

Diseño de Software para Comparar los Algoritmos de Stehfest e Iseger para Inversión Laplaciana en Pruebas de Presión

Software Design to Compare Stehfest's and Iseger's algorithms for Laplacian Inversion in Pressure Well Tests

Freddy Humberto Escobar¹, Fabián Andrés Leguizamo² y José Humberto Cantillo³

Resumen

Este artículo presenta el diseño de un software cuyo objetivo es realizar la comparación de dos algoritmos que sirven para determinar la transformada inversa de Laplace. Esta comparación se lleva a cabo mediante la evaluación dos modelos matemáticos de pozos muy comunes en la industria petrolera que están dados en el dominio de Laplace, para los cuales se realiza la inversión y se grafican los resultados para establecer comparación. La herramienta de software se desarrolla en Visual Basic 2008, y consta de una interfaz gráfica que permite visualizar las inversiones realizadas por cada algoritmo y también genera dos archivos de texto donde se encuentran los datos de tiempo, presión y derivada de presión. La interfaz gráfica brinda varias opciones como poder escoger el modelo de pozo deseado o introducir mediante teclado otro modelo o función que se quiera invertir, también ofrece la opción de suprimir o agregar la gráfica de la derivada y de realizar las gráficas en escala logarítmica.

El estudio permitió establecer que el algoritmo de Iseger maneja más eficazmente las discontinuidades de una función y se observó que para casos de pruebas de presión que al cumplir las condiciones que relacionan el tiempo con el número de puntos muestreados, se garantizan unos resultados de inversión más estables. También se comprobó que entre más puntos de inversión se usen, se obtienen unos resultados más precisos. Se encontró, además, en el parámetro n_{rp} de 3 genera inversiones más estables que el propuesto originalmente por Iseger con valor de 8.

Palabras Clave: Laplace; interfaz gráfica, pruebas de pozos, yacimiento homogéneo, yacimiento heterogéneo.

Abstract

This paper presents a software design with the purpose of comparing two algorithms used for the inverse Laplace transform determination. The comparison was applied to two well-known oil-industry reservoir models in the Laplacian domain, for which the inversion is made and the results are plotted to establish comparison. The code is written in Visual Basic 2008, and consists of a graphical interface that displays the inversions made by each algorithm and also generates two output text files which include data of time, pressure and pressure derivative. The graphical interface allows to choose the desired well model or to introduce another model or function to invert using the keyboard. It also offers the option of removing or adding the plot of the derivative and making plots in logarithmic scale.

The study allowed to establish that Iseger's algorithm handles more efficiently function discontinuities. It was observed for well tests that fulfilling the conditions relating time with sample number size provides more stable inversions. It was also proved that the greater the number of points to invert, the more accurate the solution. Besides, parameter n_{rp} originally set by Iseger to be 8, was found to provide more stable solutions with a value of 3.

Keywords: Laplace, graphics interface, well tests, homogeneous reservoir, heterogeneous reservoir

¹ Ingeniero de Petróleos Ph.D., Docente Universidad Surcolombiana - Neiva. Av. Pastrana - Carrera 1. fescobar@usco.edu.co

² Ingeniero Electrónico. Universidad Surcolombiana - Neiva. Av. Pastrana - Carrera 1. fabian.leguizamo@gmail.com

³ Ingeniero de Petróleos, M.Sc. Ecopetrol. Km 7 vía Pídecueta, Bucaramanga, jose.cantillo@usco.edu.co

1. Introducción

Desde el trabajo seminal de van Everdingen y Hurst (1953), la transformada de Laplace ha sido una herramienta estándar para problemas de transientes en el flujo de fluidos en medios porosos. La transformada de Laplace se usa sobre todo para la solución de problemas de valor inicial de frontera. Para algunas de estas aplicaciones, una inversión exacta (analítica) no es posible y la inversión numérica es el único recurso. Para algunos otros, la inversión numérica es escogida también por su conveniencia. Para esta inversión se usan software basados en el algoritmo de Stehfest (1970) el cual tiene algunas limitaciones como lo es por ejemplo el manejo de funciones discontinuas, en este caso la inversión realizada usando el algoritmo de Iseger no presenta este problema (Al-Ajmi *et al.*, 2008).

El desarrollo de software para diferentes tipos de sistemas es una herramienta muy importante porque permite que analistas e ingenieros realicen pruebas, cambios y traten de resolver cualquier duda mediante este tipo de software que proporciona un ambiente virtual que se comporta lo más cercano posible a la realidad y facilita luego la puesta en marcha, así como también puede ayudar a mejorar el funcionamiento y desempeño del sistema en cuestión.

En el caso específico de la industria petrolera se hace indispensable poder contar con estos simuladores que sean capaces de realizar diversas mediciones de todas y cada una de las variables que influyen en el comportamiento de un yacimiento que hacen que éste se desempeñe de cierta forma.

Actualmente en Colombia, no se desarrollan este tipo de simuladores y es por esto que la industria de software está aún incipiente, por tanto es una muy buena oportunidad lograr incursionar en este campo, en donde como ingenieros se aprendería de una temática inexplorada y mostrarse ante la industria nacional e internacional.

Las pruebas de presión tienen múltiples aplicaciones. Entre otros se pueden mencionar: determinar los límites del yacimiento (obviamente reservas de hidrocarburos), su comerciabilidad, capacidad conductora de fluidos (permeabilidad) de la roca productora, hallar distancias a fallas y a otras barreras, determinar la presión promedio del yacimiento (un parámetro cuyo conocimiento es tan importante para el ingeniero como la presión arterial del paciente lo es al médico), y cuantificar el daño (Escobar, 2009).

