

**DISEÑO DE SOFTWARE PARA COMPARACIÓN DE ALGORITMOS STEHFEST
E ISEGER PARA DETERMINAR LA INVERSA DE LAPLACE**

FABIÁN ANDRÉS LEGUÍZAMO POLANÍA

**UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
NEIVA, HUILA
2012**

**DISEÑO DE SOFTWARE PARA COMPARACIÓN DE ALGORITMOS STEHFEST
E ISEGER PARA DETERMINAR LA INVERSA DE LAPLACE**

FABIÁN ANDRÉS LEGUÍZAMO POLANÍA

**Proyecto de grado presentado para optar
al título de Ingeniero Electrónico**

**Director
FREDDY HUMBERTO ESCOBAR MACUALO
Ingeniero de Petróleos, Ph. D
Director del Grupo de Investigaciones en Pruebas de Pozos GIPP**

**UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
NEIVA, HUILA
2012**

Nota de aceptación:

Firma del presidente del jurado

Firma del primer jurado

Firma del segundo jurado

Neiva, 2 de Noviembre de 2012

A mi madre Amparo Polanía;
A mi tía Yolanda Polanía que es otra madre para mí;
A mis primos Sindy, Bryan y Jennifer que son como mis hermanos;
A mi Abuelita que desde el cielo me guía;
A todos mis amigos y compañeros de Universidad.
FABIÁN ANDRÉS LEGUÍZAMO POLANÍA

AGRADECIMIENTOS

A mi familia por toda su comprensión, apoyo incondicional, y porque depositaron toda su confianza en mi formación universitaria.

Al ingeniero Freddy Humberto Escobar Macualo por todo el tiempo dedicado y enseñanzas impartidas durante el desarrollo del proyecto.

Al ingeniero Mahmood Ahmadi cuya ayuda fue indispensable en el desarrollo de este proyecto.

A mi compañero y amigo Jonathan Tovar por su gran ayuda y asesoría en la programación de este proyecto.

A ECOPETROL-ICP y la Universidad Surcolombiana por la financiación de mi proyecto mediante el convenio 007/08.

A todos los profesores que durante la carrera me guiaron y compartieron todo su conocimiento en las diversas asignaturas.

A todos mis amigos del código y de la carrera con los cuales se vivieron miles de anécdotas y que hicieron de la estancia en la Universidad algo muy agradable y memorable.

CONTENIDO

	pág.
INTRODUCCIÓN	18
1. FUNDAMENTOS SOBRE PRUEBAS DE PRESIÓN	20
1.1. CONCEPTOS BÁSICOS	20
1.2. GENERALIDADES SOBRE LAS PRUEBAS DE PRESIÓN	25
1.2.1. Otras Pruebas	26
1.3. FACTORES ADIMENSIONALES	28
1.4. MODELOS DE YACIMIENTOS PETROLEROS UTILIZADOS	29
1.4.1. Pozo en yacimiento infinito con almacenamiento (C) y daño (S)	29
1.4.2. Pozo en yacimiento infinito sin almacenamiento (C) y daño (S)	30
1.4.3. Pozo en yacimiento infinito naturalmente fracturado con almacenamiento (C) y daño (S)	30
2. LA TRANSFORMADA DE LAPLACE	32
2.1. ORIGEN	32
2.2. DEFINICIÓN DE LA TRANSFORMADA	34
2.3. LA TRANSFORMADA INVERSA	34
2.4. USO DE LA TRANSFORMADA	36
2.5. ALGORITMO DE STEHFEST	38
2.6. ALGORITMO DE ISEGER	38
2.6.1. Introducción	38
2.6.2. Esquema del método	40

2.6.3. La Regla de Cuadratura de Gauss	41
2.6.4. Resumen y Análisis del método	45
2.6.4.1. Ejemplos de verificación	48
3. DESARROLLO DEL SOFTWARE	52
4. ANÁLISIS DE RESULTADOS	56
4.1. ANÁLISIS DEL YACIMIENTO HOMOGÉNEO INFINITO	61
4.2. ANÁLISIS DEL YACIMIENTO NATURALMENTE FRACTURADO	63
5. CONCLUSIONES	66
6. RECOMENDACIONES	67
NOMENCLATURA	68
BIBLIOGRAFÍA	70
ANEXOS	74

