% Resolver Vstar para todos los nodos, congelar U,P,T
N=DimV(1)*DimV(2); % numero total de nodos
n=1; A=zeros(N,N); bV=zeros(N,1);
for j=1:DimV(2) %recorrer columnas o la variable "x"
for i=1:DimV(1) %recorrer filas o la variable "y"
if i==1 || i==DimV(1) % condiciones de borde para "U" en las paredes
arriba y abajo
A(n,n)= 1;
bV(n) = 0; % la velocidad perpendicular vale cero
else
[Vnb,Fnb ] = neighboring(Vguess,Fu ,Fv ,i,j,intm);
[Vnb,Dnb,nb] = neighboring(Vguess,Duu,Dvv,i,j,intm);
DtTerm = (rho0.V(i,j)/dt)*dx*dy*dz ;
anb = aneighbor(Fnb,Dnb, DtTerm,scheme);
A=incm(A,nb,anb,n);
bV(n) = ( Pguess(i,j) - Pguess(i-1,j) )*Ax + ...
( rho0.V(i,j)*V0(i,j)/dt + rho.V(i,j)*g )*dx*dy*dz;
dv(i,j) = Ax./anb(3);
end
n=n+1;
end
end
Vstar=A\bV;
Vstar=reshape(Vstar,DimV(1),DimV(2));
[Duu,Dvv,DTu,DTv,Fum,Fvm,Fu,Fv,Ax,Ay,rho,Cp] =
flux(Tguess,Pguess,Ustar,Vstar,constants);

%% -----resolver Pprime-----

% Resolver P' para todos los nodos, congelar U,V,T
N=DimT(1)*DimT(2); % numero total de nodos
n=1; A=zeros(N,N); bP=zeros(N,1);
for j=1:DimT(2) %recorrer columnas o la variable "x"
for i=1:DimT(1) %recorrer filas o la variable "y"
if i==1 && j==1 % condicion de borde para p, donde p es conocida en i =1
A(n,n)=1;
bP(n) = 0; % p' = 0 en los sitios donde se conoce la presion
else

```

```

                [Pnb,Fmnb    ] = neighboring(Pguess,Fum,Fvm,i,j,intm); % hallar
rho*Area de vecinos
                [Pnb,dnb    ] = neighboring(Pguess,du ,dv ,i,j,intm); % hallar
coeficientes d- vecinos
                [Pnb,Fnb,nb] = neighboring(Pguess,Fu ,Fv ,i,j,intm); % hallar
flujos masicos vecinos
anb = dnb.*Fmnb; % calcular "a" vecinos
anb(3) = sum(anb); % calcular "aP"
                A=incm(A,nb,anb,n); % insertar coeficientes en la matriz
signF = [1 -1 0 1 -1]; % signos para los flujos vecinos

% hallar el termino independiente "b" que es la misma ecuacion
% de continuidad evaluado en Ustar y Vstar
bP(n) = ( -(rho.T(i,j) - rho0.T(i,j))*dx*dy*dz/dt - sum(signF.*Fnb) );
end
                n=n+1; % siguiente nodo
end
end
Pprime=A\bP; % resolver sistema para P'
Pprime=reshape(Pprime,DimT(1),DimT(2)); % redimensionar

%% -----corregir U,V,P-----
DeltaP_u=Ustar*0;
for i=1:DimU(1) %recorrer filas o la variable "y"
for j=2:DimU(2)-1 %recorrer columnas o la variable "x"
                DeltaP_u(i,j) = Pprime(i,j-1) - Pprime(i,j); % calcular delta P para
este nodo en "u"
end
end

DeltaP_v=Vstar*0;
for i=2:DimV(1)-1 %recorrer filas o la variable "y"
for j=1:DimV(2) %recorrer columnas o la variable "x"
                DeltaP_v(i,j) = Pprime(i,j) - Pprime(i-1,j); % calcular delta P para
este nodo en "v"
end
end