El objetivo primario del análisis de prueba de pozos es identificar el modelo de reservorio y estimar sus propiedades a partir de la respuesta de presión. Ésto se consigue construyendo un modelo matemático, el cual genera una respuesta de salida similar al del sistema real. Este modelo matemático se resuelve en el dominio Laplace y luego la solución se invierte al dominio del tiempo, para lo cual se utilizan los algoritmos de inversión numérica.

Muchos algoritmos han sido propuestos, como lo son el de Bellman *et al.* (1966), Crump (1976) y Talbot (1979), pero no tuvieron mucha aceptación. El método más usado en las pruebas de presión es el algoritmo de Stehfest (1970), es el único presente en las aplicaciones comerciales diseñadas para obtener la inversa de Laplace en el campo de la ingeniería de petróleos, pero presenta inconvenientes para manejar funciones singulares y/o discontinuas. El algoritmo de Iseger (2006) elimina este tipo de restricciones y brinda oportunidades para muchas aplicaciones prácticas.

La parte fundamental de esta investigación está orientada hacia la programación de los algoritmos de Stehfest e Iseger con el fin de obtener la transformada inversa de Laplace de una función, herramienta utilizada en muchos problemas de pruebas de pozos y determinar cuál de los dos métodos realiza la inversión de modo más preciso, adecuado y práctico.

2. Metodología

2.1 Recopilación y selección de información

Se realiza una consulta y se recolecta información sobre el algoritmo de Iseger para realizar la implementación de este algoritmo. Se busca documentación sobre la inversión numérica de la transformada de Laplace aplicada a la solución de problemas de transientes en pozos y la utilización de diversos algoritmos para lograr este fin, de este modo poder tener una guía para el desarrollo de este trabajo. Posteriormente se investigó la utilización del algoritmo de Iseger en el campo de las pruebas de presión para tener en cuenta observaciones que se hayan tenido en investigaciones previas sobre el comportamiento y desempeño de este método en el manejo de discontinuidades y singularidades.

2.2 Elección del lenguaje y entorno a utilizar

Para la elección del entorno que se iba a utilizar para el desarrollo de la herramienta de software, se tuvo en cuenta la facilidad de implementación pero sin descuidar la precisión y confiabilidad de los resultados que podría proporcionar el

software. El Visual Studio 2008 cuenta con herramientas muy versátiles que facilitan la creación de un entorno gráfico bajo el sistema operativo Windows; específicamente se utilizó el Visual Basic. Esto también contribuye a futuras mejoras de este software y a la integración de este mismo a un macro proyecto que se viene adelantando en este mismo lenguaje.

El algoritmo de interés, el de Iseger (2006), trabaja en el plano de los números complejos. Debido a que el Visual Studio 2008 no cuenta con herramientas propias para el manejo de los números complejos, fue necesario la utilización de unas librerías externas que contarán con todo lo necesario para la declaración y operación de variables complejas de alta precisión.

2.3 Revisión de la información recopilada, interpretación y elección del algoritmo

Se revisó la información recopilada, entre la cual estaba el documento original del autor del algoritmo, el señor Peter Den Iseger, en el cual se muestra el esquema del método, un algoritmo de inversión sencillo y múltiples modificaciones que le permiten manejar una gama más amplia de funciones, hasta llegar al que él llama “el algoritmo robusto para la inversión de transformadas de Laplace”. Esta última variante del algoritmo fue la implementada debido a que tiene las modificaciones adecuadas para poder manejar con éxito funciones discontinuas, singulares y demás, incluso sin conocer de antemano estas discontinuidades y singularidades.

Entre la información también se encontró un documento que fue de mucha ayuda, una gran guía para el desarrollo del software comparativo aquí presentado.

2.4 Modelos de pozos y funciones para comparación

Para llevar a cabo la comparación entre los algoritmos de Stehfest e Iseger, se utilizaron dos modelos de pozos dados. El primero es el modelo de yacimiento homogéneo infinito y el segundo el yacimiento naturalmente fracturado. Estos dos modelos son comunes en las pruebas de presión de pozos y la evaluación de éstos permite comparar los resultados obtenidos por ambos algoritmos.

Para comprobar el buen funcionamiento del algoritmo de Iseger en funciones discontinuas, como por ejemplo funciones tipo paso o multipaso, se decidió integrar un código que permitiera interpretar ecuaciones ingresadas por teclado; de este modo se puede invertir cualquier función deseada que esté en el dominio Laplace al dominio del tiempo.

2.5 Métodos de comparación

Para la comparación de los resultados obtenidos en la inversión de la transformada de Laplace por los dos algoritmos presentes en el software, se decidió incluir campos de gráficas, donde se realicen las gráficas de presión y derivada de presión vs tiempo, datos obtenidos mediante la inversión realizada por ambos métodos; también opciones que permitan escoger si los parámetros van a ser dimensionales o adimensionales, graficar solo la presión y que la escala sea logarítmica. Aquí se puede apreciar la estabilidad de cada método y sus diferencias.

Se generan archivos de texto plano que incluyen los datos obtenidos y que se usan para las gráficas. Así se puede establecer una comparación de los datos punto a punto.

2.6 Algoritmo de Iseger

El renovado interés en la inversión numérica de funciones singulares y discontinuas ha sido provocado por un nuevo algoritmo presentado por Iseger que elimina la restricción de continuidad que presentan algunos de los algoritmos de inversión más comunes y brinda la posibilidad de afrontar con éxito muchas aplicaciones prácticas.