LISTA DE TABLAS

	pág.
Tabla 1. Parámetros obtenidos de pruebas de pozo	23
Tabla 2. Gráficas y regímenes de flujo encontrados en pruebas de pozo	24
Tabla 3. Resultados para las funciones analíticas de prueba	45
Tabla 4. Algunos pesos y nodos para la regla de cuadratura	74

LISTA DE FIGURAS

	pág.
Fig. 1. Esquema de la representación matemática de una prueba de presión	20
Fig. 2.a. Cartas de Identificación de yacimientos	21
Fig. 2.b. Resumen de reacciones de modelos de pozos - yacimientos	22
Fig. 3. Resumen de reacciones de modelos de pozos - yacimientos	22
Fig. 4. Representación esquemática de pruebas de restauración (abajo) y declinación o caída de presión (arriba)	25
Fig. 5. Prueba de inyección (izquierda) y prueba de abatimiento (derecha)	26
Fig. 6. Presión Adimensional para diferentes valores del radio adimensional	28
Fig. 7. Presión Adimensional para un pozo sin almacenamiento y daño en un yacimiento infinito	29
Fig. 8. Inversión numérica de una función escalón unitario por el algoritmo de Iseger. Efecto del número de puntos de inversión	47
Fig. 9. Efecto de nrp en la inversión numérica de una función por el algoritmo de Iseger	48
Fig. 10. Inversión numérica de una función escalón unitario; comparación de los algoritmos de Iseger y Stehfest	49
Fig. 11. Inversión numérica de una función con múltiples cambios tipo escalón; comparación de los algoritmos de Iseger y Stehfest	50
Fig. 12. Inversión numérica de una función derivable por partes; comparación de los algoritmos de Iseger y Stehfest	51
Fig. 13. Inversión de función escalón unitario por algoritmo de Iseger	56
Fig. 14. Inversión de función rampa por algoritmo de Iseger	57
Fig. 15. Inversión de función escalón unitario con retraso por algoritmo de Iseger	58
Fig. 16. Inversión de la función escalón múltiple por algoritmo de Iseger	59
Fig. 17. Inversión de la función seno por algoritmo de Iseger	60
Fig. 18. Resultados inversión del modelo homogéneo infinito	61
Fig. 19. Gráfica de ΔP contra el tiempo para el yacimiento homogéneo infinito	62
Fig. 20. Gráfica del error (%) contra el tiempo para el yacimiento homogéneo infinito	62
Fig. 21. Resultados inversión del modelo naturalmente fracturado	63
Fig. 22. Gráfica de ΔP contra el tiempo para el yacimiento naturalmente fracturado	64
Fig. 23. Gráfica del error (%) contra el tiempo para el yacimiento naturalmente fracturado	65
Fig. 24. Espacio de trabajo del programa	78
Fig. 25. Campos de gráficos de resultados	79
Fig. 26. Botón Iniciar	79

Fig. 27. Advertencia para escoger función	79
Fig. 28. Menú para escoger pozo o ecuación	80
Fig. 29. Campo función según la opción escogida	80
Fig. 30. Tipo de Gráfico	80
Fig. 31. Graficar Derivada y/o Usar Gráfica Logarítmica	81
Fig. 32. Parámetros	81
Fig. 33. Barra de progreso	82
Fig. 34. Resultados en escala logarítmica para modelo de pozo homogéneo infinito	82
Fig. 35. Resultados en escala lineal para modelo de pozo homogéneo infinito	83
Fig. 36. Gráfica de inversión de la función seno	83
Fig. 37. Gráfica de inversión de función multipaso	84
Fig. 38. Gráfica de inversión de función rampa	84
Fig. 39. Diagrama de flujo de algoritmo de Iseger	85

LISTA DE ANEXOS

	Pág.
ANEXO A. ALGUNOS PESOS Y NODOS PARA LA REGLA DE CUADRATURA	74
ANEXO B. LA TRANSFORMADA DE FOURIER	76
ANEXO C. CUADRATURA GAUSSIANA	77
ANEXO D. MANUAL DE USUARIO	78
ANEXO E. DIAGRAMA DE FLUJO DE ALGORITMO DE ISEGER	85

GLOSARIO

ALGORITMO: es una lista bien definida, ordenada y finita de operaciones que permite hallar la solución a un problema. Dado un estado inicial y una entrada, a través de pasos sucesivos y bien definidos se llega a un estado final, obteniendo una solución.