U1 = Ustar + du.*DeltaP_u; % corregir "u" mediante ue = ue* + de*(PP - PE)
V1 = Vstar + dv.*DeltaP_v; % corregir "v" mediante vn = vn* + dn*(PP - PN)
P1 = Pprime + Pguess; % corregir "P" mediante P = P' + Pguess

% actualizar flujos másicos y difusivos
[Duu,Dvv,DTu,DTv,Fum,Fvm,Fu,Fv,Ax,Ay,rho,Cp] =
flux(Tguess,P1,U1,V1,constants);

%% -----resolver T-----

% Resolver T para todos los nodos, congelar U,V,P
N=DimT(1)*DimT(2); % numero total de nodos
n=1; A=zeros(N,N); bT=zeros(N,1);

for j=1:DimT(2) %recorrer columnas o la variable "x"
for i=1:DimT(1) %recorrer filas o la variable "y"
if i==1
                [Tnb,Fnb    ] = neighboring(Tguess,Fu ,Fv ,i,j,intm);
                [Tnb,Dnb,nb] = neighboring(Tguess,DTu,DTv,i,j,intm);
DtTerm = (rho0.T(i,j)/dt )*dx*dy*dz + 2*DTv(i,j);

```

```

anb = aneighbor(Fnb,Dnb, DtTerm ,scheme);
    A=incm(A,nb,anb,n);
bT(n) = rho0.T(i,j)*T0(i,j)*dx*dy*dz/dt + 2*DTv(i,j)*Tt;
elseif i==DimT(1) && constants.IsotBound %-----
%-----
    [Tnb,Fnb ] = neighboring(Tguess,Fu ,Fv ,i,j,intm);
    [Tnb,Dnb,nb] = neighboring(Tguess,DTu,DTv,i,j,intm);
    DtTerm = (rho0.T(i,j)/dt )*dx*dy*dz + 2*DTv(i,j);
anb = aneighbor(Fnb,Dnb, DtTerm ,scheme);
    A=incm(A,nb,anb,n);
    bT(n) = rho0.T(i,j)*T0(i,j)*dx*dy*dz/dt + 2*DTv(i,j)*Tb +
q0(i,j)*Ax/(Cp(i,j));
%-----
%-----
else
    [Tnb,Fnb ] = neighboring(Tguess,Fu ,Fv ,i,j,intm);
    [Tnb,Dnb,nb] = neighboring(Tguess,DTu,DTv,i,j,intm);
    DtTerm = (rho0.T(i,j)/dt )*dx*dy*dz;
anb = aneighbor(Fnb,Dnb, DtTerm ,scheme);
    A=incm(A,nb,anb,n);
    bT(n) = rho0.T(i,j)*T0(i,j)*dx*dy*dz/dt + q0(i,j)*Ax/(Cp(i,j));
end
    n=n+1;
end
end
T1=A\bT;
T1=reshape(T1,DimT(1),DimT(2));

%% -----mostrar informacion de convergencia-----
clc;
fprintf('Iteracion = %i , Relacion de convergencia ec. continuidad =
%.4f\n',It,norm(bP));

DB(It+1) = norm(bP);
tt      = residual.t:dt/It:residual.t+dt;
DT(It+1) = max(max(abs(Tguess-T1)))/max(max(abs(Tguess)));
DP(It+1) = max(max(abs(Pguess-P1)))/max(max(abs(Pguess)));
DU(It+1) = max(max(abs(Uguess-U1)))/max(max(abs(Uguess)));
DV(It+1) = max(max(abs(Vguess-V1)))/max(max(abs(Vguess)));

subplot(2,1,1); cla;
plot(0:It,DT/DT(1), 'r-.'); hold on;
plot(0:It,DP/DP(1), 'b'); hold on;
plot(0:It,DU/DU(1), 'g'); hold on;
plot(0:It,DV/DV(1), 'c'); hold on;
plot(0:It,DB/DB(1), 'k');
legend('T','P','U','V','continuity');
set(gca,'YScale','log'); xlabel('iteration');

subplot(2,1,2); cla;
plot([residual.rt tt],[residual.rT DT/DT(1)], 'r-.'); hold on;
plot([residual.rt tt],[residual.rP DP/DP(1)], 'b'); hold on;
plot([residual.rt tt],[residual.rT DU/DU(1)], 'g'); hold on;
plot([residual.rt tt],[residual.rV DV/DV(1)], 'c'); hold on;
plot([residual.rt tt],[residual.rT DB/DB(1)], 'k'); hold on;
set(gca,'YScale','log'); xlabel('Time [sec]');
gf= getframe; clear gf;

```

