UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

"Interfaz gráfica basada en Python para el uso de un simulador de convección termohalina en medios porosos"

TESINA

para obtener el título de

ESPECIALISTA EN EXPLORACIÓN Y APROVECHAMIENTO DE RECURSOS GEOTÉRMICOS

P R E S E N T A

ING. BRENDA LIZETTE DE LA ROSA ESPINOZA

ASESOR

DR. FERNANDO JAVIER GUERRERO MARTÍNEZ

Ciudad Universitaria, CD. MX. 2023.

RESUMEN

Se presenta una interfaz gráfica para el uso de un simulador numérico de convección termohalina en medios porosos (GEOThermohaline) y la visualización de sus resultados. El simulador GEOThermohaline fue desarrollado por Guerrero et al. (2020) e implementado en Fortran 90. Dicho simulador permite evaluar el efecto de la inclinación del medio, el número de Rayleigh (Ra) y la relación de flotabilidad (N) en el transporte por convección termohalina en un medio poroso a lo largo del tiempo (Δt). El fenómeno físico de convección termohalina alude a la inyección del CO_2 en yacimientos geotérmicos, por lo tanto, es relevante estudiar este fenómeno en el contexto del secuestro geológico del CO_2, ya que se considera como una alternativa para la mitigación de gases de efecto invernadero en la atmósfera.

La interfaz gráfica se realiza un programa en Python, un lenguaje de programación que complementa a Fortran 90 con bibliotecas de procesamiento, graficación y visualización de datos. Asimismo, para ejecutar el programa en Python se emplea en un ambiente llamado "Jupyter Notebook", que es una plataforma interactiva basada en la web, que permite ejecutar código, la visualización de datos, escribir texto, etc. El resultado es un cuaderno interactivo que explica de manera general la teoría que sustenta al simulador y que permite al usuario: seleccionar el tipo de solucionador numérico, definir los parámetros de entrada ($Ra, N, \theta, \Delta t$), ejecutar el simulador y visualizar los resultados; todo esto en un mismo ambiente.

De modo que el cuaderno desarrollado flexibiliza el uso, procesamiento y visualización de los resultados del simulador GEOThermohaline. Adicionalmente, se tiene un mayor aprovechamiento de esta herramienta porque todas las funciones para la ejecución, graficación y visualización están en un mismo ambiente de trabajo.

La realización del cuaderno interactivo resalta que contar con herramientas eficientes de pre y post-procesamiento es clave para el mayor aprovechamiento de estas técnicas de análisis y, además, la utilización de distintos lenguajes de programación de forma complementaria es una alternativa para optimizar el análisis numérico. Finalmente, el cuaderno interactivo es también una herramienta de enseñanza ya que brinda un acercamiento practico a los métodos numéricos y las ciencias computacionales.

ABSTRACT

A graphic user interface (GUI) was developed for the use of a numerical simulator of thermohaline convection in porous media and the visualization of its results. The simulator (GEOThermohaline) was developed by Guerrero et al. (2020) and implemented in Fortran 90. It allows the evaluation of effects such as the inclination of the porous medium, the Rayleigh number (Ra), and the buoyancy ratio(N) on the convective transport in the system through the time (Δt). The physical phenomenon of thermohaline convection represents the injection of CO_2 in geothermal reservoirs, which is an important process in the context of the geological secuestration of CO_2, considered nowadays as an alternative for the mitigation of greenhouse gas emissions to the atmosphere.

The GUI proposed here is written in Python, a programming language that complements Fortran 90 with libraries for data processing, plotting, and visualization. Likewise, for the execution of the GUI the environment "Jupyter Notebook" is used, which is an interactive web-based platform that allows the execution of Python code, data visualization, writing text, etc. The result is an interactive notebook that explains in a general way the theory behind the simulator and allows the user to choose options for solutions algorithm, to define the input parameters ($Ra, N, \Theta, \Delta t$), and to run the simulator and visualize the results, all of this in the same environment.

As a result, the GUI adds flexibility to the use, processing, and visualization of results of the GEOThermohaline simulator. Additionally, it allows an efficient workflow since all functions for execution, plotting and visualization are in the same work environment.

The development GUI permits to conclude that having efficient pre- and post-processing tools are key to make the most in the utilization of numerical simulators. In addition, the use of different programming languages in a complementary way is an alternative to optimize numerical analysis. Finally, the interactive notebook is also a teaching tool as it provides a practical approach to numerical methods and computer science.

AGRADECIMIENTOS

Me gustaría expresar mi profunda gratitud a las personas que de diferentes maneras recibí su apoyo en la realización de este trabajo:

A mi supervisor, el Dr. Fernando Guerrero por su increíble paciencia, por su tiempo, y sobre todo por la retroalimentación y apoyo durante todo este proceso.

Al proyecto Macti y al Dr. Luis De la Cruz Salas por brindar la oportunidad y los recursos en la realización de este proyecto.

A los sinodales por sus valiosos comentarios para este trabajo: Dra. Mariana P. Jácome Paz, Dra. Rosa Ma. Prol Ledesma, Dr. Luis De la Cruz Salas e Ing. Martín Carlos Vidal García.

A los todos profesores de la especialidad por su pasión al enseñar, su paciencia y valiosa experiencia que compartían y transmitían en clases. Es invaluable el aprendizaje que adquirí en este año.

A todos mis compañeros de la especialidad por la buena voluntad de apoyarnos mutuamente, por el buen trabajo en equipo que hicimos a pesar de la distancia y por las amistades ahora llevo conmigo.

A mi familia por su apoyo incondicional y amor en todos los momentos de mi vida.

CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

1 INTRODUCCIÓN

1.1 Justificación

El modelado numérico del transporte de masa y calor en medios porosos ha sido de gran importancia en la ingeniería de yacimientos para estudiar el desempeño de estos sistemas durante su aprovechamiento. Debido a que estos procesos de transporte están basados en ecuaciones diferenciales parciales y ecuaciones no lineales, no pueden ser explicados de manera analítica, por lo tanto, el uso de soluciones numéricas y herramientas computacionales se ha vuelto indispensable en la ingeniería de yacimientos.

Un problema actual en el modelado de yacimientos está relacionado con el secuestro geológico del CO_2, tema que ha tomado relevancia recientemente al ser una alternativa para mitigación de gases de efecto invernadero en la atmósfera. Adicionalmente, tiene la ventaja de ser una tecnología adaptable a la ingeniería existente. En el presente trabajo se desarrolla una herramienta de cómputo que contribuya al estudio de este importante tema.

1.2 Objetivo

1.1.1 Objetivo General

Desarrollar una interfaz gráfica en Python que facilite el uso y visualización de resultados de un simulador de convección termohalina en medios porosos.

1.2.1 Objetivos Particulares

1. Desarrollar un ambiente interactivo que permita al usuario definir de forma sencilla los parámetros de entrada del simulador.
2. Programar un sistema de funciones en Python para la lectura de archivos de salida del simulador y para su graficación.
3. Crear un ambiente en Python para un uso integral del simulador: descripción del problema estudiado, opciones de uso y análisis de resultados.

1.3 Antecedentes

1.3.1 Secuestro geológico del CO_2

Debido a la emergencia climática que vivimos actualmente, se han estudiado distintas alternativas para reducir las emisiones de gases de efecto invernadero en la atmósfera. Una de estas alternativas es el secuestro geológico del CO_2, también conocido como CCS (Carbon Capture and Storage por sus siglas en ingles), que consiste en la captura y almacenamiento en formaciones geológicas del CO_2 proveniente de la quema de combustibles fósiles para la generación de energía y de procesos industriales (Freund, 2013).

En la tecnología CCS se propone almacenar permanentemente el CO_2 mediante la inyección del CO_2 en formaciones geológicas profundas; estas formaciones geológicas deben de tener características físicas óptimas para retener grandes cantidades de CO_2 como buena porosidad, alta permeabilidad y capas impermeables en los límites de la formación (Newell & Ilgen, 2018).

La inyección de fluido en el subsuelo es un problema complejo porque se perturban las condiciones de equilibrio en el sistema poroso, cambia la presión de poro y el estado de tensión de la formación geológica. Adicionalmente, se tiene que considerar un sistema bifásico por la presencia de dos fases fluidas distintas (CO_2 supercrítico y salmuera). El reequilibrio del sistema después de la inyección del CO_2 no será lineal; ya que varios procesos se desarrollarán a lo largo de escalas de tiempo muy distintos.

Los procesos a largo plazo en la retención del CO_2 en el subsuelo están principalmente relacionados con mecanismos de disolución y mineralización. Cuando el CO_2 es inyectado, quedará atrapado en la formación, pero con el tiempo se disolverá en la salmuera, formando una solución saturada de CO_2. Debido a la diferencia de densidad que puede existir entre la salmuera de la formación y la salmuera saturada de CO_2, se formará una mezcla convectiva que será el mecanismo que acelerará la disolución del CO_2 en la formación (Ennis-King & Paterson, 2005). Además, la disolución del CO_2 generará reacciones geoquímicas entre la salmuera y los minerales de la formación geológica cambiando el equilibrio geoquímico del

sistema. Estas reacciones químicas y las nuevas condiciones del sistema, con el tiempo, estimularán la precipitación de minerales de carbonato. Sin embargo, estudios recientes sugieren que el proceso de precipitación del carbono puede tardar desde algunos cientos hasta varios miles de años debido a las tasas lentas de mineralización. Aunado a lo anterior, otra perturbación importante que tiene lugar durante la inyección de CO_2 es la existencia de un gradiente térmico en la formación geológica ya que, la densidad y solubilidad del CO_2 en la salmuera serán influenciadas por el cambio de temperatura (Newell & Ilgen, 2018).

Todos estos factores deben ser estudiados y analizados a detalle para que un proyecto de secuestro geológico tenga menores riesgos y que, en consecuencia, se impulse su desarrollo comercial. En la práctica, la industria petrolera ha usado la inyección del CO_2 para mejorar la recuperación del petróleo, y se ha estimado que la capacidad de almacenamiento del CO_2 (únicamente para yacimientos petroleros) es de miles de gigatoneladas, equivalente un rango de 50 a 100 años de emisiones. La experiencia industrial y la evidencia geológica de que el CO_2 puede ser retenido durante millones de años demuestra que el almacenamiento geológico CO_2 es una alternativa eficaz para la mitigación de gases de efecto invernadero en la atmósfera (Bickle, 2009).

1.3.2 Proyectos de secuestro de CO_2 en operación

La industria petrolera fue pionera en la investigación del secuestro geológico del CO_2. A continuación, se presentan los proyectos que tienen décadas implementado la inyección de CO_2 en distintos campos petroleros.

Unidad SACROC

La unidad SACROC (Scurry Area Canyon Reef Operators Committee) se encuentra en el campo Kelly-Snyder, en el condado de Scurry al oeste de Texas, fue de los primeros sitios del mundo donde se realizaron operaciones de inyección del CO_2. Dicharry et al. (1973) publicó la primera evaluación y diseño de inyección de CO_2 para la recuperación mejorada de petróleo en el campo. En este trabajo proponen un esquema de inyección del CO_2 impulsado por agua (WAG: water-alternating-gas). Hicieron estudios de laboratorio para entender comportamiento del desplazamiento del CO_2 en el aceite de la formación y se plantearon patrones de

inyección para asegurar el desplazamiento efectivo del aceite, dichos patrones fueron evaluados con modelos matemáticos.

En el caso de SACROC se ha probado que el método de inyección de agua alternando gas tiene el efecto de redistribuir el CO_2 y el agua en zonas de baja permeabilidad. Sin embargo, para zonas de alta permeabilidad y menos heterogéneas, este esquema de inyección no ha logrado redistribuir los fluidos de manera significativa. Aun así, se considera que la unidad SACROC es uno de los proyectos de inyección de CO_2 más exitosos del mundo (Ghahfarokhi et al., 2016).

Unidad Joffre Viking

Al igual que la unidad SACROC, en Canadá se inició un proyecto de recuperación mejorada de petróleo con CO_2 y, en 1984 comenzó la inyección de CO_2 en la Unidad Terciaria de Joffre Viking en la provincia de Alberta. MacIntyre (1986) examinó procesos, instrumentación, material y consideraciones de operación incorporadas en el diseño del proyecto de la compresión e inyección del CO_2. Concluyó que las instalaciones de inyección del CO_2 requieren el reconocimiento de las características físicas y térmicas únicas de este gas y su corrosividad en presencia de agua.

La Unidad Terciaria de Joffre Viking tenía la producción secundaria completamente cerrada en la década de 1970, sin embargo, la producción aumentó de cero en 1983 a 900 barriles por día en 2000, como respuesta a la inundación de CO_2 miscible. El CO_2 que se inyecta es producido en una planta de etileno (NOVA Chemicals) que históricamente era emitido a la atmósfera. Ahora es capturado y vendido como producto a una compañía petrolera canadiense (antes PennWest, ahora Obsidian Energy) para actividades de recuperación mejorada de petróleo. Después de años de operación de la inyección del CO_2 a la unidad Joffre Viking, se estimó que de 2004 a 2011 se han reducido 39566 toneladas de emisiones CO_2 a la atmósfera (The Prasino Group, 2012).

Campo Sleipner Vest

El campo de gas Sleipner está localizado en el sector noruego del Mar del Norte. Baklid & Korbol (1996) investigaron distintas alternativas para la eliminación del

CO_2 generado en el campo, basadas en la inyección del gas en formaciones geológicas del campo.