2.6.1 Esquema del método

El algoritmo propuesto por Iseger es un método de series de Fourier, que está basado en la fórmula de sumatoria de Poisson. La sumatoria de Poisson relaciona una suma infinita de valores de transformada de Laplace con la transformada Z de los valores de la función. La suma infinita es aproximada por una suma finita basada en la regla de cuadratura Gaussiana y los valores en el dominio del tiempo de la función son calculados por un algoritmo de transformada de Fourier. Los resultados que ofrece son de precisión de máquina (Al-Ajmi *et al.*, 2008).

Lo atractivo del algoritmo de Iseger radica en su habilidad para calcular la transformada inversa de Laplace de funciones con toda clase de discontinuidades, singularidades y sin suavizado local, aunque cabe destacar que la implementación de este algoritmo puede ser un poco más complicada que la implementación del algoritmo de Stehfest, pero es comparable a la implementación de otros algoritmos comunes.

En el algoritmo de Iseger se encuentran varios parámetros para tener en cuenta debido a su importancia en los resultados obtenidos. Un parámetro crítico usado es Δ (delta), el cual se puede hallar mediante la siguiente relación:

$$\Delta = \frac{T}{M}$$

Donde T es el período para el cual son calculadas las inversiones y M es el número de puntos para los cuales es calculada la inversa de Laplace.

El error de aproximación también depende de Δ ; para garantizar resultados precisos y estables se deben cumplir dos condiciones. La primera condición que debe cumplirse es que $(1/\Delta) \in \mathbb{N}$ y además se debe cumplir que,

$$M \geq 10\Delta M, \text{ lo que equivale a decir que } M \geq 10T \text{ (Al-Ajmi et al., 2008).}$$

Al cumplir estas condiciones se garantizan resultados estables y precisos en la inversión de la transformada de Laplace. Es importante destacar que al tener un mayor número de puntos de muestra también se tiene una mayor precisión, por lo que para obtener inversiones estables y precisas se requiere que M sea lo más grande posible.

El algoritmo de Iseger (2006) usa $M_s = nrp \times M$ puntos de sobremuestreo para calcular las inversiones en M puntos de datos. El tiempo de ejecución real es $M_s \log(M_s)$; por lo tanto, desde esta perspectiva, se desea escoger M_s tan pequeño como sea posible. Sin embargo, para obtener un resultado numéricamente estable, se debe escoger M_s tan grande como sea posible (Al-Ajmi et al., 2008).

Para el parámetro nrp , dado por el autor, no se especifica el significado de las letras pero sí menciona que éste es el factor de sobremuestreo, lo que incrementa la precisión de los resultados. Iseger (2006) recomienda usar $M_s = 8M$, es decir recomienda usar $nrp = 8$, especialmente para las funciones de buen comportamiento. Esta elección particular para nrp se hizo debido a las condiciones impuestas por la Transformada Rápida de Fourier usada originalmente por Iseger, que requiere que M y nrp sean potencias de 2 (Iseger, 2006).

En el algoritmo de Iseger se definió el parámetro $n = 16$, por lo tanto las constantes beta y lambda para este valor de n se definieron de acuerdo a los valores brindados por el autor.

- n=16 para todas las funciones suaves.
- n=32 para las funciones singulares muy oscilantes.
- n=48 cuando la función tiene muchos picos.

El autor muestra en su documento varios algoritmos con diversas modificaciones. El usado aquí es el más completo, es decir el que tiene una aplicación más amplia, aquel que demostró manejar sin problema discontinuidades de cualquier tipo y que sin importar donde se encuentre la discontinuidad no presenta problema, por lo tanto su rango de confianza en cuanto al manejo de discontinuidades no está limitado (Iseger, 2006).

2.6.2 Evaluación del método y resultados esperados

Aquí se muestran varios casos en los cuales se puede afectar la estabilidad y precisión de la inversión realizada por el algoritmo de Iseger.

Se consideró una función paso unitario con un período arbitrario de prueba de $T=20$ horas. Tomando en cuenta las condiciones anteriores el número de puntos muestreados debe ser $M \geq 200$. Para mostrar el efecto del número de puntos de inversión en los resultados se consideró realizar la inversión con tres valores diferentes de M.

En la figura 1 se puede observar que las inversiones con 103 datos y 233 datos no son adecuadas y pierden la naturaleza de la función en el punto de discontinuidad. Con 103 datos no se cumple ninguna de las condiciones de estabilidad anteriormente mencionadas y con 233 solo se cumple una. La inversión deseada se obtiene usando 220 datos de muestra, con esta elección se cumplen ambas condiciones.

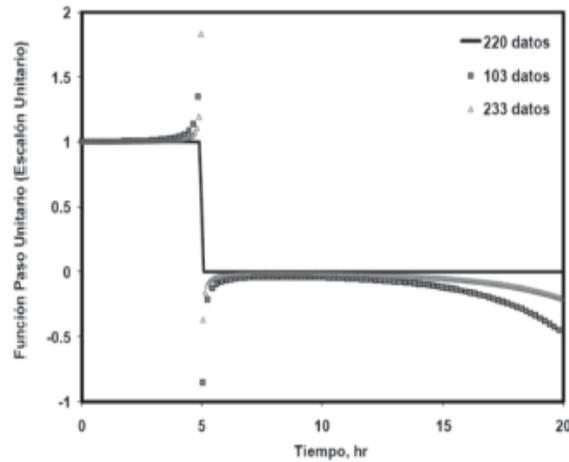


Fig. 1. Inversión numérica de una función escalón unitario por el algoritmo de Iseger. Efecto del número de puntos de inversión (Al-Ajmi *et al.*, 2008)

Para el tipo de aplicaciones de este proyecto no se obtuvieron resultados estables con $n_{rp} = 8$. Se estableció, mediante pruebas, que usando Transformada de Fourier Discreta y $n_{rp} = 3$, se generaron inversiones más estables para estas aplicaciones, las cuales son comunes en el campo de la ingeniería de petróleos (Al-Ajmi *et al.*, 2008). La figura 2 muestra el efecto de n_{rp} en la inversión numérica de una función típica de presión encontrada en problemas de flujo de fluidos usando el algoritmo de Iseger.