```

%% -----preparar a la siguiente iteracion-----
ADT=max(max(abs(Tguess-T1)))/max(max(abs(Tguess)))>TolT; % verificar
tolerancia para T
ADP=max(max(abs(Pguess-P1)))/max(max(abs(Pguess)))>TolP; % verificar
tolerancia para P
ADU=max(max(abs(Uguess-U1)))/max(max(abs(Uguess)))>TolU; % verificar
tolerancia para "u"
ADV=max(max(abs(Vguess-V1)))/max(max(abs(Vguess)))>TolV; % verificar
tolerancia para "v"

cond = ADT || ADP || ADU || ADV; % verificar todas las tolerancias al mismo
tiempo
Tguess=T1; % actualizar variables para la siguiente iteracion
Pguess=P1;
Uguess=U1;
Vguess=V1;
It=It+1;

end
residual.rt=[residual.rt tt];
residual.rT=[residual.rT DT/DT(1)] ;
residual.rP=[residual.rP DP/DP(1)];
residual.rU=[residual.rU DU/DU(1)];
residual.rV=[residual.rV DV/DV(1)];
residual.rb=[residual.rb DB/DB(1)];

end

%% -----funcion para insertar "a" vecinos en la matriz "A"-----
function A=incm(A,nb,a,i) % Insert Neighboring Coefficients in Matrix
for j = 1:5
if nb(j)~=0
if j~=3
A(i,nb(j)) = -a(j);
else
A(i,nb(j)) = a(j);
end
end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% FUNCIÓN QUE REALIZA LA RESOLUCIÓN DEL SISTEMA DE ECUACIONES POR EL %%
%% MÉTODO DE ECUACIONES ACOPLADAS RESUELTAS DE MANERA SIMULTANEA %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function
[U1,V1,P1,T1,residual]=SolveSystem(Tguess,Pguess,Uguess,Vguess,T0,P0,U0,V0,con
stants,options,residual)
if nargin < 10
intm = 'spline'; scheme='upwind';
MaxIt=500; TolT=.00001; TolP=.00001; TolU=.00001; TolV=.00001;
else
intm=options.InterpMethod; scheme=options.Scheme;
MaxIt=options.MaxIt;
TolT=options.TolT; TolP=options.TolP;

```

```

    TolU=options.TolU; TolV=options.TolV;
end

DimT=size(Tguess); DimU =size(Uguess); DimV =size(Vguess);
dt=constants.dt; dx=constants.dx; dy=constants.dy; dz=constants.dz;
q0 = constants.q0; Tt=constants.Tt; Tb=constants.Tb;
g = -constants.g ;

[Duu,Dvv,DTu,DTv,Fum,Fvm,Fu,Fv,Ax,Ay,rho0,Cp]= flux(T0,P0,U0,V0,constants);
It=0; cond=true;
while cond && It<MaxIt

[Duu,Dvv,DTu,DTv,Fum,Fvm,Fu,Fv,Ax,Ay,rho,Cp] =
flux(Tguess,Pguess,Uguess,Vguess,constants);