Se seleccionó la formación 'Utsira' como la formación objetivo para la inyección del CO_2 y se realizó un estudio de simulación basado en el mapa de contorno estructural de la formación Utsira. Esta simulación se centró en la distribución vertical del CO_2 durante 20 años de inyección y el objetivo fue investigar cómo se distribuiría el CO_2 dentro de la formación.

Los principales resultados de la simulación indicaron que el CO_2 debe inyectarse en el fondo de la formación para asegurar mayor disolución en el agua de la formación; se estimó que la extensión máxima del CO_2 después de 20 años de inyección es de 3 km en cualquier dirección y que más del 18% del CO_2 inyectado sería disuelto en el agua de la formación; y además se debe esperar un aumento en la presión del yacimiento de 30 bar cerca del punto de inyección.

La operación de inyección del CO_2 se llevó a cabo en la formación Utsira y durante 20 años de operación se implementó un extenso programa de monitoreo geofísico con estudios de sísmica de reflexión y gravimetría. Con la experiencia de Sleipner, se ha demostrado que el monitoreo sísmico es la herramienta de monitoreo de inyección del CO_2 más eficiente, debido a la capacidad de cubrir de forma remota un área grande y obtener resultados de alta resolución que pueden detectar pequeños cambios en la saturación del CO_2. Los estudios gravimétricos son un complemento para los casos menos profundos debido a su capacidad para detectar cambios de masa, sin embargo, la principal limitación del monitoreo gravimétrico es la resolución espacial en comparación con la resolución sísmica (Furre et al., 2017).

El proyecto de inyección de CO_2 en Sleipner ha sido un éxito técnico y económico, y el monitoreo geofísico remoto ha demostrado de manera convincente que el CO_2 permanece seguro en la unidad de almacenamiento.

1.4 Secuestro geológico en sistemas geotérmicos

Después de años de investigación (MacIntyre, 1986; Baklid & Korbol, 1996; Zweigel et al., 2004; Bickle, 2009) y la puesta en marcha del secuestro geológico del CO_2 en campos petroleros, se ha considerado que esta tecnología también podría

desarrollarse en sistemas geotérmicos mejorados (EGS por sus siglas en ingles). Los sistemas geotérmicos mejorados son aquellos donde se pretende extraer energía geotérmica mediante la estimulación hidráulica o fracturamiento para generar permeabilidad. Este proceso implica la inyección de fluidos mediante pozos profundos para activar las fracturas o crear nuevas fracturas en la roca caliente. Una vez estimulada la red de fracturas, es posible establecer y mantener la circulación de fluidos a través de éstas, de modo que la energía geotérmica se transmite a la superficie mediante pozos de producción para ser aprovechada (Pruess, 2006).

Los intentos de desarrollar EGS alrededor del mundo han sido implementados usando agua como el fluido de inyección y transmisión de calor; a pesar de que el agua tiene muchas propiedades que la convierten en un medio propicio para este fin, también tiene ciertos inconvenientes. Una propiedad desfavorable del agua es que es un solvente poderoso para muchos minerales de roca, especialmente a temperaturas elevadas. La inyección de agua en las fracturas de roca caliente provoca fuertes efectos de disolución y precipitación que cambian la permeabilidad de la fractura y dificultan mucho la operación de un yacimiento EGS de manera estable (Xu & Pruess, 2004). También es importante considerar que el agua es un bien valioso y puede resultar escaso para algunas zonas del mundo, además, las inevitables pérdidas de agua durante la circulación del fluido pueden representar una grave responsabilidad económica (Pruess, 2006).

Brown (2000) propuso un nuevo concepto de EGS que usaría CO_2 supercrítico como fluido de inyección y transmisión de calor en lugar de agua, y como beneficio adicional lograría el almacenamiento geológico de CO_2. El estudio de Brown (2000) se realizó en las montañas de Jemez en la parte norte-central del estado de Nuevo México, en un sitio de prueba llamado Fenton Hill. En esta locación se hicieron pruebas flujo a largo plazo y los resultados respaldan la existencia y la estabilidad duradera de una región altamente presurizada de roca fracturada, que es relevante para estudiar el secuestro profundo de CO_2 supercrítico (sCO_2).

Brown (2000) presentó un modelo esquemático de la inyección del CO_2 en un sistema geotérmico mejorado (Fig. 1). El sCO_2 se usaría tanto para el fluido de fracturamiento como para el fluido de transporte de calor en los sistemas de extracción de calor profundos. Para el transporte de calor se usaría un circuito

cerrado de tierra, y con el calor contenido en el sCO_2 producido se transferiría a un fluido de trabajo secundario en un intercambiador de calor en la superficie. Este segundo fluido se usaría para impulsar una turbina de expansión en un sistema de generación de energía de ciclo binario. Sin embargo, este intercambiador de calor de ciclo binario sería no convencional debido al cambio de densidad del fluido (y los efectos de presión asociados) del lado del sCO_2 en el intercambiador de calor. La fuerza de la flotación resultante del empleo del sCO_2 como el fluido de transporte de calor se usaría para proporcionar una presión de producción superficial cercana a la presión de inyección, lo que reduciría en gran medida los requisitos de potencia de bombeo una vez que el yacimiento EGS haya sido presurizado inicialmente.

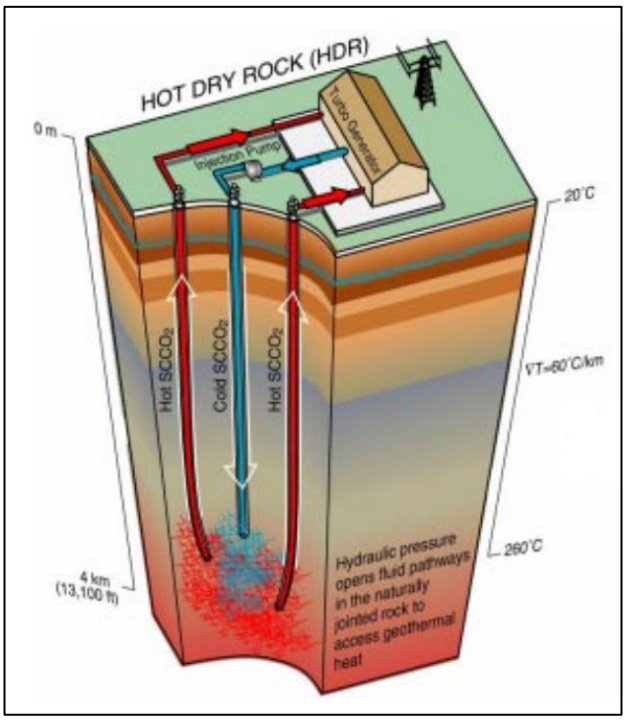

Figura 1*. Modelo esquemático de la inyeccion del* CO_2 *en un sistema geotérmico mejorado (Brown, 2000).*

Brown (2000) notó que el CO_2 tiene propiedades físicas y químicas que serían ventajosas para los sistemas geotérmicos mejorados, como su gran expansividad que generaría diferencias de densidad considerables entre el CO_2 frío en el pozo de inyección y el CO_2 caliente en el pozo de producción, y esto generaría una fuerza de flotación que reduciría el consumo de energía del sistema de circulación de fluidos; su baja viscosidad produciría mayores velocidades de flujo para un gradiente de

presión dado; y es menos efectivo como solvente para los minerales de las rocas, lo que reduciría los problemas de incrustaciones, como la disolución y precipitación de sílice en sistemas a base de agua.

Pruess & Azaroual (2006) sugirieron que el CO_2 es aproximadamente comparable al agua como fluido de transmisión de calor, pero ofrece ventajas considerables en términos de la hidráulica del pozo. Pruess (2006), por su parte, evaluó las propiedades termofísicas del CO_2 y realizó simulaciones numéricas para explorar la dinámica de fluidos y los problemas de transporte de calor, utilizando un modelo homogéneo con una configuración de 5 pozos en un área de 1 km^2. Sus resultados indicaron que debido a la alta expansividad y compresibilidad del CO_2, el gas a condiciones supercríticas generaría fuerzas de flotación más fuertes entre los pozos de inyección y producción. Esto reduciría el consumo energético para el sistema de circulación de fluidos y permitiría una circulación de fluidos adecuada sin bombeo externo. El CO_2 se calentará con la compresión (en el pozo de inyección) y se enfriará con la expansión (en el pozo de producción), lo que dará lugar a cambios de temperatura sustanciales (de 10°C a 25°C). Con su modelo propuesto se encontró que, para una caída de presión total dada entre los pozos de inyección y producción, el CO_2 generará flujos de masa cuatro veces mayores y tasas de extracción de calor 50% mayor que el agua. El autor propone que un trabajo futuro sería explorar el flujo de masa y el comportamiento de la extracción de calor para yacimientos heterogéneos y con diferentes configuraciones de pozos.

Adicionalmente, los sistemas geotérmicos mejorados impulsados por la inyección del CO_2 ofrecen la posibilidad de almacenar geológicamente este gas de efecto invernadero. Dicho almacenamiento podría proporcionar beneficios e incentivos económicos en escenarios de gestión del carbono en los que se imponen impuestos a las emisiones atmosféricas de CO_2, lo que podría proporcionar una fuente adicional de ingresos, mejorando así aún más la economía de los EGS (Pruess, 2006).

1.5 Mecanismo de disolución y convección termohalina

En una formación geológica, es importante estudiar el mecanismo de disolución para establecer el diseño, operación y mantenimiento de una instalación de almacenamiento de CO_2. Al cuantificar la tasa de disolución y comprender los

mecanismos de transporte se puede estimar una escala de tiempo de la captura del CO_2 en una solución acuosa. Distintos autores han hecho estimaciones de escalas de tiempo para la difusión del CO_2 al ser mezclado en una solución acuosa, y concluyeron que el proceso de disolución es muy lento y que la fase de CO_2 puro permanecería prácticamente intacta durante unos 10 000 años, a menos que se produzcan otros procesos (Taheri et al., 2012) (Ennis-King & Paterson, 2005).

El almacenamiento geológico del CO_2 basado en sistemas geotérmicos presenta un mecanismo de disolución distinto, pues al inyectar CO_2 el sistema estaría sometido a dos tipos de gradientes: el de temperatura y el de la concentración del CO_2, dando lugar a una convección natural o convección termohalina.

Durante la fase de inyección en un proyecto de almacenamiento del CO_2, el CO_2 está presente en el yacimiento como un gas denso. La cantidad de CO_2 disuelto en el acuífero de la formación depende, entre otras variables, de las curvas de permeabilidad relativa de la formación y particularmente de la saturación residual del agua (Ennis-King & Paterson, 2005).

Después de la inyección el gas libre migra hacia arriba porque es menos denso que la salmuera, eventualmente el CO_2 se acumula entre la capa sello y la superficie del acuífero. El CO_2 queda capturado por dos mecanismos diferentes: atrapamiento capilar y atrapamiento de solubilidad. El atrapamiento capilar sucede porque parte del CO_2 sube a través de formaciones rocosas porosas debido a la flotación y las fuerzas capilares quedando atrapado en los poros de la roca. Por otro lado, la disolución del CO_2 en la interfaz entre la fase rica en CO_2 y la salmuera comienza con la difusión molecular. En este proceso aumenta la densidad de la salmuera aproximadamente 1% en la superficie superior del acuífero. Cuando esta capa de salmuera saturada se vuelve suficientemente gruesa, se produce una inestabilidad y debido al gradiente del soluto las plumas de salmuera saturada migran hacia abajo, diluyéndose a medida que avanzan. A este fenómeno se denomina atrapamiento de solubilidad (Islam et al., 2013).

Otro agente desestabilizador es el gradiente geotérmico de temperatura que induce la convección natural ascendente de la salmuera. La interacción de estos dos procesos opuestos es denominado transporte de doble difusión y determina la tasa

de concentración resultante del CO_2 en la salmuera. La importancia de esta mezcla convectiva es que, por lo general, es mucho más rápida que la difusión pura al mezclar el gas disuelto con agua. Por lo tanto, acelera la disolución general del CO al remover continuamente la salmuera rica en CO_2 de la capa superior y pone en contacto la salmuera subsaturada con la pluma de CO_2 que avanza hacia abajo (Islam et al., 2013). Ennis-King y Paterson (2005) encontraron que el fenómeno de doble difusión en el sistema se debe principalmente a un efecto de densidad, ya que el CO es de los pocos gases que, en condiciones de yacimiento, al ser disuelto en agua aumenta la densidad de la solución.

La importancia de la convección de doble difusión generalmente se reconoce a partir de la dinámica y los patrones de flujo asociados que se generan; en particular los fenómenos de los 'dedos convectivos' (Fig. 2) y la convección en capas, por lo que este tipo de convección presenta un reto en la simulación convencional ya que se requiere un modelado de los dedos convectivos a escala fina sobre una ruta de migración que puede tener decenas de kilómetros de largo (Ennis-King & Paterson, 2005).