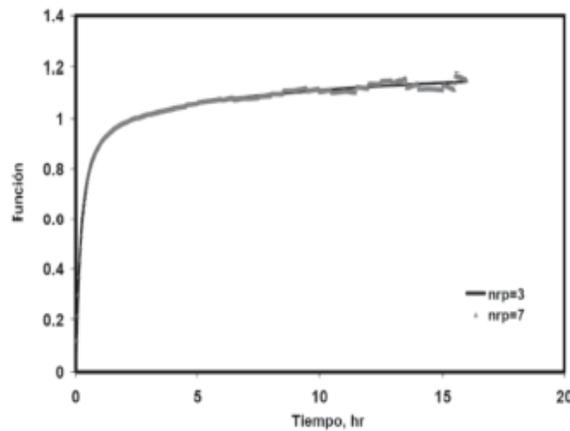


Fig. 2. Efecto de n_{rp} en la inversión numérica de una función por el algoritmo de Iseger (Al-Ajmi *et al.*, 2008)

En los siguientes ejemplos se muestra el éxito del algoritmo de Iseger en el manejo de funciones que son derivables por partes o discontinuas. Se establece una comparación con el algoritmo de Stehfest para trazar las diferencias respecto a los algoritmos estándar.

La figura 3 muestra la inversión de una función escalón unitario (función Heaviside) con el polo en $t = 10$ horas usando los algoritmos de Iseger y Stehfest. Para el algoritmo de Iseger, la inversión se llevó a cabo con $M = 220$ y $n_{rp} = 3$. Para el algoritmo de Stehfest, se usaron tres valores del parámetro $N = 6, 8$ y 12 (N controla el número de evaluaciones funcionales en el algoritmo de Stehfest y teóricamente, valores más altos de N ofrecen mejores resultados). La figura 3

muestra que el algoritmo de Iseger recupera la verdadera naturaleza de paso a paso mientras que el algoritmo de Stehfest daña la función alrededor del punto de discontinuidad ($t = 10$ horas).

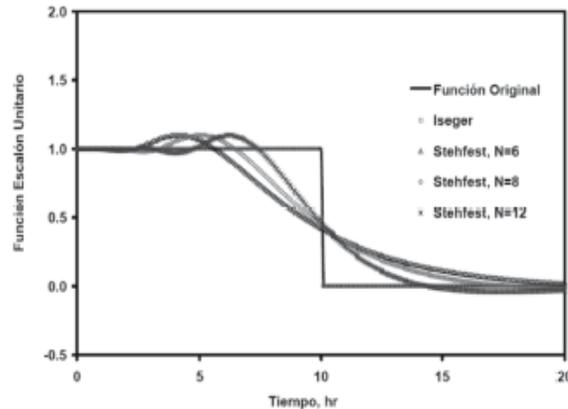


Fig. 3. Inversión numérica de una función escalón unitario; comparación de los algoritmos de Iseger y Stehfest (Al-Ajmi *et al.*, 2008)

En la figura 4 se considera una función con múltiples cambios tipo escalón. Esta función corresponde al tipo de secuencia usado en un ejemplo de deconvolución sintética que se muestra a continuación. Para los resultados en la figura 4, se usó $M = 512$ y $mp = 3$ en el algoritmo de Iseger y $N = 6, 8$ y 12 en el algoritmo de Stehfest. El daño de la función en los puntos de discontinuidad con el algoritmo de Stehfest destruyen completamente las características locales y globales de la función original. El éxito del algoritmo de Iseger en este ejemplo es notable.

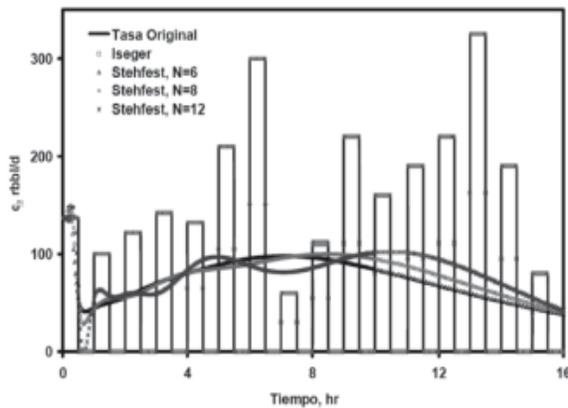


Fig. 4. Inversión numérica de una función con múltiples cambios tipo escalón; comparación de los algoritmos de Iseger y Stehfest (Al-Ajmi *et al.*, 2008)

Para demostrar que los algoritmos especiales, tales como el algoritmo de Iseger, pueden ser necesarios incluso cuando la función es continua pero solo derivable por partes, se presentan los resultados de la figura 5. Para generar la función de este ejemplo se usó la función de la figura 5 como el tipo de secuencia y se generaron los cambios de presión correspondientes. La transformada de Laplace de esta función de cambios de presión tabulados se obtuvo por el algoritmo de Rouboutsos – Stewart (el algoritmo Omur – Reynolds arrojó los mismos resultados) y se invirtió usando los algoritmos de Iseger y Stehfest. Como se muestra en la figura 5, mientras el algoritmo de Stehfest produce una función suave y continuamente derivable, el algoritmo de Iseger lleva a cabo una inversión precisa que conserva todas las características de la función derivable por partes.