NU=DimU(1)*DimU(2); % numero total de nodos
NV=DimV(1)*DimV(2); % numero total de nodos
NP=DimT(1)*DimT(2); % numero total de nodos
N = NU+NV+NP;
n=1;
AU=zeros(N,NU); b=zeros(N,1); % inicializar variables
AV=zeros(N,NV);
AP=zeros(N,NP);

ru=reshape(1:DimU(1)*DimU(2),DimU(1),DimU(2));
for j=1:DimU(2) %recorrer columnas o la variable "x"
for i=1:DimU(1) %recorrer filas o la variable "y"
if j==1 || j==DimU(2) % condiciones de borde para "U" en las paredes
izquierda y derecha
AU(n,ru(i,j))=1;
else
[Unb,Fnb ] = neighboring(Uguess,Fu ,Fv ,i,j,intm); %
calcular flujos masicos vecinos
[Unb,Dnb,nb,Pnb] = neighboring(Uguess,Duu,Dvv,i,j,intm); %
calcular flujos difusivos vecinos
DtTerm = (rho0.U(i,j)/dt)*dx*dy*dz; % calcular termino estado no-
estacionario
anb = aneighbor(Fnb,Dnb,DtTerm,scheme); % calcular coeficientes vecinos a =
f(D,F)
AU=incm(AU,nb,anb,n); % insertar estos coeficientes en la matriz "A"
AP(n,Pnb)=[-1 1]*Ay; % insertar coeficientes de presiones
b(n) = ( rho0.U(i,j)*U0(i,j)/dt )*dx*dy*dz; % calcular el termino
independiente "b"
end
n=n+1; % siguiente nodo
end
end

rv=reshape(1:DimV(1)*DimV(2),DimV(1),DimV(2));
for i=1:DimV(1) %recorrer columnas o la variable "x"
for j=1:DimV(2) %recorrer filas o la variable "y"
if i==1 || i==DimV(1) % condiciones de borde para "U" en las paredes
arriba y abajo
AV(n,rv(i,j))=1;
else
[Vnb,Fnb ] = neighboring(Vguess,Fu ,Fv ,i,j,intm);
[Vnb,Dnb,nb,Pnb] = neighboring(Vguess,Duu,Dvv,i,j,intm);
DtTerm = (rho0.V(i,j)/dt)*dx*dy*dz ;

```

```

anb = aneighbor(Fnb,Dnb, DtTerm,scheme);
      AV=incm(AV,nb,anb,n);
AP(n,Pnb)=[1 -1]*Ax; % calcular el termino independiente
"b"
b(n) = ( rho0.V(i,j)*V0(i,j)/dt + rho.V(i,j)*g ) *dx*dy*dz;
end
      n=n+1; % siguiente nodo
end
end

rp=reshape(1:DimT(1)*DimT(2),DimT(1),DimT(2));
for j=1:DimT(2) %recorrer columnas o la variable "x"
for i=1:DimT(1) %recorrer filas o la variable "y"
if i==1 && j==1 % condiciones de borde para "U" en las paredes arriba y
abajo
AP(n,rp(i,j))=1;
b(n) = constants.Pt;
else
      [Pnb,Fmnb,nb,unb] = neighboring(Pguess,Fum,Fvm,i,j,intm); %
hallar rho*Area de vecinos
signF = [1 -1 0 1 -1]; % signos para los flujos vecinos
anb=signF.*Fmnb;
      AU=incm(AU,unb([2 4]),anb([2 4]),n);
AV=incm(AV,unb([1 5]),anb([1 5]),n);
b(n) = -(rho.T(i,j) - rho0.T(i,j))*dx*dy*dz/dt;
end
      n=n+1; % siguiente nodo
end
end

A=[AU AV AP];

X=A\b;
U1=reshape(X(1:NU),DimU(1),DimU(2));
V1=reshape(X(NU+1:NU+NV),DimV(1),DimV(2));
P1=reshape(X(NU+NV+1:N),DimT(1),DimT(2));

[Duu,Dvv,DTu,DTv,Fum,Fvm,Fu,Fv,Ax,Ay,rho,Cp] =
flux(Tguess,P1,U1,V1,constants);
% Resolver T para todos los nodos, congelar U,V,P
N=DimT(1)*DimT(2); % numero total de nodos
n=1; AT=zeros(N,N); bT=zeros(N,1);