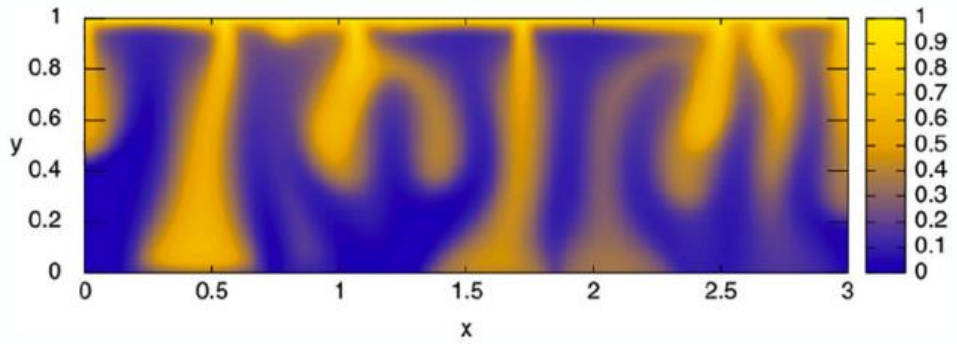

Figura 2. *Representación esquemática del fenómeno de los 'dedos convectivos' en un campo de concentración. Imagen tomada de Guerrero et al., 2020*

2 MARCO TEÓRICO

Para tener un mejor entendimiento del problema físico de la disolución del CO_2 en el subsuelo, investigadores del Instituto de Geofísica de la UNAM elaboraron un modelo matemático-computacional para el estudio del transporte de calor y de masa en un medio poroso que está saturado de fluido (Guerrero et al., 2020). En este trabajo se desarrolló un simulador numérico para evaluar los efectos de la inclinación, el número de Rayleigh y la relación de flotación del transporte convectivo en un medio poroso rectangular.

Simbología

Letras romanas		*Símbolos griegos*	
B	Altura del recinto poroso	α	Difusividad térmica
C	Ancho del recinto poroso	β	Coeficiente de expansión térmica
D	Difusividad de masa	ε	Porosidad normalizada (φ/σ)
g	Constante gravitacional	ν	Viscosidad cinemática
k	Permeabilidad	φ	Porosidad
Le	Número de Lewis	ψ	Función de corriente
N	Relación de flotación	σ	Relación de capacidades caloríficas sólido-líquido
Ra	Número de Rayleigh	θ	Ángulo de inclinación
S	Concentración de soluto		
T	Temperatura		
t	Tiempo		
$\boldsymbol{u}$	Velocidad de Darcy		
x, y	Coordenadas cartesianas		

2.1 Generalidades del problema

El problema considera un medio poroso rectangular de paredes impermeables con altura B y longitud C, y que está saturado de un fluido incompresible. El medio es calentado por debajo a temperatura constante y saturado de arriba con una concentración constante. Inicialmente el medio se encuentra en ausencia de soluto disuelto (S = 0), sus límites laterales son adiabáticos y tiene una inclinación θ con respecto a la horizontal (Fig. 3). Se supone un medio homogéneo e isotrópico que

está en equilibrio térmico local, el flujo del fluido es descrito por la Ley de Darcy y se aplica la aproximación de Boussinesq para modelar la convección termohalina.

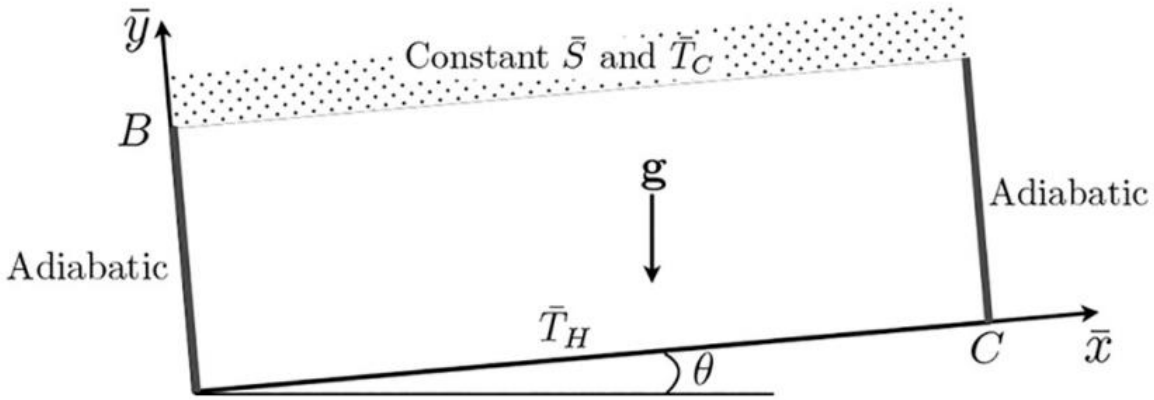

Figura 3. *Modelo esquemático del problema estudiado (Guerrero et al., 2020).*

2.2 Ecuaciones gobernantes

Ley de Darcy

El ingeniero francés Henry Darcy hizo contribuciones importantes en la disciplina de la hidráulica, una de sus principales contribuciones fue la determinación de una ley que rige el flujo del agua a través de arena. Evaluando el caudal del agua que fluye a través de una columna vertical rellena de arena, estableció que el caudal es directamente proporcional a la altura e inversamente proporcional al espesor de la capa atravesada, estableciendo la siguiente ecuación:

$$Q = KA\frac{h_1 - h_2}{L} \tag{2.1}$$

Donde Q es la tasa de flujo volumétrico del agua, A es el área de la sección transversal y L la altura de la columna de arena, K es una constante de proporcionalidad llamada conductividad hidráulica ($K = \frac{k\rho g}{\mu}$). h_1 y h_2 es la altura piezométrica superior e inferior de la columna, respectivamente. La ecuación 2.1 se puede escribir en términos del gradiente hidráulico definido como $J = \frac{h_1 - h_2}{L}$ y la tasa de flujo volumétrico a través de un área de sección transversal unitaria, q, conocida como velocidad de Darcy.

$$q = KJ \tag{2.2}$$

$h_1 - h_2$ representa una pérdida de carga, que es la pérdida de energía por unidad de peso. Esta pérdida de energía se debe a la resistencia de la viscosidad del fluido a medida que se desplaza por los caminos tortuosos del medio poroso. Una implicación de la ley de Darcy es que los cambios de la energía cinética en el fluido a medida que fluye a través de la columna de arena son insignificantes. Kolditz (2001) señaló que el límite máximo para que la ley de Darcy sea válida es mucho antes de la transición de flujo laminar a turbulento, dicha condición se cumple en la mayoría de los casos en el flujo de agua subterránea.

Darcy encontró que el flujo toma lugar en dirección de la disminución de la altura, en lugar de la dirección de disminución de la presión. Este potencial es descrito matemáticamente en términos del gradiente de h, así que la tasa del flujo volumétrico por unidad de área esta dado por:

$$q = -K\nabla h \tag{2.3}$$

La velocidad promedio del flujo a través de la columna es obtenida del área de la sección transversal A multiplicada por la porosidad φ. Para un flujo de tres dimensiones se tiene que:

$$V = \frac{q}{\varphi} \tag{2.4}$$

Medio homogéneo e isotrópico

Un medio homogéneo supone que sus propiedades físicas son invariables en el espacio (para *x, y* y *z*). La isotropía, significa que tiene la misma capacidad para el transporte de energía todas las direcciones. En este problema la energía que se transporta en el medio es calor, masa y momento, y suponemos que el medio es isotrópico para todas estas tres formas de energía.

Ecuación de momento

Considerando las definiciones de conductividad hidráulica y altura piezométrica, los componentes de la Ecuación 2.3 para un medio poroso homogéneo, la ecuación de momento se puede escribirse:

$$\bar{u} = -\frac{k}{\mu}\frac{\partial \bar{P}}{\partial \bar{x}} \tag{2.5}$$

$$\bar{v} = -\frac{k}{\mu}\frac{\partial \bar{P}}{\partial \bar{y}} \tag{2.6}$$

$$\bar{w} = -\frac{k}{\mu}\left(\frac{\partial \bar{P}}{\partial \bar{z}} + \rho g\right) \tag{2.7}$$

Las fuerzas que impulsan el movimiento se clasifican como fuerzas externas y de cuerpo. Las fuerzas externas son aquellas asociadas con gradientes de presión, mientras que las fuerzas de cuerpo están asociadas con efectos gravitatorios, que solo son importantes para la componente $\bar{z}$ del sistema.

Ecuación de transferencia de calor y masa

El transporte de energía a través de un medio poroso saturado con fluido se obtiene bajo el supuesto de que existe un equilibrio térmico local entre la matriz porosa y el fluido y que la disipación viscosa es insignificante, por lo que se requiere que no haya cambios drásticos de temperatura en el sistema. Estas condiciones generalmente se cumplen en la convección libre en medios porosos en el sistema. Además, se supone que la densidad y el calor específico no varían con el tiempo ni con la posición, y la conductividad térmica se considerará en general una función de las coordenadas espaciales. También se supone que no hay fuentes de calor en el medio poroso. Bajo estos supuestos, el balance de energía se compone de transferencia de calor por conducción y convección en el medio y toma la forma ((Nield & Bejan, 2006); (Guerrero Martínez, 2017)):

$$\sigma\frac{\partial \bar{T}}{\partial \bar{t}} + \bar{u}\cdot\bar{\nabla}\bar{T} = \bar{\nabla}\cdot(\alpha\bar{\nabla}\bar{T}) \tag{2.8}$$

donde σ es el factor de escala y α representa la difusión térmica. El primer término se refiere a la energía acumulada en el tiempo, el segundo término es el transporte convectivo de calor y el tercer término es el transporte difusivo de calor.

De manera análoga, la ecuación que representa el transporte de un soluto de un fluido dentro de un medio poroso es la siguiente:

$$\varphi\frac{\partial \bar{S}}{\partial \bar{t}} + \bar{u}\cdot\bar{\nabla}\bar{S} = \bar{\nabla}\cdot(D\bar{\nabla}\bar{S}) \tag{2.9}$$

Conservación de masa

Un balance de masa en un volumen de control en un medio poroso saturado de fluido es el resultado de cuantificar las entradas y salidas de fluido en el volumen de control más las variaciones de densidad con el tiempo. El resultado es la ecuación de continuidad:

$$\frac{\partial \rho}{\partial t} + \bar{\nabla} \cdot (\rho \bar{u}) = 0 \tag{2.10}$$

Manteniendo la suposición de densidad constante en cuanto a la posición y el tiempo, el balance de masa toma la forma de la ecuación de continuidad para un fluido incompresible:

$$\bar{\nabla} \cdot \bar{u} = 0 \tag{2.11}$$

Aproximación de Boussinesq

La aproximación de Boussinesq supone que variaciones de temperatura son lo suficientemente pequeñas como para considerar la densidad ρ como una constante en el sistema, excepto en el término de flotación ρg, donde g es la fuerza gravitatoria. La densidad ρ se puede aproximar de la siguiente manera: $\rho = \rho_0[1 - \beta(T - T_0)]$, donde T es la temperatura, ρ_0 y T_0 son la densidad y temperatura de referencia, respectivamente. Las propiedades del fluido como la viscosidad dinámica μ, la permeabilidad k, la expansión térmica β, la difusividad térmica α, y el calor específico c_P se suponen constantes. Para grandes gradientes de densidad la aproximación de Boussinesq no puede ser aplicada. Escribiendo la fuerza de cuerpo en la ecuación 2.7 en términos de la densidad, la ecuación de momento resulta:

$$\bar{u} = -\frac{k}{\mu}\frac{\partial \bar{P}}{\partial \bar{x}} \tag{2.12}$$

$$\bar{v} = -\frac{k}{\mu}\frac{\partial \bar{P}}{\partial \bar{y}} \tag{2.13}$$

$$\bar{w} = -\frac{k}{\mu}\left(\frac{\partial \bar{P}}{\partial \bar{z}} + \rho g(1 - \beta_0(T - T_0)\right) \tag{2.14}$$

La ecuación se puede simplificar redefiniendo el gradiente de presión vertical para que se tome en relación con un gradiente de presión hidrostática de referencia ρ_0g. Esto conduce a la versión más común de la ecuación de cantidad de movimiento

basada en la ley de Darcy y la aproximación de Boussinesq. En forma vectorial esta ecuación es (Guerrero Martínez, 2017):

$$\bar{u} = -\frac{k}{\mu}(\bar{\nabla}\bar{P} - \rho_0\beta_0(T - T_0)g) \tag{2.15}$$

2.3 Formulación del problema adimensional

Para la formulación del problema adimensional, primero se identifican los parámetros y operadores adimensionales (Guerrero et al., 2020):

$$Ra = \frac{gBk\beta_T\Delta\bar{T}}{\alpha\nu}; \qquad Le = \frac{\alpha}{D}; \qquad N = \frac{\beta_s\Delta\bar{S}}{\beta_T\Delta\bar{T}}$$

Ra es el número de Rayleigh. Este número indica una medida de la capacidad de flotación que tiene el fluido (gravedad, permeabilidad, coeficiente de expansión volumétrica y diferencia de temperatura característica), sobre la medida de la capacidad de transporte de calor y momento por difusión (difusividad térmica y viscosidad cinemática).

Le es el número de Lewis. Este número mide la relación entre el transporte difusivo de calor y el transporte difusivo de masa.

N es la relación de flotación que compara la capacidad de flotación por efectos de disolución del soluto con respecto a la flotación debida gradientes de temperatura.