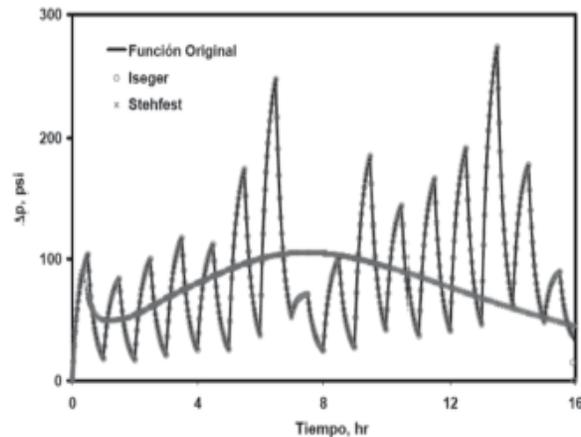


Fig. 5. Inversión numérica de una función derivable por partes; comparación de los algoritmos de Iseger y Stehfest (Al-Ajmi *et al.*, 2008)

Como se verá más adelante, se traza el error con respecto al algoritmo de Stehfest. A pesar del enorme éxito del algoritmo de Iseger manejando discontinuidades, es notable el aumento del tiempo de ejecución cuando se aumenta el tiempo de inversión que a su vez aumenta los puntos de muestreo de la función. Esto se debe a la complejidad del algoritmo, a la cantidad de cálculos que debe realizar y a que el tiempo y el número de muestras son directamente proporcionales, además de que se especifica que el tiempo debe ser al menos 10 veces mayor que el número de muestras para que los resultados tengan una mayor estabilidad y precisión.

2.7 Desarrollo del software

Como se habían mencionado, el software se realizó utilizando Visual Basic .NET 2008 y tiene como objetivo comparar dos algoritmos que se usan para obtener la transformada inversa de Laplace de dos modelos de yacimientos dados y de cualquier función que se desee ingresar mediante teclado.

Se incluyó el algoritmo de Stehfest, que sirve como punto de comparación, y para establecer comparación punto a punto con los resultados obtenidos por Iseger, se realizó un ajuste en el tiempo del algoritmo de Stehfest para que se incrementara de manera análoga al algoritmo de Iseger.

Para el manejo de las funciones complejas, presentes en el algoritmo de Iseger, se utilizó un complemento llamado "Extreme Optimization Numerical Libraries for .NET 3.6" que contiene todas las herramientas necesarias para el manejo y operación de variables y funciones complejas.

Después de revisar la documentación e investigaciones sobre el tema, se modificaron parámetros para obtener resultados más precisos y estables. Uno de los parámetros es mp que en el documento original se recomienda un valor que sea potencia de 2 y en particular recomiendan el valor de 8 para funciones bien planteadas, pero para las aplicaciones que se abordan en este proyecto no se obtienen resultados estables con $mp = 8$; luego se estableció, mediante pruebas, que usando $mp = 3$ con la transformada de Fourier discreta se obtienen resultados más estables para estas aplicaciones (Al-Ajmi *et al.*, 2008).

Otro de los parámetros que se ajustó es Δ , parámetro crítico del algoritmo de Iseger, para el cual se agrega un bucle en el código para que garantice el cumplimiento de la condición $M \geq 10\Delta M$ y así los resultados obtenidos en la inversión sean más estables y precisos.

Las funciones de los modelos de yacimientos tienen funciones de Bessel. Para el correcto funcionamiento del algoritmo de Iseger fue necesario buscar estas funciones de Bessel en el plano complejo e incluirlas en el código. Las funciones de Bessel en el plano real, usadas por Stehfest, no presentaron ningún problema y ya se contaba con ellas.

2.8 Funciones del software

Al iniciar el programa, la interfaz gráfica nos muestra varias opciones entre las que se encuentran los espacios para modificar parámetros en el análisis de pruebas de presiones, las casillas para graficar derivada y usar escala logarítmica, el menú desplegable para dimensional o adimensional, el menú desplegable para el modelo de yacimiento o si se quiere ingresar una función, si este último es escogido se activa la casilla función para ingresar la función deseada mediante teclado, y se encuentran los campos de gráficos donde se muestran las gráficas correspondientes a la presión (y derivada de presión si se escoge) vs tiempo obtenidos por cada algoritmo. La figura 6 muestra el entorno de inicio del programa.

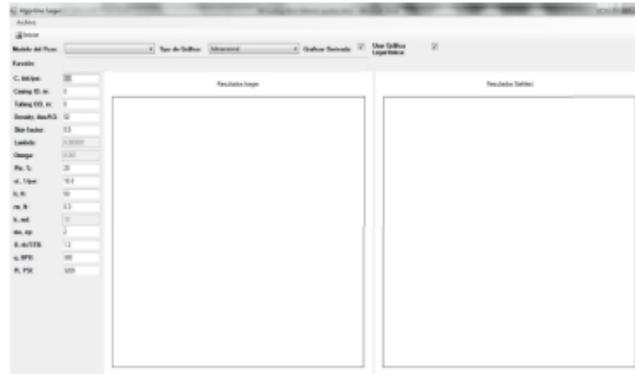


Fig. 6. Espacio de trabajo del programa

Para operar el programa se debe proceder de la siguiente forma:

- a. Se inicia el ejecutable del programa.
- b. Posteriormente se escoge el modelo de pozo deseado o la opción de “Otro modelo o función” si se quiere ingresar otra función mediante teclado. Si se escogió esta última opción, se procede a ingresar la función en dominio Laplace por medio del teclado.
- c. Se escogen las opciones que se deseen, si se desea graficar la derivada de presión (además de la presión), se marca la casilla correspondiente. Si se desea un gráfico en escala logarítmica se marca la casilla de esta función, de lo contrario se deja desmarcada. También es posible escoger si se desea que los datos usados para la gráfica sean dimensionales o adimensionales.
- d. Se procede a iniciar la inversión dando click en el botón “Iniciar”, aquí se generan los archivos de texto plano con los datos obtenidos en la inversión por ambos métodos. También se visualizan las gráficas deseadas en los campos de gráficas correspondientes.