for j=1:DimT(2) %recorrer columnas o la variable "x"
for i=1:DimT(1) %recorrer filas o la variable "y"
if i==1
      [Tnb,Fnb ] = neighboring(Tguess,Fu ,Fv ,i,j,intm);
      [Tnb,Dnb,nb] = neighboring(Tguess,DTu,DTv,i,j,intm);
      DtTerm = (rho0.T(i,j)/dt ) *dx*dy*dz + 2*DTv(i,j);
anb = aneighbor(Fnb,Dnb, DtTerm ,scheme);
      AT=incm(AT,nb,anb,n);
bT(n) = rho0.T(i,j)*T0(i,j)*dx*dy*dz/dt + 2*DTv(i,j)*Tt;
elseif i==DimT(1) && constants.IsotBound %-----
-----
      [Tnb,Fnb ] = neighboring(Tguess,Fu ,Fv ,i,j,intm);
      [Tnb,Dnb,nb] = neighboring(Tguess,DTu,DTv,i,j,intm);
      DtTerm = (rho0.T(i,j)/dt ) *dx*dy*dz + 2*DTv(i,j);
anb = aneighbor(Fnb,Dnb, DtTerm ,scheme);

```

```

        AT=incm(AT,nb,anb,n);
        bT(n) = rho0.T(i,j)*T0(i,j)*dx*dy*dz/dt + 2*DTv(i,j)*Tb +
q0(i,j)*Ax/(Cp(i,j));
%-----
---
else
    [Tnb,Fnb ] = neighboring(Tguess,Fu ,Fv ,i,j,intm);
    [Tnb,Dnb,nb] = neighboring(Tguess,DTu,DTv,i,j,intm);
    DtTerm = (rho0.T(i,j)/dt )*dx*dy*dz;
    anb = aneighbor(Fnb,Dnb, DtTerm ,scheme);
    AT=incm(AT,nb,anb,n);
    bT(n) = rho0.T(i,j)*T0(i,j)*dx*dy*dz/dt + q0(i,j)*Ax/(Cp(i,j));
end
    n=n+1;
end
end

T1=AT\bT;
T1=reshape(T1,DimT(1),DimT(2));

%% -----mostrar informacion de convergencia-----
fprintf('It = %i , AbsTolT = %.6f, AbsTolP = %.6f, AbsTolU = %.6f, AbsTolV =
%.6f\n',...
It,max(max(abs(Tguess-T1))), max(max(abs(Pguess-P1))),...
max(max(abs(Uguess-U1))), max(max(abs(Vguess-V1))));

tt      = residual.t:dt/It:residual.t+dt;
DT(It+1) = max(max(abs(Tguess-T1)))/max(max(abs(Tguess)));
DP(It+1) = max(max(abs(Pguess-P1)))/max(max(abs(Pguess)));
DU(It+1) = max(max(abs(Uguess-U1)))/max(max(abs(Uguess)));
DV(It+1) = max(max(abs(Vguess-V1)))/max(max(abs(Vguess)));

subplot(2,1,1); cla;
plot(0:It,DT/DT(1),'r-.'); hold on;
plot(0:It,DP/DP(1),'b'); hold on;
plot(0:It,DU/DU(1),'g'); hold on;
plot(0:It,DV/DV(1),'k');
legend('T','P','U','V');
set(gca,'YScale','log'); xlabel('iteration');

subplot(2,1,2); cla;
plot([residual.rt tt],[residual.rT DT/DT(1)],'r-.'); hold on;
plot([residual.rt tt],[residual.rP DP/DP(1)],'b'); hold on;
plot([residual.rt tt],[residual.rU DU/DU(1)],'g'); hold on;
plot([residual.rt tt],[residual.rV DV/DV(1)],'k');
set(gca,'YScale','log'); xlabel('Time [sec]');
gf= getframe; clear gf;

%% -----preparar a la siguiente iteracion-----

ADT=max(max(abs((Tguess-T1)))>TolT; % verificar tolerancia para T
ADP=max(max(abs((Pguess-P1)))>TolP; % verificar tolerancia para P
ADU=max(max(abs(Uguess-U1)))>TolU; % verificar tolerancia para "u"
ADV=max(max(abs(Vguess-V1)))>TolV; % verificar tolerancia para "v"

```