Después se definen los números que representan una medida del transporte convectivo en el sistema: el número de Nusselt, el número de Sherwood y la concentración media del soluto que son los parámetros físicos para el análisis de los resultados numéricos:

$$Nu = \int \left|\frac{\partial T}{\partial \bar{y}}\right| dx; \qquad Sh = \int \left|\frac{\partial T}{\partial \bar{y}}\right| dx; \qquad S_a = \frac{\int S dA}{A}$$

Introduciendo las consideraciones expuestas en la sección anterior y los parámetros gobernantes, la forma final y adimensional de las ecuaciones de transporte de calor, masa y momento, respectivamente, se expresan como:

$$\frac{\partial T}{\partial t} - \nabla^2 T + u \cdot \nabla T = 0 \tag{2.16}$$

$$\varepsilon \frac{\partial S}{\partial t} - \frac{1}{Le} \nabla^2 S + u \cdot \nabla S = 0 \tag{2.17}$$

$$u + \nabla P = Ra(T + NS)e \tag{2.18}$$

además, se supuso una constante $\varepsilon = 1$ en el modelo matemático. La ecuación de cantidad de momento (Ec. 2.18) se escribe en términos de la función de corriente, ψ: $u = \partial\psi/\partial y$, $v = -\partial\psi/\partial y$, lo que lleva a una ecuación de Poisson de la forma (Guerrero, et al. 2020):

$$\Gamma \nabla^2 \psi = \left(\frac{\partial T}{\partial x} + N \frac{\partial T}{\partial x}\right) cos\,\theta \; - \; \left(\frac{\partial T}{\partial y} + N \frac{\partial T}{\partial y}\right) sin\,\theta \tag{2.19}$$

Donde $\Gamma = -1/Ra$. Las Ecuaciones (2.16), (2.17) y (2.19) son las que representan el problema a resolver. Se tienen que establecer las condiciones de frontera para cada ecuación; Para la ecuación de transferencia de calor, el campo de temperatura satisface:

$\frac{\partial T}{\partial x} = 0$, para $x = 0$ y $x = 3$

$T = 1$, para $y = 0$ y $t > 0$

$T = 0$, para $y = 1$ y $t > 0$

La ecuación de transferencia de masa está sujeta a:

$\frac{\partial S}{\partial x} = 0$, para $x = 0$ y $x = 3$

$\frac{\partial S}{\partial y} = 0$, $para$ $y = 0$

$S = 1$, $para$ $y = 1$ y $t > 0$

Para la ecuación de momento se establece $\psi = 0$ en los límites para lograr que no fluya fluido a través de las paredes:

$\psi = 0$, $para$ $x = 0$ y $x = C$

$\psi = 0$, $para$ $y = 0$ y $y = 1$

2.4 Solución y validación del modelo

La solución del modelo matemático consta de tres etapas. En la primera etapa los autores implementaron un algoritmo iterativo de punto fijo, ya que el sistema de tres ecuaciones diferenciales necesita satisfacerse de manera simultánea para cada paso de tiempo considerado (Báez y Nicolás, 2006), y utilizaron un criterio de convergencia para estos algoritmos iterativos de $1x10^{-6}$. La segunda etapa se asocia a la solución de las ecuaciones diferenciales individualmente, donde se emplea el método de volumen finito, el cual se aplica en una malla uniforme de 100x100 elementos. Y finalmente, la tercera etapa soluciona los sistemas lineales asociados a cada ecuación diferencial, los solucionadores pueden ser Gauss-Seidel, SOR y Gradiente Conjugado.

El modelo numérico reportado por Guerrero et al. (2020) fue implementado en Fortran 90 con bibliotecas de cómputo paralelo en memoria compartida (OpenMP de Intel®). La malla seleccionada fue de $\Delta x = \Delta y = 1/250$, ya que esta resolución permite capturar las características del flujo a pequeña escala.

Este modelo fue validado comparando los resultados reportados por Islam et al. (2013). Se compararon los resultados de $S_a(t)$ con diferentes números de Rayleigh ($Ra = 1, 10\ y\ 100$) (Fig. 4). Los resultados son consistentes a pesar de algunas diferencias para *Ra* = 100. Estas diferencias pueden ser atribuidas a la resolución de los sistemas de ecuaciones diferenciales y algebraico de cada autor.

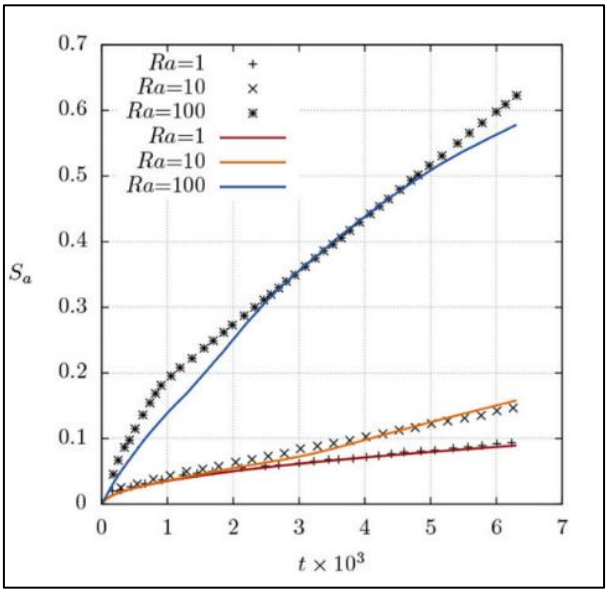

Figura 4. *Comparación entre los resultados obtenidos por Guerrero et al. (2020) y la referencia Islam et al. (2013). (Guerrero et al., 2020).*

2.5 Resultados y retos al programar en Fortran.

Los resultados de Guerrero et al. (2020) ayudan a mejorar la comprensión del comportamiento de la doble convección difusiva en medios porosos en el contexto de disolución de soluto. Sus principales conclusiones fueron:

- La relación de flotación es el parámetro más efectivo para mejorar la mezcla.
- Con respecto al número de Rayleigh, las curvas muestran una mezcla más rápida para valores de Ra más grandes.
- Las fuerzas de flotación para los gradientes de temperatura y saturación incrementan con Ra, y esto conduce a un sistema altamente convectivo.
- El efecto de variar θ muestra una un impacto moderado de la saturación dada N y Ra, es decir que, la mayor inclinación (15°) no implica un sistema notoriamente más convectivo en comparación con un estrato poroso horizontal.

Para llegar a estas conclusiones, los autores tuvieron que ejecutar distintas simulaciones y realizar varias pruebas. Estas pruebas consistieron en el análisis de los números de Nusselt y Sherwood, utilizando distintos valores del número de Rayleigh, cambiando el ángulo de inclinación para una relación de flotación dada, además, el análisis de la simulación de los gradientes de temperatura y saturación a lo largo del tiempo.

El modelo numérico fue implementado en lenguaje Fortran 90, un lenguaje compilado de alto nivel con un amplio soporte de bibliotecas utilizadas en ingeniería y ciencia. Sin embargo, una de sus principales desventajas es que, al ser compilado, limita la interactividad con el usuario, y esto dificulta tanto la definición de casos de simulación, así como el análisis de resultados. Por consiguiente, se encuentra un área de oportunidad para desarrollar una herramienta que facilite el uso de este simulador y el análisis de sus resultados.

3 PROGRAMA EN PYTHON PARA EL USO DEL SIMULADOR GEOTHERMOHALINE

En el presente trabajo se desarrolló un código escrito en Python para definir las condiciones de la simulación y visualizar los resultados generados por el simulador numérico de convección termohalina publicado por Guerrero et al. (2020), al que se le denominará 'GEOThermohaline'. Se utilizó Python ya que es un lenguaje de programación de alto nivel (Downey, 2012) que ofrece una gran cantidad de módulos científicos para el procesamiento y visualización de datos. La Tabla 1 presenta los principales módulos que se utilizaron para el desarrollo del presente programa. Adicionalmente, existen diferentes interfaces mediante las cuales se puede ejecutar el lenguaje Python. Para este trabajo se utilizó la interfaz 'Jupyter Notebook' (https://jupyter.org), el cual permite al usuario escribir, documentar y ejecutar código, además de la manipulación de datos y la generación de salidas gráficas de diversos tipos; por lo tanto, resulta una buena opción para los fines del trabajo.

Tabla 1. *Módulos de Python utilizados en el programa.*

Módulo	Descripción
Numpy	Biblioteca que da soporte para crear vectores y matrices multidimensionales, colección de funciones matemáticas de alto nivel para operar con ellas.
Matplotlib	Biblioteca para la generación de gráficos en dos dimensiones
Display	Modulo para mostrar diferentes representaciones de objetos incluyendo: HTML, JSON, PGN, JPEG, etc.
Ipywidgets	Son *widgets* HTML interactivos para Jupyter Notebooks. Al usarlos los usuarios obtienen el control de sus datos y pueden visualizar los cambios en los datos.
Os	Proporciona una forma portátil de realizar tareas dependientes de un sistema operativo.

Para la realización de este cuaderno interactivo se planteó que el primer problema a resolver sería codificar una función que leyera los datos de salida del simulador y los acomodara en una matriz cuyo tamaño depende de los pasos de tiempo y los datos generados en el eje *x* y en el eje *y* de ambos campos (concentración y temperatura). Posteriormente, se ajustaron parámetros de visualización de los resultados para usar función de animación que permite visualizar la simulación a lo largo de los pasos de tiempo seleccionado.

Teniendo estas dos funciones resueltas, se creó en constructor de la clase para iniciar el ejercicio seleccionando el tipo de solucionador numérico, los parámetros de entrada, la escritura de los archivos de entrada y la ejecución del simulador.

A continuación, se presenta el diagrama de flujo del funcionamiento del código (Fig. 5). Adicionalmente, se enlistan las principales funciones del programa en Python desarrollado en el presente trabajo.

- Permitir que el usuario seleccione el tipo de solucionador que resolverá los sistemas de ecuaciones lineales (Gauss-Seidel, SOR o Gradiente Conjugado).
- Permitir que el usuario defina los parámetros de entrada para el solucionador (número de Rayleigh, relación de flotación, ángulo de inclinación y pasos de tiempo).
- Ejecutar el simulador.
- Graficar *Nu(t), Sh(t), Sa(t)* y el desempeño del solucionador para cada ecuación diferencial (número de iteraciones por cada paso de tiempo).
- Visualizar la simulación de un medio poroso que está siendo saturado por su tapa superior y calentado a temperatura constante en la parte inferior.

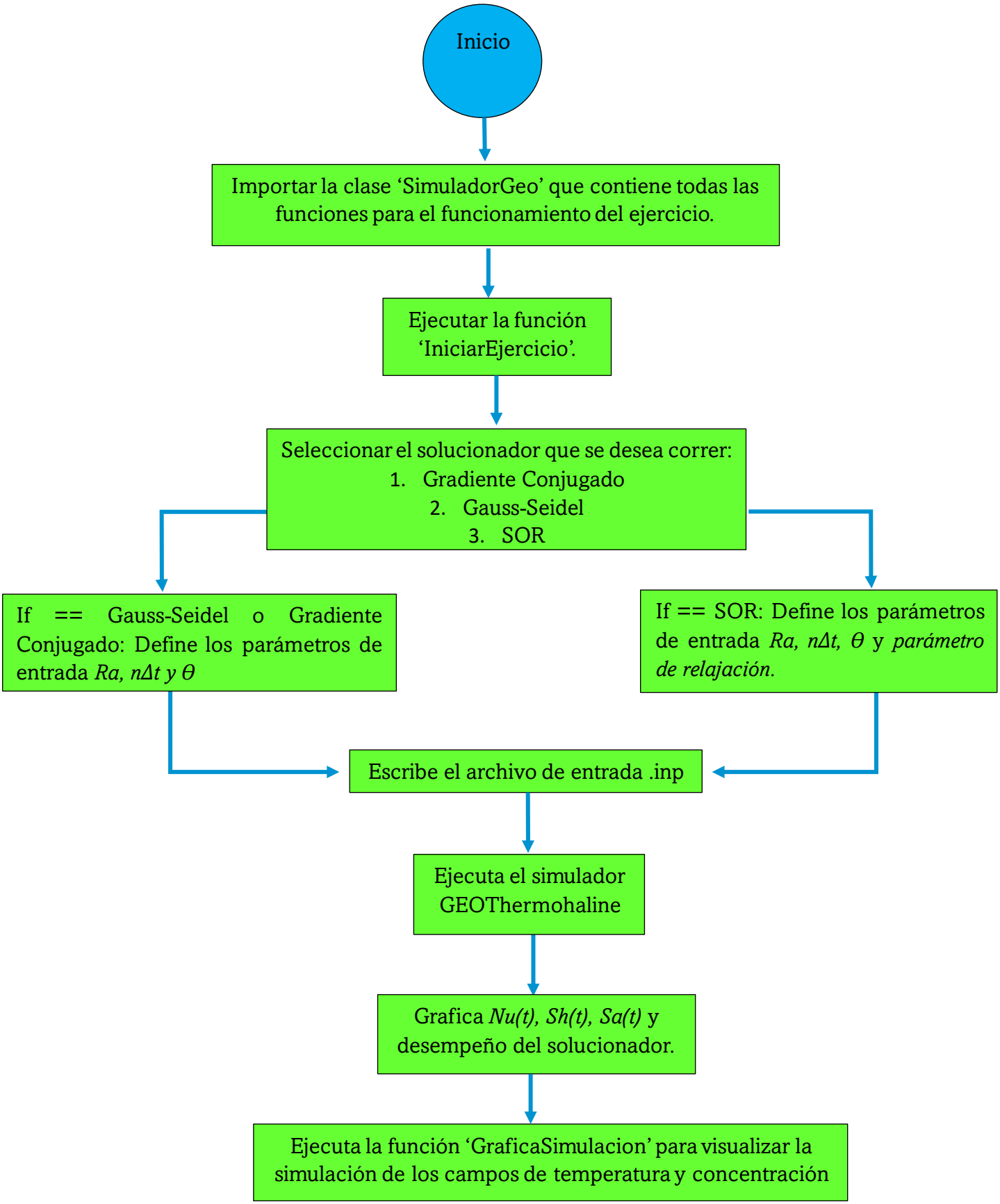

Figura 6. *Diagrama de flujo que indica el funcionamiento de la interfaz creada.*

3.1 Detalle del código

El programa esquematizado en la Figura 4 consta de una clase y un conjunto de métodos asociados a ella, estos se describen en la Tabla 2 (se puede consultar el código completo en el Anexo I).