En la figura 7 se muestran varias opciones presentes en la interfaz del software desarrollado.

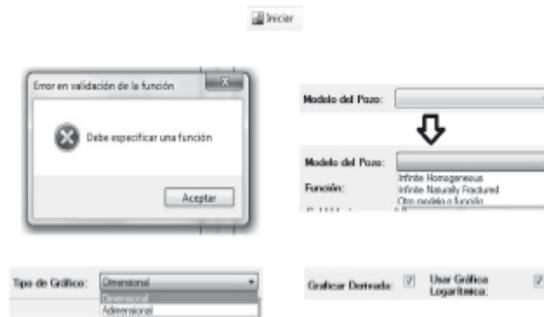


Fig. 7. Varias opciones y advertencias del software.

Si al iniciar la inversión no se ha escogido un modelo de pozo, el programa muestra una advertencia sobre esto. Al escoger el modelo de pozo o ingresar una función, se procede a modificar las opciones necesarias para el gráfico deseado. Así mismo se pueden modificar los parámetros como almacenamiento, factor de daño, etc. Luego de iniciar la inversión, aparece una barra de progreso que desaparece cuando termina el proceso. En la figura 8 se muestra la apariencia del apartado para modificar los parámetros y de la barra de progreso al ejecutar el programa.



Fig. 8. Parámetros y barra de progreso

3. Pruebas del software y resultados

Se realizaron pruebas utilizando diversas funciones. Para todas las pruebas se estableció un tiempo arbitrario de prueba de $T=20$ y para cumplir las condiciones de estabilidad se escogió un número de puntos de inversión $M=220$. Entre las funciones ingresadas para realizar la inversión por el algoritmo de Iseger se encuentran el escalón unitario, la rampa, escalón unitario con retraso, escalón múltiple y la función seno. En la figura 9 se ven los resultados de inversión para las funciones escalón unitario y rampa por el algoritmo de Iseger.

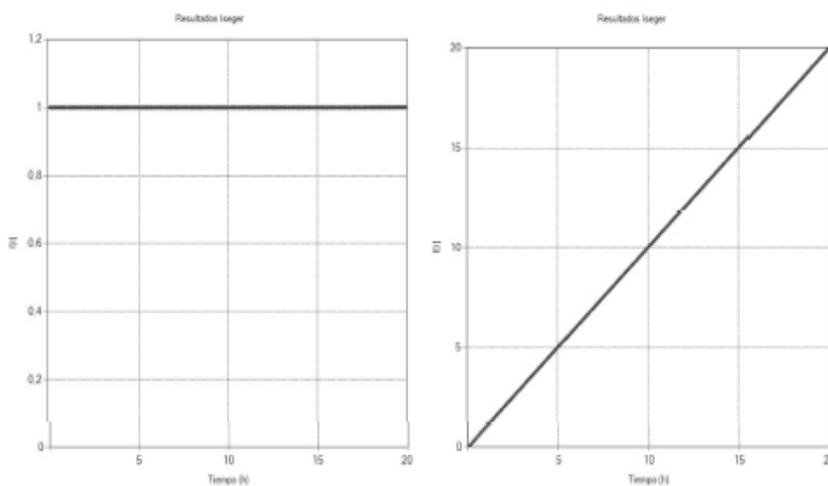


Fig. 9. Inversión de funciones escalón unitario y rampa por algoritmo Iseger

En la figura 10 se ven los resultados obtenidos al realizar la inversión de las funciones escalón unitario con retraso y escalón múltiple por el algoritmo de Iseger.

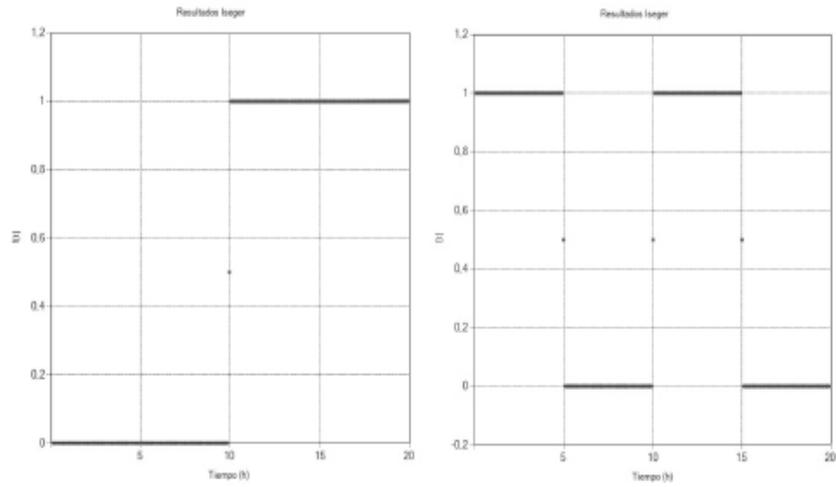


Fig. 10. Inversión de funciones escalón unitario con retraso y escalón múltiple por algoritmo de Iseger

Por último, antes de evaluar los yacimientos dados, se evaluó la función seno por el algoritmo de Iseger.