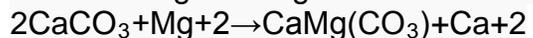
CANTIDAD ADIMENSIONAL: en análisis adimensional, se refiere a una cantidad sin ninguna unidad física, siendo un número puro. Las cantidades adimensionales con ampliamente usadas en los campos de matemáticas, física, ingeniería y economía, así como también en la vida cotidiana.

CONVOLUCIÓN: Es un operador matemático que transforma dos funciones f y g en una tercera función que en cierto sentido representa la magnitud en la que se superponen f y una versión trasladada e invertida de g .

CURVA TIPO: Es una curva estándar o una curva patrón, es una referencia para determinar el comportamiento de un muestra incógnita mediante interpolación. Esta referencia se obtiene determinando el comportamiento de dos componentes de valores conocidos.

DERIVADA: en geometría, la derivada de una función en un punto representa el valor de la pendiente de la recta tangente en dicho punto. La pendiente está dada por la tangente del ángulo que forma la recta tangente a la curva (función) con el eje de las abscisas, en ese punto. La derivada de una función mide el coeficiente de variación de dicha función.

DOLOMITIZACIÓN: Es el proceso mediante el cual la caliza se transforma en dolomita según la siguiente reacción:



(caliza) (Dolomita)

Algunas rocas carbonatas están constituidas solamente por calizas. Si el agua circulante a través del espacio poroso contiene suficientes cantidades de magnesio disuelto, el calcio en la roca puede intercambiarse por el magnesio en solución. Como el magnesio es considerablemente más pequeño que el calcio, la resultante dolomita tendrá una porosidad mayor, cuyo incremento oscila entre el 12-13.

ECUACIÓN DE DIFUSIÓN: Es una ecuación en derivadas parciales que describe fluctuaciones de densidad en un material que se difunde. También se usa para describir procesos que exhiben un comportamiento de difusión.

ESTOCÁSTICO: Sistemas cuyo comportamiento es intrínsecamente no determinístico. Un proceso estocástico es aquel cuyo comportamiento es no determinista, en la medida que el subsiguiente estado del sistema está determinado tanto por las acciones predecibles del proceso como por elementos aleatorios. No obstante, cualquier desarrollo temporal (sea determinístico o esencialmente probabilístico) que pueda ser analizable en términos de probabilidad merece ser denominado como un proceso estocástico.

FUNCIÓN CONTINUA: Es aquella para la cual, intuitivamente, para puntos cercanos del dominio se producen pequeñas variaciones en los valores de la función. Si la función no es continua, se dice que es discontinua. Generalmente una función continua es aquella cuya gráfica puede dibujarse sin levantar el lápiz del papel.

FUNCIÓN SINGULAR: Que contiene una o más singularidades en su dominio.

FLUJO EN MEDIO POROSO: Es un sólido con suficiente espacio abierto dentro o alrededor de las partículas para permitir el paso de un fluido.

JOSEPH FOURIER: Fue un matemático francés que desarrolló la teoría de las series de Fourier cuando estudiaba la ecuación de calor. Fue el primero en estudiar estas series sistemáticamente.

MICROSOFT VISUAL BASIC: constituye un IDE (entorno de desarrollo integrado o en inglés Integrated Development Environment) que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código (programa donde se escribe el código fuente), un depurador (programa que corrige errores en el código fuente para que pueda ser bien compilado), un compilador (programa que traduce el código fuente a lenguaje de máquina), y un constructor de interfaz gráfica o GUI (es una forma de programar en la que no es necesario escribir el código para la parte gráfica del programa, sino que se puede hacer de forma visual).

MODELO MATEMÁTICO: es uno de los tipos de modelos científicos, que emplea algún tipo de formulismo matemático para expresar relaciones, proposiciones sustantivas de hechos, variables, parámetros, entidades y relaciones entre variables y/o entidades u operaciones, para estudiar comportamientos de sistemas complejos ante situaciones difíciles de observar en la realidad.

MODELO ESTOCÁSTICO: Modelo matemático en el que la ley de probabilidad que da la evolución de un sistema depende del tiempo.

PERMEABILIDAD: Es la capacidad que tiene un material de permitirle a un líquido que lo atraviese sin alterar su estructura interna. Un material es permeable si deja pasar a través de él una cantidad apreciable de fluido en un tiempo dado, e impermeable si la cantidad de fluido es despreciable.