Tipo	Nombre	Funcionamiento
Clase	*SimuladorGeo*	Contiene todos métodos para el funcionamiento del ejercicio
Método	*IniciarEjercicio(self)*	Seleccionar el tipo de solucionador: 1 para gradiente conjugado, 2 para Gauss-Seidel o 3 para SOR.
	interfazGrafica(self)	Interfaz gráfica para definir los parámetros de entrada
	generaInput(self)	Escribe el archivo de entrada 'entrada. inp' a partir de los parámetros definidos por el usuario.
	ejecutaSimulador(self)	Ejecuta el simulador GEOThermohaline para los sistemas operativos macOS y Linux.
	graficaSolucion(self)	Lee el archivo de salida de GEOThermohaline y grafica el número de pasos de tiempo vs Numero de Nusselt, Sherwood y Saturación, y el número de iteraciones para las ecuaciones de velocidad, masa y calor.
	graficaSimulacion(self)	Lee los archivos de salida para los campos de temperatura y concentración de GEOThermohaline para generar la visualización de la simulación.

Tabla 2. *Nombre y funcionamiento de la clase y los métodos que fueron desarrollados en el código.*

4 RESULTADOS

En este capítulo se describe el funcionamiento de la interfaz gráfica que, en síntesis, consiste en tres pasos: la carga del código que contiene la clase simuladorGeo y sus métodos; la definición de parámetros para el simulador GEOThermohaline y ejecución; visualización de resultados. En la siguiente sección Se muestran ejemplos de su funcionalidad.

4.1 Ejecución del código

El código es un archivo con extensión '.py' que puede ser ejecutado desde un cuaderno interactivo (Jupyter Notebook). A continuación, se muestra un ejemplo de uso del programa para realizar una simulación en GEOThermohaline. La ejecución de las celdas es con el botón '▶Run' (que se encuentra en la parte superior del cuaderno) (Fig. 6).

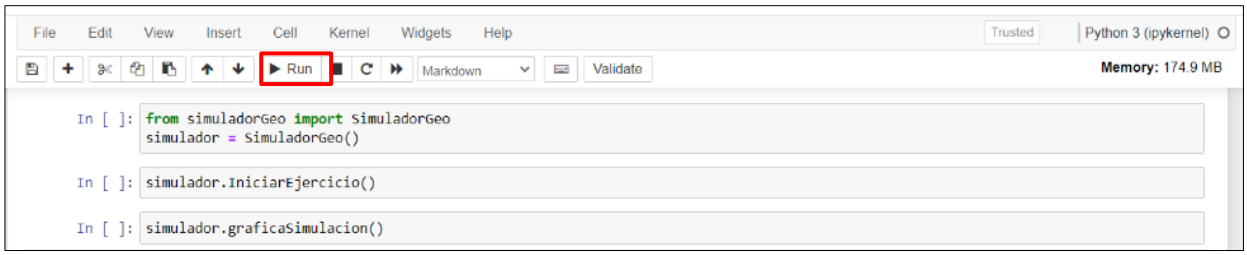

Figura 7. *Las celdas del cuaderno interactivo tienen que ser ejecutadas con el botón 'Run' (recuadro rojo).*

En la primera celda se importa la clase 'simuladorGeo' que contiene todos los métodos para el funcionamiento del programa, y se define la variable 'simulador' del cuaderno.

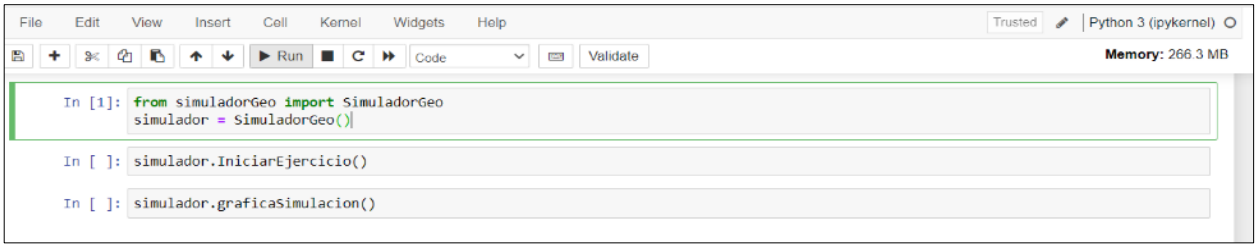

Figura 10. *Al ejecutar la primera celda (señalada con verde) se importa la clase 'simuladorGeo'.*

Para iniciar el ejercicio se ejecuta la siguiente celda que llama al método 'IniciarEjercicio()'. Al ejecutarse se despliegan las opciones para seleccionar tipo de solucionador (Fig. 8).

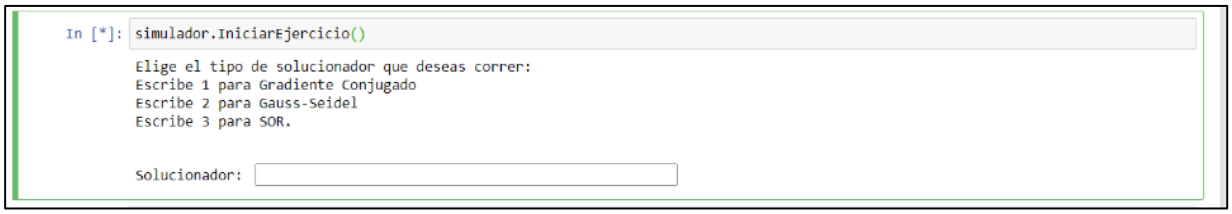

***Figura 13**. Se ejecuta la segunda celda para iniciar con el ejercicio. El usuario tendrá que elegir el tipo de solucionador que desea correr.*

Cuando se elige el solucionador, se despliegan barras horizontales deslizantes para definir los parámetros de entrada, sin embargo, también se pueden definir seleccionando y escribiendo los valores numéricos manualmente (Fig. 9).

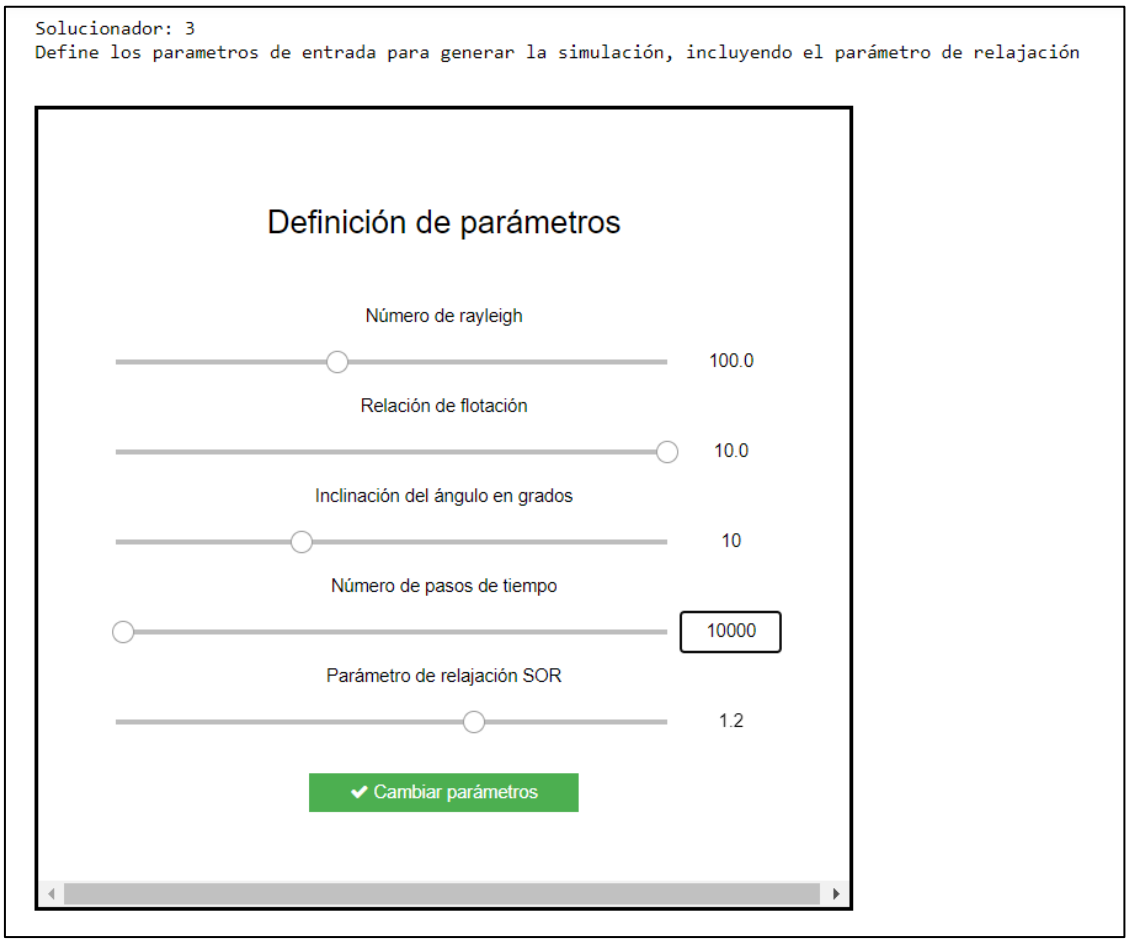

***Figura 16**. Definición de los parámetros de entrada. Para escribir el archivo de entrada con los parámetros definidos por el usuario se presiona el botón que indica 'Cambiar parámetros'.*

Al presionar el botón de 'Cambiar parámetros' se escribe el archivo de entrada para el simulador. En seguida aparece una leyenda que indica que el archivo de entrada ha sido escrito correctamente y enlista los valores definidos por el usuario (Fig. 10).

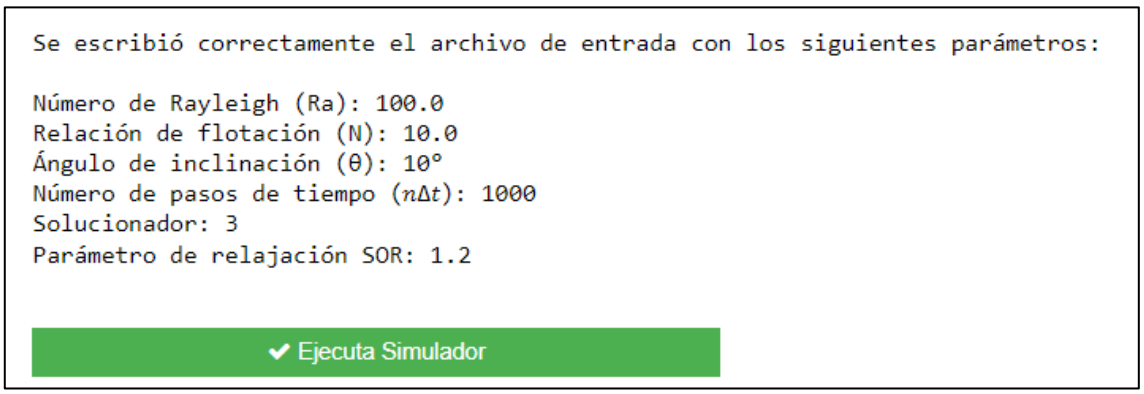

Figura 19. *Leyenda que indica que el archivo de entrada se ha escrito correctamente. Para ejecutar el simulador se presiona el botón verde.*

El botón 'Ejecuta simulación' ejecuta el simulador GEOThermohaline. Cuando se selecciona este botón saldrá una leyenda que la simulación ha comenzado y al terminar indicará el tiempo que tomó la simulación (Fig. 11). La ejecución del simulador puede tomar desde algunos segundos hasta varios minutos, esto dependerá de los valores de los parámetros de entrada y el tipo de solucionador que el usuario haya elegido.

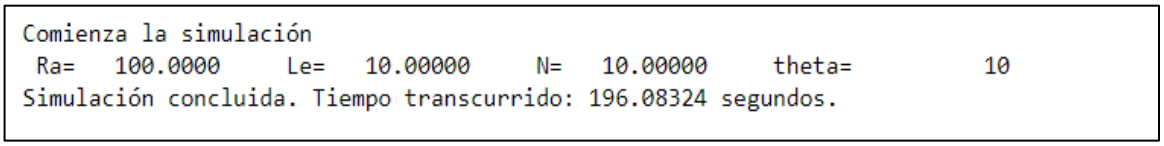

Figura 20. *Leyenda que indica el estado de la simulación y cuánto tiempo tomó.*

Al concluir la simulación se puede graficar los números de Nusselt, Sherwood y la saturación en función del tiempo, que son la medida del transporte convectivo en el sistema (Fig. 12a). También se puede graficar desempeño del solucionador en cada ecuación gobernante (transporte de calor, masa y velocidad), es decir que se indica el número de iteraciones que necesitó cada ecuación (Fig. 12b).