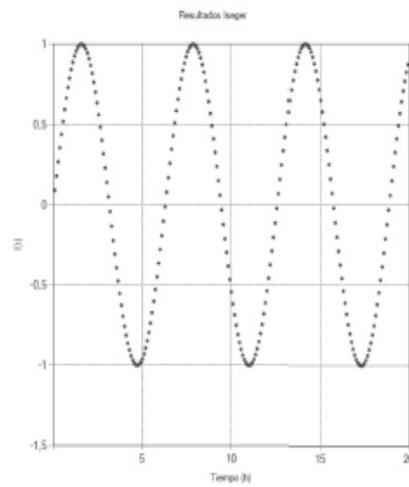


Fig. 11. Inversión de la función seno por algoritmo de Iseger

3.1 Análisis de Yacimiento Homogéneo Infinito

A continuación, en la figura 12 se muestran las inversiones realizadas para este modelo por ambos algoritmos en escala logarítmica:

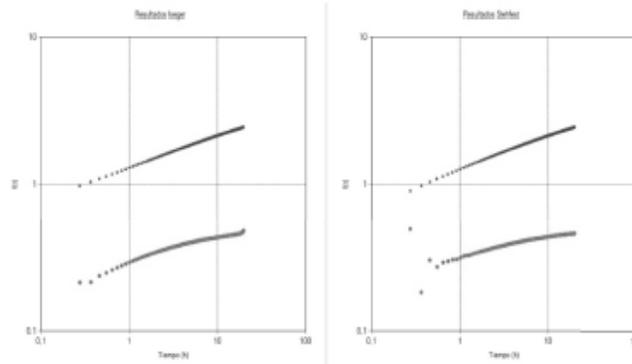


Fig. 12. Resultados inversión del modelo homogéneo infinito

Los resultados de la inversión son muy similares, en cuanto a la gráfica de presión (azul) la similitud es altísima. En la derivada se nota una diferencia al inicio donde Stehfest presenta una dispersión mientras que Iseger estabiliza más rápido.

Para ver de manera más clara las variaciones entre uno y otro método, se trazó una gráfica de la diferencia de presión entre ambos algoritmos en cada punto en el tiempo cuyo valor se ajustó a 20. La figura 13 muestra esta diferencia de presión y su respectivo error.

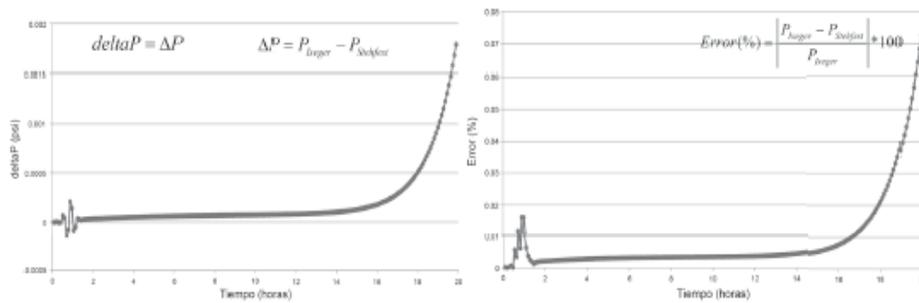


Fig. 13. Gráfica de ΔP y del error (%) contra el tiempo para el yacimiento homogéneo infinito

Tal y como muestran la gráficas, la diferencia entre las presiones es mínima. Al principio se muestra una variación en forma de distorsión de la gráfica pero luego estabiliza, solo hay una infima variación hasta la hora 15 y a partir de ahí aumenta ligeramente pero sigue siendo una variación muy baja. Esta variación puede interpretarse como una acumulación de error, por lo cual el aumento en la zona de la hora 15 a la 20 es mayor. De esto se concluye que los resultados de la inversión del yacimiento homogéneo infinito por ambos algoritmos, son muy parecidos y que la variación entre ambos resultados es mínima.

3.2 Análisis del Yacimiento Naturalmente Fracturado

Para este modelo se repite el proceso, aunque difieren en un solo aspecto. Para este modelo hay dos parámetros que se pueden modificar y que en el anterior modelo no se podía; estos parámetros son Lambda y Omega y los valores que se asumieron fueron 0.000001 y 0.001 respectivamente. Se reportan en la figura 14 las gráficas de los resultados obtenidos para este yacimiento:

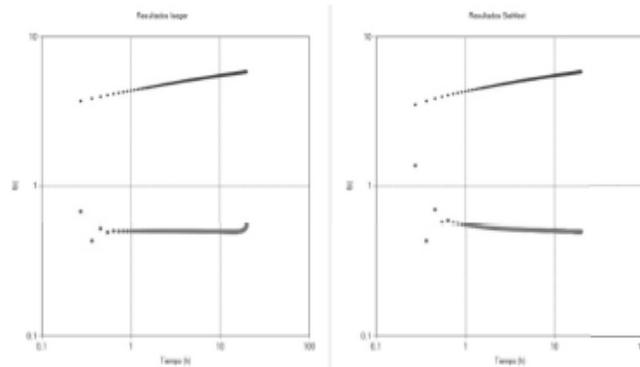


Fig. 14. Resultados inversión del modelo naturalmente fracturado

Para este yacimiento se observa que para la presión (azul) ambas inversiones son muy similares, las diferencias son imperceptibles, aunque para la presión en Stehfest existe una pequeña curvatura antes de la hora 1. En la derivada se nota la estabilidad de la inversión por Iseger al principio siendo los puntos bastante continuos, por otro lado, en la inversión por Stehfest hay una notable dispersión en los primeros puntos. Después, la pendiente casi cero de la gráfica de Iseger se mantiene mientras que en Stehfest hay una pendiente negativa, una curva que baja y luego tiende a estabilizar.