PIERRE-SIMON LAPLACE: Astrónomo, físico y matemático francés que inventó y desarrolló la transformada de Laplace y la ecuación de Laplace.

PLANO COMPLEJO: Es una forma de visualizar el espacio de los números complejos. Puede entenderse como un plano cartesiano modificado, en el que la parte real está representada en el eje x y la parte imaginaria en el eje y . El eje x también recibe el nombre de eje real y el eje y el nombre de eje imaginario.

PRESIÓN: es una magnitud física que mide la fuerza por unidad de superficie, y sirve para caracterizar como se aplica una determinada fuerza resultante sobre una superficie.

En el Sistema Internacional de Unidades (SI) la presión se mide en una unidad derivada que se denomina pascal (Pa) que es equivalente a una fuerza total de un newton actuando uniformemente en un metro cuadrado.

PRESIÓN TRANSIENTE: Presión transitoria o presión instantánea.

PROGRAMACIÓN LINEAL: es un procedimiento o algoritmo matemático mediante el cual se resuelve un problema indeterminado, formulado a través de ecuaciones lineales, optimizando la función objetivo, también lineal.

PROGRAMACIÓN NO LINEAL: En matemáticas, es el proceso de resolución de un sistema de igualdades y desigualdades sujetas a un conjunto de restricciones sobre un conjunto de variables reales desconocidas, con una función objetivo a maximizar, cuando alguna de las restricciones o la función objetivo no son lineales.

PRUEBAS DST: Una prueba DST (Drillstem Test) es una prueba de presión corta que se efectúa durante la perforación utilizando la tubería de perforación. Está formada por pruebas de inclinación y caída de presión consecutivas. Para correr un DST, una herramienta especial se coloca en la sarta de perforación y se baja a la zona a probar. La herramienta aísla la formación de la columna de lodo en el anular y permite que los fluidos de la formación fluyan a la sarta de perforación mientras se registra continuamente la presión.

PSI: una unidad de presión cuyo valor equivale a 1 libra por pulgada cuadrada.

RECTA TANGENTE: en geometría, es aquella que sólo tiene un punto en común con una curva, es decir la *toca* en un solo punto, que se llama punto de tangencia. La recta tangente indica la pendiente de la curva en el punto de tangencia.

SERIES DE FOURIER: Es una serie infinita que converge puntualmente a una función periódica y continua a trozos (o por partes). Constituyen la herramienta matemática básica del análisis de Fourier empleado para analizar funciones periódicas a través de la descomposición de dicha función en una suma infinita de funciones senoidales mucho más simples (como combinación de senos y cosenos con frecuencias enteras). Es usada en muchas ramas de la ingeniería, además de ser una herramienta sumamente útil en la teoría matemática abstracta.

SINGULARIDAD (MATEMÁTICA): La singularidad de una función es el comportamiento extraño e inesperado que presentan algunas funciones (dentro de la amplia variedad de funciones matemáticas existentes) cuando se le asignan determinados valores a la(s) variable(s) independiente(s).

TERCERA LEY DE NEWTON: Es completamente original de Newton y hace de las leyes de la mecánica un conjunto lógico y completo. Dice que por cada fuerza que actúa sobre un cuerpo (empuje), este realiza una fuerza de igual intensidad, pero de sentido contrario sobre el cuerpo que la produjo, es decir que las fuerzas, situadas sobre la misma recta, siempre se presentan en pares de igual magnitud y de dirección, pero con sentido opuesto.

TRANSIENTE: Un transiente es una señal o forma de onda que empieza en una amplitud cero. Un ejemplo es el sonido de un disparo de un rifle, o la

vibración de un golpe de un martillo. Cuando se hace el análisis de espectro a transientes, generalmente no generan series de armónicos, pero generan un espectro continuo en el que la energía está distribuida sobre el rango de frecuencias. Cuando se analiza transientes con un analizador TRF, se tiene que cuidar que el transiente está incluido en la grabación en tiempo del analizador y que se use una ventana rectangular en lugar de una ventana de Hanning.

TRANSFORMADA DE MELLIN: Es una transformada integral que puede ser considerada como una versión multiplicativa de la transformada bilateral de Laplace. Está íntimamente relacionada con la teoría de las series de Dirichlet, y es usada habitualmente en teoría de números y la teoría de series asintóticas; también está fuertemente relacionada con la transformada de Laplace, la transformada de Fourier y la teoría de la función gamma, y forma parte de las funciones especiales.