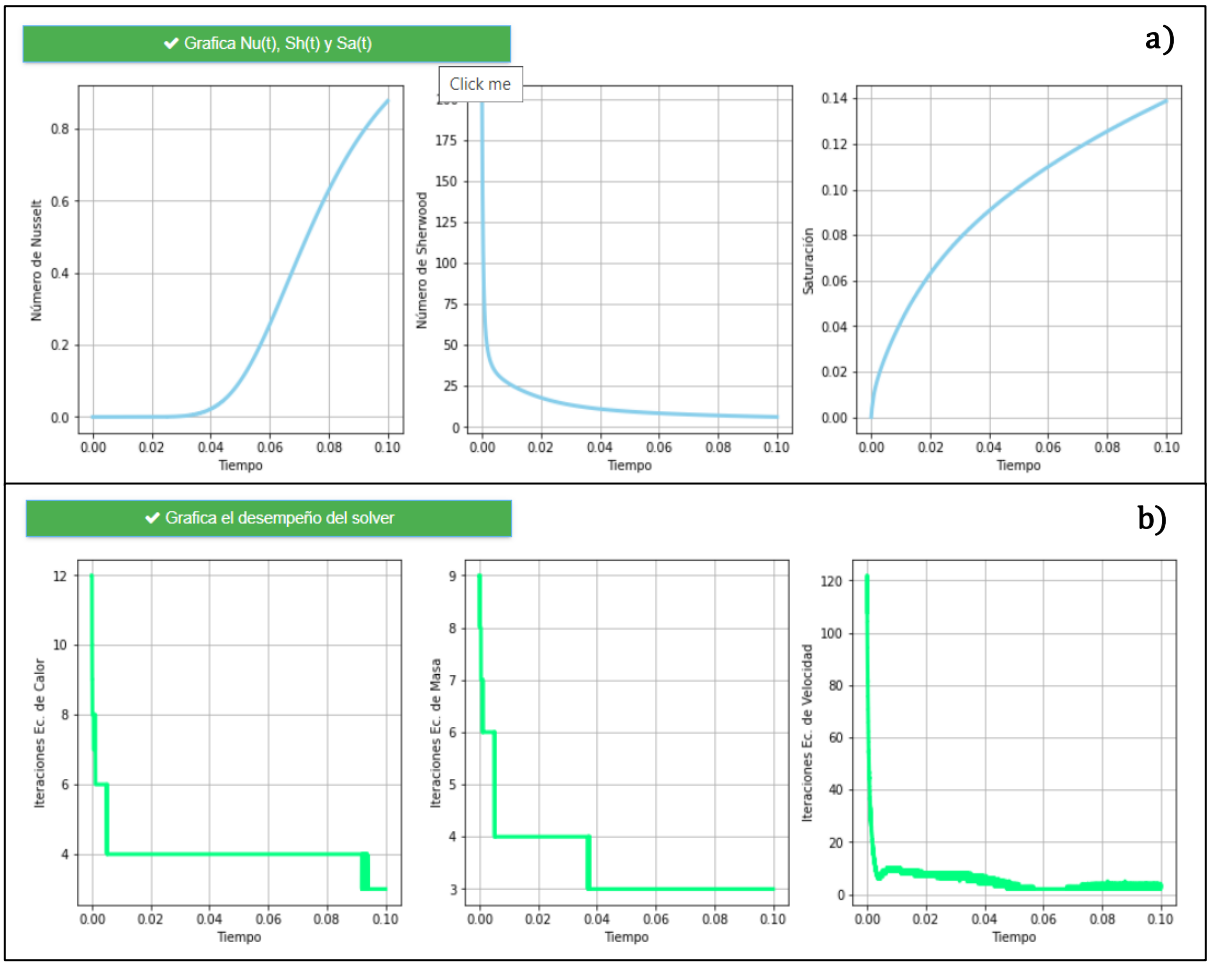

Figura 22. *Graficas para el análisis del ejercicio.* ***a)*** *Muestra Nu(t), Sh(t) y Sa(t), que son medidas del transporte convectivo en el sistema en función del tiempo.* ***b)*** *Muestra el número de iteraciones para cada ecuación gobernante.*

Finalmente, para visualizar la simulación de los campos de concentración y temperatura se ejecuta la tercera celda. Esta celda llama al método 'graficaSimulacion()', que genera un video animado de la simulación (Fig. 13). La visualización del video puede tardar algunos minutos debido a la gran cantidad de datos que lee la función de animación.

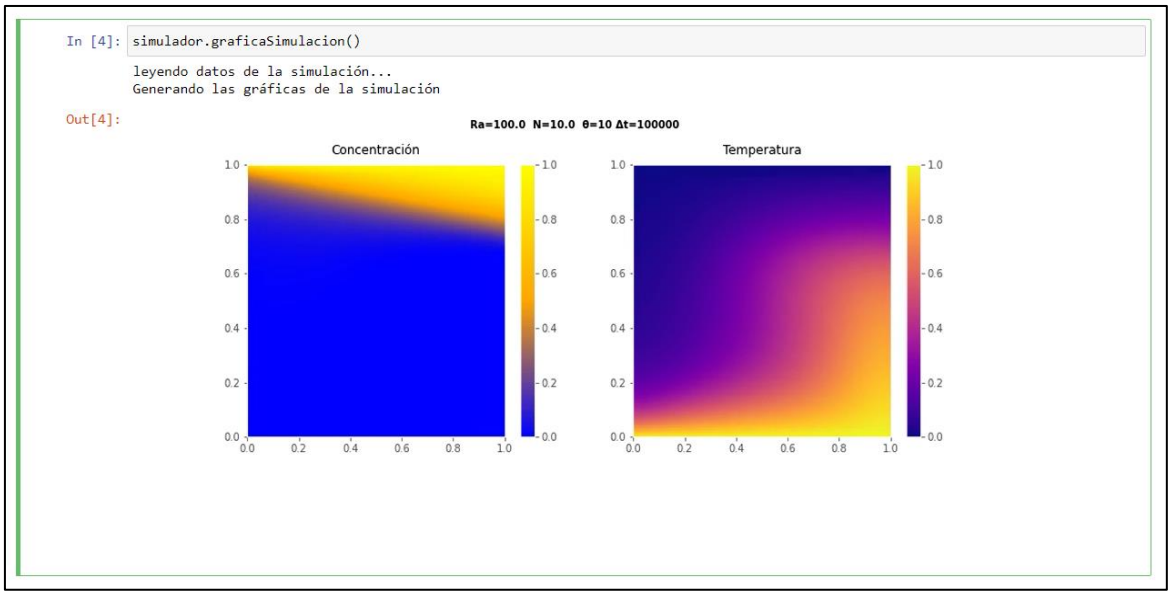

Figura 25. *La ejecución de la tercera celda muestra la simulación de los campos de concentración y temperatura en un video animado.*

4.2 Discusión de la importancia y aportaciones de la interfaz gráfica

El resultado de este código es un cuaderno interactivo que permite definir los parámetros de entrada para ejecutar el simulador GEOThermohaline y visualizar sus resultados. Es decir que con esta herramienta es posible probar valores distintos de los parámetros de entrada y cambiar el tipo de solucionador para analizar sus resultados, porque facilita el uso del simulador y la visualización de sus resultados.

La principal intensión de experimentar con distintos valores es para tener una mejor comprensión de la física del fenómeno de convección termohalina en un medio poroso hidrotermal. Por ejemplo, si en dos casos se establecen los siguientes parámetros: N= -10, θ= 0 *y* $n\Delta t$ = 100000, pero para el primer caso el valor de Ra= 50 y, para el segundo es Ra= 250, se observará un comportamiento muy distinto, pues para el caso de Ra= 250 el fenómeno de los dedos convectivos en ambos gradientes es notoriamente visible (Fig. 14b)., por el contrario de Ra= 50, donde la convección se observa más estable (Fig. 14a). Por lo tanto, se puede decir que valores altos de Ra indicarían un medio altamente convectivo.

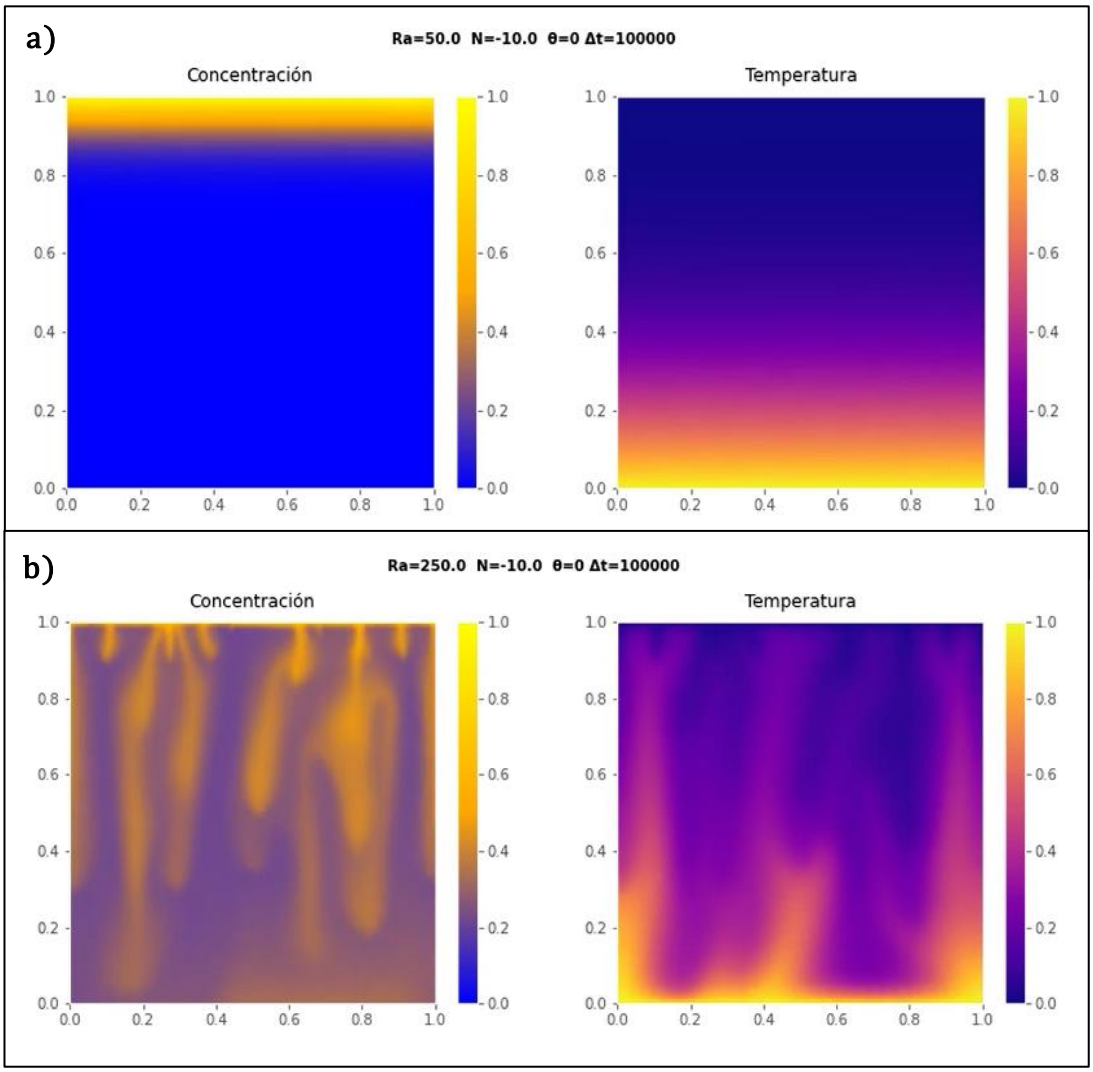

Figura 28*. Resultados de la simulación para dos casos similares, pero con diferente número de Rayleigh. a) Ra = 50 y b) Ra=250.*

Esta herramienta también permite hacer análisis del desempeño de los solucionadores. Por ejemplo, se establecieron los mismos parámetros de entrada para cada solucionador (*Ra= 25, N= 10, θ= 0 y nΔt = 25000*), y se encontró que el solucionador SOR es el que requiere más iteraciones en cada ecuación, sin embargo, esto no implicó un mayor tiempo de cómputo para realizar la simulación (Fig. 17). Por el contrario, el solucionador de Gradiente Conjugado es el que requiere el mayor tiempo de cómputo (Fig. 15). Además, se observa que la ecuación de calor es la que demanda mayor número de iteraciones para cada solucionador.