Se trazó una gráfica de la diferencia de presión entre ambos algoritmos en cada punto en el tiempo cuyo valor se ajustó a 20. Esto se hizo para apreciar mejor las diferencias entre los resultados de ambos algoritmos. En la figura 15 se muestran las gráficas correspondientes a la diferencia de presión y al error.

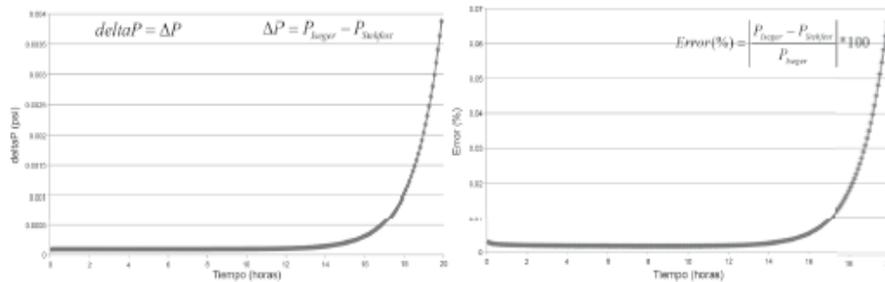


Fig. 15. Gráfica de ΔP y del error (%) contra el tiempo para el yacimiento naturalmente fracturado

En este modelo de yacimiento la diferencia entre las presiones también es mínima, aunque aquí, no hay variación notable al principio y la diferencia se mantiene muy baja y estable hasta la hora 15; a partir de ahí aumenta ligeramente pero sigue siendo una variación muy baja. Como se dijo anteriormente, esta variación puede interpretarse como una acumulación de error, por lo cual el aumento en la zona de la hora 15 a la 20 es mayor. De esto se concluye que los resultados de la inversión del yacimiento naturalmente fracturado por ambos algoritmos, son muy parecidos y que la variación entre ambos resultados es mínima.

4. Conclusiones

Se desarrolló una aplicación de software amigable que permite obtener la transformada inversa de Laplace para dos modelos de yacimientos específicos usando los algoritmos de Stehfest y de Iseger, con una interfaz gráfica que permite comparar los resultados obtenidos mediante ambos algoritmos para poder observar mejor el rendimiento y las diferencias entre ambos. Esta aplicación también puede realizar la inversión del dominio Laplace al dominio del tiempo de cualquier función que se introduzca.

Se eliminaron los problemas de discontinuidades mediante el método de Iseger. Se pudo comprobar que el método maneja de manera adecuada las discontinuidades. Esto se corroboró al realizar la inversión de funciones tipo escalón y

escalón múltiple donde los cambios instantáneos son evidentes y a pesar de esto el resultado de la inversión del algoritmo de Iseger conserva toda la naturaleza y forma de la función original.

Se observó en las pruebas que al cumplir las condiciones que relacionan el tiempo con el número de puntos muestreados, se garantizan unos resultados de inversión más estables. También se comprobó que entre más puntos de inversión (o de muestreo) se usen, se obtienen unos resultados más precisos, pero al usar una mayor cantidad de puntos también aumenta el tiempo de prueba debido a la relación que manejan mediante el parámetro Δ , por lo tanto, el tiempo de ejecución del algoritmo aumenta debido a que la cantidad de cálculos necesarios para realizar la inversión también aumentan y la diferencia es notable respecto al algoritmo de Stehfest que demora mucho menos, lo que se considera una desventaja clara a la hora de realizar un análisis con un tiempo muy grande.

El parámetro n_{rp} fue indicado por el autor del algoritmo, el señor Peter Den Iseger, como una potencia de 2 y específicamente recomendaba el valor de 8, pero se concluyó que para este tipo de aplicaciones desarrolladas en el proyecto se generaron inversiones más estables usando Transformada de Fourier Discreta y un valor de $n_{rp} = 3$.

5. Referencias Bibliográficas

1. Al-Ajmi, N., Ahmadi, M., Ozkan, E. and Kazemi, H. T. A. 2008. Numerical Inversion of Laplace Transforms in the Solution of Transient Flow Problems With Discontinuities. Paper SPE 116255 presented at the SPE Annual Technical Conference and Exhibition. Denver, Colorado. 21-24 Sep.
2. Bellman, R., Kalaba, R. E., and Lockett, J. A. 1966. Numerical Inversion of Laplace Transform: Applications to Biology, Economics, Engineering, and Physics. American Elsevier Publishing Co. Inc., New York, NY.
3. Crump, K. S. 1976. Numerical Inversion of Laplace Transforms Using a Fourier Series Approximation. *J. ACM* 23(1): 89-96.
4. Escobar, M. F. 2009. Análisis Moderno de Pruebas de Presión. Editorial Universidad Surcolombiana. Segunda edición.
5. Iseger, P. D. 2006. Numerical Transform Inversion Using Gaussian Quadrature. *Probability in the Engineering and Informational Sciences*, 20. 1-44.
6. Stehfest, H. 1970. Numerical Inversion of Laplace Transforms. *Communications, ACM* 13 (1): 47-49.
7. Talbot, A. 1979. The Accurate Numerical Inversion of Laplace Transforms, *J. Inst. Maths. Applics.*, 23: 97-120.
8. Van Everdingen, A. F. and Hurst, W. 1953. The Application of the LaPlace Transformation to Flow Problems in Reservoirs, *Trans. AIME* 198: 171-176.