```

cond = ADT || ADP || ADU || ADV; % verificar todas las tolerancias
Tguess=T1; % actualizar variables para la siguiente iteracion
Pguess=P1;
Uguess=U1;
Vguess=V1;
It=It+1;

end
residual.rT=[residual.rT tt];
residual.rT=[residual.rT DT/DT(1)] ;
residual.rP=[residual.rP DP/DP(1)];
residual.rU=[residual.rU DU/DU(1)];
residual.rV=[residual.rV DV/DV(1)];

end

% -----funcion para insertar "a" vecinos en la matriz "A"-----
function A=incm(A,nb,a,i) % Insert Neighboring Coefficients in Matrix
if length(nb)<3
for j = 1:2, if nb(j)~=0, A(i,nb(j)) = a(j); end
end
else
for j = 1:5
if nb(j)~=0
if j~=3
A(i,nb(j)) = -a(j);
else
A(i,nb(j)) = a(j);
end
end
end
end
end
end

```

### Módulo de gráficos:

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% FUNCIÓN QUE REALIZA LA REPRESENTACIÓN GRÁFICA OBTENIDA DE RESOLVER   %%
%% LAS ECUACIONES                                     %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function plotdata(U,V,T,P,t,PlotType,constants,options)
switch PlotType
case'transient'
if nargin<8
options=100;
end
transient(U,V,T,P,t,options);
case'stream'
if nargin<8
options.StaticColorbar=true;
options.StreamPlot ='streamslice';
options.FrameSpeed=.01; % .5
options.MaxTime=100;
end
figure;

```

```

astream(U,V,T,P,t,constants,options);
case'contour'
contourplot(U,V,T,P,t,constants,options)
case'properties'
propertiesplot(U,V,T,P,t,constants,options)
end

end

function contourplot(U,V,T,P,t,constants,time)
if time>=max(t)
    k= length(t);
else
    k = max(find(t<=time));
end
x=constants.x; y=constants.y;
y=-y(end:-1:1);
[Us,Vs,Ts,Ps]=postprocess(U(:,:,k),V(:,:,k),T(:,:,k),P(:,:,k),constants);

subplot(1,2,1); contourf(x,y,Ps(end:-1:1,:)/1e6); colorbar; title('Pressure
[MPa]')
axis([min(x) max(x) min(y) max(y)])
subplot(1,2,2); contourf(x,y,Ts(end:-1:1,:)); colorbar; title('Temperature
[°C]')
%colormap(jet); shading interp; view(0,90);
axis([min(x) max(x) min(y) max(y)])
end

function propertiesplot(U,V,T,P,t,constants,time)
addpath([cd '/waterprop']);
if time>=max(t)
    k= length(t);
else
    k = max(find(t<=time));
end
x=constants.x; y=constants.y;
y=-y(end:-1:1);
[Us,Vs,Ts,Ps]=postprocess(U(:,:,k),V(:,:,k),T(:,:,k),P(:,:,k),constants);
Ps(1,:)=min(Ps(2,:));
[Cp,rho,mu,k,Psat]=ComputeProperties(Ts,Ps);

subplot(2,2,1); contourf(x,y,rho(end:-1:1,:)); colorbar; title('Density
[kg/m^3]')
axis([min(x) max(x) min(y) max(y)])
subplot(2,2,2); contourf(x,y,mu(end:-1:1,:)); colorbar; title('Viscosity
[kg/m*s]')
axis([min(x) max(x) min(y) max(y)])
subplot(2,2,3); contourf(x,y,k(end:-1:1,:)); colorbar; title('Thermal
conductivity [W/m*°C]')
axis([min(x) max(x) min(y) max(y)])
subplot(2,2,4); contourf(x,y,Psat(end:-1:1,:)); colorbar; title('Vapour
pressure [MPa]')
axis([min(x) max(x) min(y) max(y)])
end

function astream(U,V,T,P,t,constants,options)

%mov = avifile('exempl.avi');