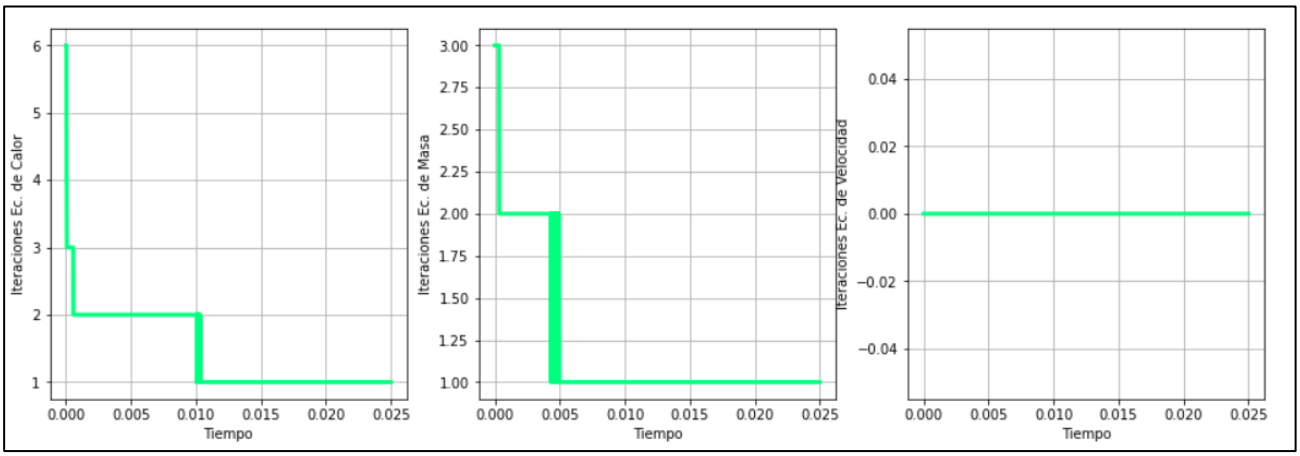

Figura 31. *Desempeño del solucionador Gradiente Conjugado (tiempo transcurrido: 93.46 s).*

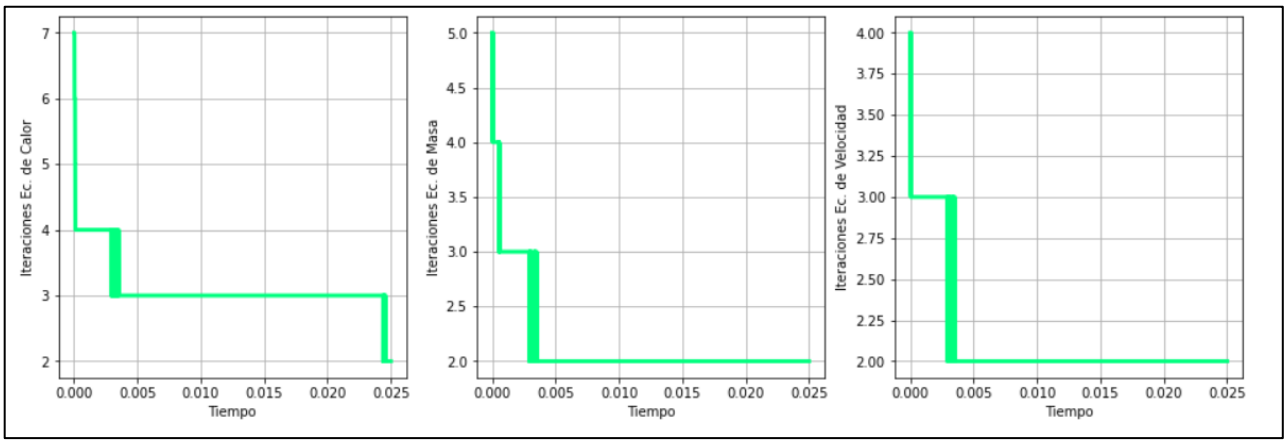

Figura 34. *Desempeño del solucionador Gauss-Seidel (tiempo transcurrido: 29.39 s).*

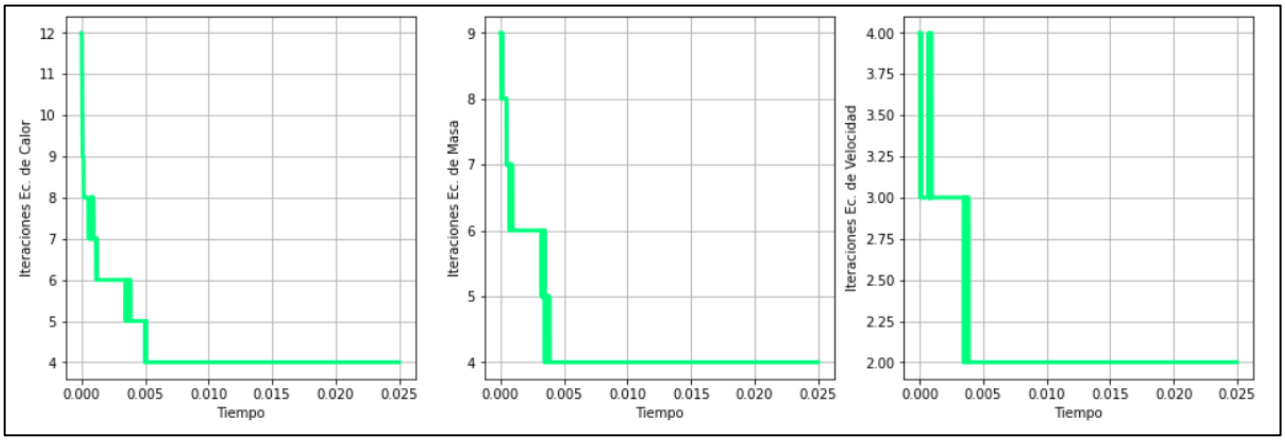

Figura 37. *Desempeño del solucionador SOR (tiempo transcurrido: 42.88 s).*

La interfaz grafica fue desarrollada en el marco del proyecto MACTI, que es un proyecto impulsado por profesores de la Universidad Nacional Autonoma de México (UNAM) para la enseñanza y el aprendizaje en temas de ecuaciones diferenciales y análisis numerico; el principal recurso son los cuadernos interactivos (Jupyter Notebooks) con ejemplos prácticos y aplicaciones de estos temas. Por lo tanto, el cuaderno interactivo desarrollado en este trabajo estará disponible en la plataforma de MACTI y en su repositorio; podrá ser usado libremente por profesores y estudiantes como herramienta complementaria en temas de simulaciones numéricos y modelamiento matemático (https://www.macti.unam.mx/).

CONCLUSIONES

Se realizó un programa en Python para dar interactividad y facilitar el uso de un simulador de convección termohalina escrito en Fortran 90 (GEOThermohaline, Guerrero et al., 2020). El programa en Python permitió flexibilizar la forma en la que se ejecuta el simulador y con ello, sacar el mayor provecho a sus capacidades. Adicionalmente, facilitó el procesamiento de resultados al tener en un mismo ambiente de trabajo todas las herramientas necesarias para la graficación de parámetros de salida y visualización, mediante gráficas y animaciones. A continuación, se presentan las principales conclusiones.

1. El modelado numérico tiene un papel fundamental en la ingeniería de yacimientos ya que ayuda a tener un mejor entendimiento de los fenómenos físicos que ocurren en el sistema cuando su estado natural está siendo alterado. Contar con herramientas eficientes de pre y post-procesamiento es clave para el mayor aprovechamiento de estas técnicas de análisis.
2. Un problema en el modelado numérico de yacimientos que está tomando relevancia en la actualidad es la inyección de CO_2 en formaciones geológicas (secuestro geológico del CO_2). El simulador numérico propuesto por Guerrero et. al (2020) permite una mejor comprensión de la convección termohalina en sistemas hidrotermales, que hace referencia a la inyección del CO_2 en estos sistemas. Tal simulador fue escrito en Fortran 90, por ser un lenguaje de altas capacidades en cómputo numérico, sin embargo, puede tener algunas desventajas para la ejecución y la visualización de resultados. El presente trabajo demostró que la utilización de forma complementaria de distintos lenguajes de programación es una alternativa para optimizar el análisis numérico.
3. El programa propuesto no solo favoreció el uso del simulador GEOThermohaline, sino que también es una oportunidad para crear cuadernos interactivos de enseñanza, en los que los estudiantes de ingeniería puedan tener un mayor acercamiento a los métodos numéricos y las ciencias computacionales con ejemplos prácticos.

REFERENCIAS

Baklid, A., & Korbol, R. (1996). Sleipner Vest CO2 Disposal, CO2 Injection Into a Shallow Underground Aquifer. *Society of Petroleum Engineers*, 269–277.

Bickle, M. J. (2009). Geological carbon storage. *Nature Geoscience*, *2*(12), 815–818. https://doi.org/10.1038/ngeo687

Brown, D. (2000). A Hot Dry Rock Geothermal Energy Concept Utilizing Supercritical Co2 Instead of Water. *Twenty-Fifth Workshop on Geothermal Resevoir Engineering*, *1995*(April 1992), 1–6.

Dicharry, R. M., Perryman, T. L., & Ronquille, J. D. (1973). Evaluation and Design of a Co2 Miscible Flood Project - Sacroc Unit, Kelly-Snyder Field. *JPT, Journal of Petroleum Technology*, *25*, 1309–1318. https://doi.org/10.2118/4083-PA

Downey, A. (2012). Think Python. How to Think Like a Computer Scientist. In *Journal of Chemical Information and Modeling*.

Ennis-King, J., & Paterson, L. (2005). Role of convective mixing in the long-term storage of carbon dioxide in deep saline formations. *SPE Journal*, *10*(3), 349–356. https://doi.org/10.2118/84344-PA

Freund, P. (2013). Anthropogenic climate change and the role of CO2 capture and storage (CCS). In *Geological Storage of Carbon Dioxide (co2): Geoscience, Technologies, Environmental Aspects and Legal Frameworks* (pp. 3–25). Woodhead Publishing Limited. https://doi.org/10.1533/9780857097279.1.3

Furre, A. K., Eiken, O., Alnes, H., Vevatne, J. N., & Kiær, A. F. (2017). 20 Years of Monitoring CO2-injection at Sleipner. *Energy Procedia*, *114*(November 2016), 3916–3926. https://doi.org/10.1016/j.egypro.2017.03.1523

Ghahfarokhi, R. B., Pennell, S., Matson, M., & Linroth, M. (2016). Overview of CO2 injection and WAG sensitivity in SACROC. *Proceedings - SPE Symposium on Improved Oil Recovery*, *2016-Janua*(April). https://doi.org/10.2118/179569-ms

Guerrero, F. J., Prol-Ledesma, R. M., & Karimi, N. (2020). Transient thermo-solutal convection in a tilted porous enclosure heated from below and salted from above. *International Communications in Heat and Mass Transfer*, *118*(September), 104875. https://doi.org/10.1016/j.icheatmasstransfer.2020.104875

Guerrero Martínez, F. J. (2017). *Three-dimensional numerical models for free convection in porous enclosures heated from below*. University of Glasgow.

Islam, A. W., Sharif, M. A. R., & Carlson, E. S. (2013). Numerical investigation of double diffusive natural convection of CO2 in a brine saturated geothermal reservoir. *Geothermics*, *48*, 101–111. https://doi.org/10.1016/j.geothermics.2013.07.001

Kolditz, O. (2001). Non-linear flow in fractured rock. *International Journal of Numerical Methods for Heat and Fluid Flow*, *11*(5-6), 547–575. https://doi.org/10.1108/eum0000000005668

MacIntyre, K. J. (1986). Design Considerations for Carbon Dioxide Injection Facilities. *Journal of Canadian Petroleum Technology*, *25*(2), 90–95. https://doi.org/10.2118/86-02-09

Newell, P., & Ilgen, A. G. (2018). Overview of geological carbon storage (GCS). In *Science of Carbon Storage in Deep Saline Formations: Process Coupling across Time and Spatial Scales* (pp. 1–13). https://doi.org/10.1016/B978-0-12-812752-0.00001-0

Nield, D. A., & Bejan, A. (2006). *Convection in porous media* (5th ed.). Springer.

Pruess, K. (2006). Enhanced geothermal systems (EGS) using CO2 as working fluid-A novel approach for generating renewable energy with simultaneous sequestration of carbon. *Geothermics*, *35*(4), 351–367. https://doi.org/10.1016/j.geothermics.2006.08.002

Pruess, K., & Azaroual, M. (2006). On the Feasibility of using Supercritical CO2 as Heat Transmission Fluid in an Engineered Hot Dry Rock Geothermal System. *PROCEEDINGS, Thirty-First Workshop on Geothermal Reservoir Engineering*, 1–8. https://doi.org/10.1016/j.apenergy.2014.02.027

Taheri, A., Wessel-Berg, D., Torsaeter, O., & Soroush, M. (2012). The effects of anisotropy and heterogeneity on CO2 dissolution in deep saline aquifers. *Society of Petroleum Engineers - Carbon Management Technology Conference 2012*, *2*(January), 607–616. https://doi.org/10.7122/151345-ms

The Prasino Group. (2012). *Joffre Viking CO 2 Injection Project. Offset Project Report* (Issue March).

Xu, T., & Pruess, K. (2004). Numerical Simulation of Injectivity Effects of Mineral Scaling and Clay Swelling in a Fractured Geothermal Reservoir. *Lawrence Berkeley National Laboratory, Earth Science Division.*, *July*, 1–14.

ANEXO I. CÓDIGO DE PYTHON

```
from ipywidgets import Layout, interact, IntSlider
import ipywidgets as widgets
from IPython.display import display, HTML
import os
from sys import platform
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
import numpy as np
import seaborn as sns
from matplotlib import animation, rc
from time import time


class SimuladorGeo:
    """Clase que contiene los métodos para el funcionamiento y
visualización de resultados de un simulador de convección
termohalina"""
    def __init__(self, rayleigh = 100, radio=10, angulo=10,
pasos=1000, sor_param= 1.2, solver=1 ):
        """Constructor de la clase que recibe como argumento los
parámetros de entrada. si no se definen inicialmente se generan
los parametros por default"""
        self.rayleigh = rayleigh
        self.radio = radio
        self.angulo = angulo
        self.pasos = pasos
        self.sor_param= sor_param
        self.solver= solver


    def IniciarEjercicio(self):
        """Método que permite seleccionar el tipo de
solucionador que el usuario desea correr"""

        print('Elige el tipo de solucionador que deseas
correr:\nEscribe 1 para Gradiente Conjugado\nEscribe 2 para
Gauss-Seidel\nEscribe 3 para SOR.\n')

        solver= int(input('Solucionador: '))
        self.solver=solver

        if solver == 1 or solver == 2:
            print('Define los parametros de entrada para generar
la simulación\n')
            self.interfazGrafica()


        elif solver == 3:
            print('Define los parametros de entrada para generar
la simulación, incluyendo el parámetro de relajación\n')
```

```
            self.interfazGrafica()


        else:
            print('Vuelve a intentarlo')

        return

    def interfazGrafica(self):
        """Método que permite al usuario ingresar los parámetros
de entrada para la simulación de manera dinámica"""

        ## Selecciona el número de Rayleigh
        rayleigh = widgets.FloatSlider(
            value=100,
            min=0,
            max=250.0,
            step=1.0,
            description='',
            disabled=False,
            continuous_update=False,
            orientation='horizontal',
            readout=True,
            readout_format='.1f',
            layout=widgets.Layout(width='100%'))

        ## Selecciona el radio de Bouyancy
        radio = widgets.FloatSlider(
            value=-10.0,
            min=-10.0,
            max=10.0,
            step=1.0,
            description='',
            disabled=False,
            continuous_update=False,
            orientation='horizontal',
            readout=True,
            readout_format='.1f',
            layout=widgets.Layout(width='100%'))

        ## Selecciona el ángulo de inclinacion en grados
        grados = widgets.IntSlider(
            value=10,
            min=0,
            max=30,
            step=1,
            description='',
            disabled=False,
            continuous_update=False,
            orientation='horizontal',
            readout=True,
            readout_format='d',
```

```
            layout=widgets.Layout(width='100%'))

        ## Número de pasos de tiempo
        pasos = widgets.IntSlider(
            value=1000,
            min=100,
            max=100000,
            step=100,
            description='',
            disabled=False,
            continuous_update=False,
            orientation='horizontal',
            readout=True,
            readout_format='d',
            layout=widgets.Layout(width='100%'))

         ## Selecciona el parámetro de relajación if solver == 3
        sor_param= widgets.FloatSlider(
                value=1.2,
                min=0.1,
                max=1.9,
                step=0.1,
                description='',
                disabled=False,
                continuous_update=False,
                orientation='horizontal',
                readout=True,
                readout_format='.1f',
                layout=widgets.Layout(width='100%'))

        button = widgets.Button(
            description='Cambiar parámetros',
            disabled=False,
            button_style='success',
            tooltip='Click me',
            icon='check',
            layout = widgets.Layout(width='40%',
                                    margin='20px 0px 0px 0px'))
        output = widgets.Output()

        titulo = widgets.HTML(
            value="<h3>Definición de parámetros</h3>",
            layout = widgets.Layout(margin='0px 0px 30px 0px'))

        parametro1 = widgets.Label(value= "Número de rayleigh")
        parametro2 = widgets.Label(value= "Relación de
flotación")
        parametro3 = widgets.Label(value= "Inclinación del
ángulo en grados")
        parametro4 = widgets.Label(value= "Número de pasos de
tiempo")
```

```
        parametro5 = widgets.Label(value= "Parámetro de
relajación SOR")

        if self.solver == 1 or self.solver == 2:
            def on_button_clicked(b):
                with output:
                    button.disabled = True
                    self.rayleigh = rayleigh.value
                    self.radio = radio.value
                    self.angulo = grados.value
                    self.pasos = pasos.value
                    button.disabled = False

                self.generaInput()

            button.on_click(on_button_clicked)

            items = [titulo, parametro1, rayleigh, parametro2,
radio, parametro3, grados, parametro4, pasos, button]

        elif self.solver == 3:
            def on_button_clicked(b):
                with output:
                    button.disabled = True
                    self.rayleigh = rayleigh.value
                    self.radio = radio.value
                    self.angulo = grados.value
                    self.pasos = pasos.value
                    self.sor_param = sor_param.value
                    button.disabled = False

                self.generaInput()

            button.on_click(on_button_clicked)

            items = [titulo, parametro1, rayleigh, parametro2,
radio, parametro3, grados, parametro4, pasos, parametro5,
sor_param, button]

        box_layout = widgets.Layout(display='flex',
                            flex_flow='column',
                            align_items='center',
                            padding = '5%',
                            border='solid',
                            width='60%')

        pantallaselec = widgets.GridBox(items, layout =
box_layout)

        display(pantallaselec, output)
```

```
    def generaInput(self):
        """Método que a partir de los parámetros definidos,
genera un archivo .inp que sirve como input del simulador"""

        f= open("in_param.inp","w+")

        if self.solver == 1 or self.solver == 2:

            f.write("Input parameters GEOThermohaline
Simulator:\nRayleigh number between 0 and
250:\nRa="+str(self.rayleigh)+"\nBuoyancy ratio between -10 and
10:\nBR="+str(self.radio)+"\nInclination angle in degrees
(integer between 0 and 30)\nAngle
(degrees)="+str(self.angulo)+"\nNumber of time steps (multiples
of 100, max 100000):\nTime steps="+str(self.pasos)+'\nSelected
solver (1: Conjugate gradient; 2: Gauss-Seidel; 3:
SOR)\nsolver='+str(self.solver))
            f.close()
            print('Se escribió correctamente el archivo de
entrada con los siguientes parámetros: \n\nNúmero de Rayleigh
(Ra): '+str(self.rayleigh)+'\nRelación de flotación (N):
'+str(self.radio)+'\r\nÁngulo de inclinación (θ):
'+str(self.angulo)+'°'+'\nNúmero de pasos de tiempo (n∆t):
'+str(self.pasos)+'\nSolucionador: '+str(self.solver))

        elif self.solver == 3:
            f.write("Input parameters GEOThermohaline
Simulator:\nRayleigh number between 0 and
250:\nRa="+str(self.rayleigh)+"\nBuoyancy ratio between -10 and
10:\nBR="+str(self.radio)+"\nInclination angle in degrees
(integer between 0 and 30)\nAngle
(degrees)="+str(self.angulo)+"\nNumber of time steps (multiples
of 100, max 100000):\nTime steps="+str(self.pasos)+'\nSelected
solver (1: Conjugate gradient; 2: Gauss-Seidel; 3:
SOR)\nsolver='+str(self.solver)+'\nParámetro de relajación SOR
(entre 0.1 Y 1.9)\nsor_param='+str(self.sor_param))
            f.close()
            print('Se escribió correctamente el archivo de
entrada con los siguientes parámetros: \n\nNúmero de Rayleigh
(Ra): '+str(self.rayleigh)+'\nRelación de flotación (N):
'+str(self.radio)+'\r\nÁngulo de inclinación (θ):
'+str(self.angulo)+'°'+'\nNúmero de pasos de tiempo (n∆t):
'+str(self.pasos)+'\nSolucionador:
'+str(self.solver)+'\nParámetro de relajación SOR:
'+str(self.sor_param))

        BtnEjecuta= widgets.Button(
            description='Ejecuta Simulador',
            disabled=False,
            button_style='success',
            tooltip='Click me',
            icon='check',
```

```
            layout = widgets.Layout(width='40%',
                                    margin='20px 0px 0px 0px'))
        def button_clicked(b):
            self.ejecutaSimulador()

        BtnEjecuta.on_click(button_clicked)
        display(BtnEjecuta)

    def ejecutaSimulador(self):
        """Método que ejecuta el simulador a segun el sistema
operativo"""
        start_time = time()

        if platform == "darwin":
          print('Comienza la simulación')
          os.system('Simuladores/GEOThermohaline_macos')
        elif platform == "linux":
          print('Comienza la simulación')
          os.system('Simuladores/GEOThermohaline_linux')
        elif platform == "win32":
            print('Lo sentimos, aún no tenemos versión para
windows')

        elapsed_time = time() - start_time
        print("Simulación concluida. Tiempo transcurrido: %.5f
segundos."  % elapsed_time)

        self.graficaSolucion()

    def graficaSolucion(self):
        """Método que grafica los resultados de la simulación:
el Tiempo vs Numero de Nusselt, Sherwood y Saturación así como
el número de iteraciones para las ecuaciones de velocidad, masa
y calor"""

        fileObj = open('output/run_log.txt',"r")
        lines = fileObj.readlines()
        A = []
        del lines[0:6]
        for line in lines:
            A.append(line.split())

        datos = np.array(A, dtype=np.float32)

        plt.rcParams['axes.grid'] = True

        BtnGrafica= widgets.Button(
            description='Grafica Nu(t), Sh(t) y Sa(t)',
            disabled=False,
            button_style='success',
            tooltip='Click me',
            icon='check',
```

```
            layout = widgets.Layout(width='40%',
                                    margin='20px 0px 0px 0px'))

        BtnDesempeño= widgets.Button(
            description='Grafica el desempeño del solver',
            disabled=False,
            button_style='success',
            tooltip='Click me',
            icon='check',
            layout = widgets.Layout(width='40%',
                                    margin='20px 0px 0px 0px'))

        def button_clicked(b):

            fig, (ax1, ax2, ax3) = plt.subplots(1, 3,
figsize=(16,5))

            ax1.plot(datos[:,0], datos[:,1], '-',
color='skyblue', linewidth=3)
            ax1.set_xlabel('Tiempo')
            ax1.set_ylabel('Número de Nusselt')

            ax2.plot(datos[:,0], datos[:,2]*(-1), '-',
color='skyblue', linewidth=3)
            ax2.set_xlabel('Tiempo')
            ax2.set_ylabel('Número de Sherwood')

            ax3.plot(datos[:,0], datos[:,3], '-',
color='skyblue', linewidth=3)
            ax3.set_xlabel('Tiempo')
            ax3.set_ylabel('Saturación')

            resultados=plt.show()

            display(BtnDesempeño)

            return resultados

        BtnGrafica.on_click(button_clicked)
        display(BtnGrafica)

        def buttonClickedDesempeño(c):

            fig, (ax1, ax2, ax3) = plt.subplots(1, 3,
figsize=(16,5))

            ax1.plot(datos[:,0], datos[:,4], '-',
color='springgreen', linewidth=3)
            ax1.set_xlabel('Tiempo')
            ax1.set_ylabel('Iteraciones Ec. de Calor')
```

```
            ax2.plot(datos[:,0], datos[:,5], '-',
color='springgreen', linewidth=3)
            ax2.set_xlabel('Tiempo')
            ax2.set_ylabel('Iteraciones Ec. de Masa')

            ax3.plot(datos[:,0], datos[:,6], '-',
color='springgreen', linewidth=3)
            ax3.set_xlabel('Tiempo')
            ax3.set_ylabel('Iteraciones Ec. de Velocidad')

            desempeño=plt.show

            return desempeño

        BtnDesempeño.on_click(buttonClickedDesempeño)


    def graficaSimulacion(self):
        """Método que lee los datos de concentración y
temperatura y genera las gráficas de la simulación"""
        datac=np.loadtxt('output/concentration.txt',
dtype='float', skiprows=5)

        steps= int(self.pasos/100)

        conc=np.zeros(((steps,102,102)), dtype= np.float32)
        for k in range(steps):
            conc[k,:,:]=datac[(103*k)+1: (103*k)+103, 1:]

        y1=np.zeros((102,102),dtype= np.float32)
        for j in range(102):
            y1[:,j]=datac[0,1:]


        x1=np.zeros((102,102), dtype= np.float32)
        for i in range(102):
            x1[i,:]=datac[0,1:]

        datat=np.loadtxt('output/temperature.txt',
dtype='float', skiprows=5)

        temp=np.zeros(((steps,102,102)), dtype= np.float32)
        for k in range(steps):
            temp[k,:,:]=datat[(103*k)+1: (103*k)+103, 1:]


        y2=np.zeros((102,102),dtype= np.float32)
        for j in range(102):
            y2[:,j]=datat[0,1:]


        x2=np.zeros((102,102), dtype= np.float32)
```

```
        for i in range(102):
            x2[i,:]=datat[0,1:]

        print('leyendo datos de la simulación...'+'\nGenerando
las gráficas de la simulación')

        plt.style.use('ggplot')

        mycolors= sns.blend_palette(['blue', 'orange'
,'yellow'], as_cmap=True)

        fig, (ax1, ax2) = plt.subplots(1,2, figsize=(12,6))
        plt.suptitle('Ra='+str(self.rayleigh)+'
N='+str(self.radio)+'  θ='+str(self.angulo)+'
Δt='+str(self.pasos), fontsize=10, fontweight='bold')
        plt.subplots_adjust(left=0.13, right=0.93, bottom= 0.27)

        cax1 = ax1.pcolormesh(x1, y1, conc[1,:,:],
                      vmin=0, vmax=1, cmap= mycolors,
                      shading='gouraud')
        fig.colorbar(cax1, ax=ax1)
        ax1.set_title("Concentración",fontsize=12, pad=10)

        cax2 = ax2.pcolormesh(x2, y2, temp[1,:,:],
                    vmin=0, vmax=1, cmap='plasma',
                    shading='gouraud')
        fig.colorbar(cax2, ax=ax2)
        ax2.set_title("Temperatura", fontsize=12, pad=10)
        plt.close(fig)

        cax=[cax1, cax2]

        ## Función que genera una animación para visualizar la
simulación
        def animate(frame):
            cax[0].set_array(conc[frame,:,:].flatten())
            cax[1].set_array(temp[frame,:,:].flatten())

        anim= animation.FuncAnimation(fig, animate,
frames=steps, interval=100)
        video=HTML(anim.to_html5_video())

        return video
```