```

```

%mov.Quality = 70;
%mov.Fps = 2;

Tmax=max(max(max(T))); Tmin=min(min(min(T)));

for k=4:length(t)
    x=constants.x; y=constants.y; y=-y(end:-1:1);
    [Us,Vs,Ts,Ps]=postprocess(U(:,:,k),V(:,:,k),T(:,:,k),P(:,:,k),constants);
    cla;
    switch options.StreamPlot
    case 'streamslice'
        streamslice(x,y,Us(end:-1:1,:),Vs(end:-1:1,:));
    case 'quiver'
        quiver(x,y,Us(end:-1:1,:),Vs(end:-1:1,:));
    end

    hold on;
    spdg(:,:,1)= Ts(end:-1:1,:);
    spdg(:,:,2)= 0* Ts(end:-1:1,:);
    slice(x,y,-1:0,spdg,0,0,-1);
    colormap(jet); shading interp;
    view(0,90); hold on;
    colorbar;
    if options.StaticColorbar, caxis([Tmin 0.98*max(max(T(:,:,k)))+.02*Tmax ]);
    end

    axis([min(x) max(x) min(y) max(y)])
    Pp= getframe; uiwait(gcf, options.FrameSpeed*log(t(k)^2+1));
    %mov = addframe(mov,Pp);
    clearPp;
    if t(k)>max(t)*(options.MaxTime/100), break; end
    title(['t = ' num2str(t(k)/(3600*24)) ' [day]'])
    end
    %mov = close(mov);
end

function transient(U,V,T,P,t,TimeScale)
n=size(T(:,:,1)); n=n(end:-1:1); t=t/(3600*24);
[i,j] = gridshow(n,[1 1]);

P=Array3DtoTime(P,i,j,'P')/1e6;
T=Array3DtoTime(T,i,j,'T');
U=Array3DtoTime(U,i,j,'U');
V=Array3DtoTime(V,i,j,'V');
figure;
tmax=(TimeScale/100)*max(t);
subplot(4,1,1); plot(t,P,'k'); ylabel('Pressure [MPa]');
axis([0 tmax min(P)-.0001 max(P)+.0001]);

subplot(4,1,2); plot(t,T,'k'); ylabel('Temperature [°C]');
axis([0 tmax min(T) max(T)+.1]); %set(gca,'XScale','log')

subplot(4,1,3); plot(t,U,'k'); ylabel('u-velocity [m/s]');
axis([0 tmax min(U) max(U)]);

subplot(4,1,4); plot(t,V,'k'); ylabel('v-velocity [m/s]');

```

```

axis([0 tmax min(V) max(V)]);
xlabel('Time [day]');
end

function M0=Array3DtoTime(M,i,j,var)
[ n m p]=size(M);
for k=1:p
switch var
case 'P'
Ms=M(:,:,k);
case 'T'
Ms=M(:,:,k);
case 'U'
Ms=uinterp(M(:,:,k));
case 'V'
Ms=uinterp(M(:,:,k)')';
end
M0(k)=Ms(i,j);
end
end

function [U,V,T,P]=postprocess(U,V,T,P,constants)
nx=constants.nx; ny=constants.ny;
M=zeros(ny+2,nx+2);
Us=uinterp(U);
U=M;
U(2:end-1,2:end-1)=Us;

Vs=uinterp(V')';
V=M;
V(2:end-1,2:end-1)=Vs;

if constants.IsotBound
T(ny+1,:)=constants.Tb;
else
T(ny+1,:)=spline(.5:ny-.5,T',ny)';
end

Tm=M;
Tm(2:end,2:end-1)=T;
Tm(2:end,1)=T(:,1);
Tm(2:end,end)=T(:,end);
Tm(end,1)=(Tm(end-1,1)+Tm(end,2))/2;
Tm(end,end)=(Tm(end-1,end)+Tm(end,end-1))/2;
Tm(1,:)=constants.Tt;
T=Tm;

Pm=M;
Pm(2:end-1,2:end-1)=P;
Pm(2:end-1,1)=P(:,1);
Pm(2:end-1,end)=P(:,end);
Pm(end,2:end-1)=spline(.5:ny-.5,P',ny)';
Pm(1,:)=min(P(1,:))-constants.g*1000*constants.dy/2;
Pm(end,1)=Pm(end,2);
Pm(end,end)=Pm(end,end-1);
P=Pm;

end

```

```

function [us,x0]=uinterp(u)
[n,m]=size(u);
xu=0:m-1;
x0=.5:1:m-1.5;
for i =1:n
    us(i,:)=spline(xu,u(i,:),x0);
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% FUNCIÓN QUE REALIZA LA REPRESENTACIÓN DE LA MATRIZ ESCALONADA      %%%
%%% ADICIONALMENTE PERMITE ELEGIR UN PUNTO EN LA MATRÍZ Y OBTENER SUS  %%%
%%% PROPIEDADES EN TODO EL TIEMPO DE LA SIMULACIÓN                    %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [xm,ym,xu,yu,xv,yv] = gridshow(n,d,gridtype,marker,color)
if nargin<3, gridtype=[1 1 1]; marker=[2 1 1]; color=[1 2 3]; end

nx=n(1); ny=n(2); % numero de celdas
xmax=(nx)*d(1); ymax=(ny)*d(2);
xm=d(1)/2:d(1):xmax-d(1)/2; ym=d(2)/2:d(2):ymax-d(2)/2;
xv=d(1)/2:d(1):xmax-d(1)/2; yv=0:d(2):ymax;
xu=0:d(1):xmax; yu=d(2)/2:d(2):ymax-d(2)/2;

if nargin==0
    plotgrid(nx,ny,d,xm,ym,xu,yu,xv,yv,xmax,ymax,gridtype,marker,color);
elseif nargin==2
    plotgrid(nx,ny,d,xm,ym,xu,yu,xv,yv,xmax,ymax,gridtype,marker,color);
    [x,y]=ginput(1);
    j=1:nx;
    i=1:ny;
    J=max(j(xu<=x));
    I=max(i(max(yv)-yv>=y));
    cla;
    yd=yv(end:-1:1);
    area([xu(J) xu(J+1)], [yd(I) yd(I)], yd(I+1) , 'FaceColor', [1 .8 1]); hold
on;
    plotgrid(nx,ny,d,xm,ym,xu,yu,xv,yv,xmax,ymax,gridtype,marker,color);
    xm=I; ym=J;
end

end

function plotgrid(nx,ny,d,xm,ym,xu,yu,xv,yv,xmax,ymax,gridtype,marker,color)
gridColor=[ 1 1 1]*215/255;
% generar lineas guias para el grid principal
for i = 1:ny, plot(0:d(1):xmax,ym(i)*ones(1,nx+1), '--', 'Color', gridColor);
hold on;
end

for j = 1:nx, plot(xm(j)*ones(1,ny+1),0:d(2):ymax, '--', 'Color', gridColor);
hold on;
end

if gridtype(1) % grid principal

```

```

for i = 1:ny, plot(xm,ym(i)*ones(1,nx),[colorValue(color(1))
markerValue(marker(1))], 'MarkerSize',6, 'LineWidth',2); hold on; end
for j = 1:nx, plot(xm,ym(i)*ones(1,nx),[colorValue(color(1))
markerValue(marker(1))], 'MarkerSize',6); hold on; end
end

if gridtype(2) % grid para v

for i = 1:ny+1, plot(xv,yv(i)*ones(1,nx),[colorValue(color(2))
markerValue(marker(2))], 'LineWidth',2); hold on; end
end

if gridtype(3) % grid para u

for j = 1:nx+1, plot(xu(j)*ones(1,ny),yu,[colorValue(color(3))
markerValue(marker(3))], 'LineWidth',2); hold on; end
end

axis square;
axis([0 xmax 0 ymax]);
end

function c=colorValue(color)
switch color
case 1, c='k';
case 2, c='b';
case 3, c='r';
case 4, c='g';
case 5, c='c';
end
end
function c=markerValue(marker)
switch marker
case 1, c='.';
case 2, c='o';
case 3, c='*';
case 4, c='x';
end
end

```