YACIMIENTO: es una formación en la que está presente una concentración estadísticamente anómala de minerales (depósitos minerales) presentes en la corteza terrestre o litosfera. Algunos procesos geológicos internos y externos pueden producir localmente concentraciones económicas de materiales como menas explotables de metales, carbón o hidrocarburos.

RESUMEN

Este proyecto se centra en obtener la transformada inversa de Laplace y comparar dos métodos (Stehfest e Iseger) utilizados para este fin, práctica común utilizada en la industria petrolera en los análisis de pruebas de presión en pozos. La transformada de Laplace proporciona ventajas en la solución de muchos problemas de análisis de transientes de presión. Usualmente estas aplicaciones llevan a una solución que necesita ser numéricamente invertida al dominio del tiempo real.

El algoritmo presentado por Stehfest en 1970 es la herramienta más común en la Ingeniería de Petróleos para la inversión numérica de transformadas de Laplace y se utiliza como punto de comparación en este trabajo; además, es el único algoritmo empleado en software comerciales usados para interpretar pruebas de pozos. Este algoritmo, sin embargo, solo es aplicable a funciones continuas y esta limitación imposibilita su uso para una amplia variedad de problemas de interés práctico. Otros algoritmos también han sido usados, pero con limitado éxito o popularidad. Un algoritmo reciente presentado por Iseger en 2006 soluciona la restricción de continuidad y proporciona oportunidades para muchas aplicaciones prácticas.

El desarrollo del software para realizar la transformada inversa de Laplace se hizo empleando el lenguaje de programación Visual Basic .Net en el cual se realizó la interfaz amigable para escoger los archivos necesarios con los datos requeridos así como para ingresar la función en dominio Laplace que se va a invertir al dominio del tiempo real para el caso del algoritmo de Iseger, así mismo se graficará por ambos métodos para comparar resultados y así decidir que método es más eficaz y preciso en diversos casos.

Cabe destacar las características útiles del algoritmo de Iseger en la inversión de funciones continuas también como en la de funciones singulares y discontinuas que origina la solución de problemas de análisis de transientes de presión. Las aplicaciones más destacadas están en los problemas que requieren el uso de funciones continuas por partes y funciones diferenciables por partes, tales como el uso de datos tabulados en el dominio de la transformada de Laplace, algoritmos de deconvolución, y soluciones que incluyen variaciones tipo paso comúnmente utilizadas en pruebas de presión.

En los resultados obtenidos se pudo comprobar el enorme éxito que tiene el algoritmo de Iseger en la inversión de funciones discontinuas así como una mayor estabilidad, pero también se pudo concluir que el algoritmo de Iseger es muy demorado cuando se incrementan el tiempo y el número de puntos de inversión.

ABSTRACT

This project focuses on obtaining the inverse Laplace transform and compare two methods (Stehfest and Iseger) used for this purpose, a common practice used in the petroleum industry in the analysis of pressure tests in wells. Laplace transformation provides advantages in the solution of many pressure-transient analysis problems. Usually, these applications lead to a solution that needs to be inverted numerically to the real-time domain.

The algorithm presented by Stehfest in 1970 is the most common tool in petroleum engineering for the numerical inversion of Laplace transforms and it's used as a comparison in this work; besides, it's the only algorithm used in commercial software used to interpret well testing. This algorithm, however, is only applicable to continuous functions and this limitation precludes its use for a wide variety of problems of practical interest. Other algorithms have also been used, but with limited success or popularity. A recent algorithm presented by Iseger in 2006 removes the restriction of continuity and provides opportunities for many practical applications.

The software development to do the inversion of Laplace transform was made using Visual Basic .Net programming language that was used to do the friendly interface to choose the files that are needed as well as to introduce the function in Laplace domain that is going to be inverted to the real-time domain for the case of Iseger's algorithm, so it for both methods there are graphics to compare the results and in that way choose the more efficient and accurate method in several cases.

It's important to mention the useful features of the Iseger's algorithm in the inversion of continuous as well as singular and discontinuous functions that arise in the solution of pressure-transient analysis problems. The most remarkable applications are in the problems that require the use of piecewise continuous and piecewise differentiable functions, such as the use of tabulated data in the Laplace transform domain, deconvolution algorithms, and solutions that include step-rate changes as in the mini-DST tests.

In the results was noted that Iseger's algorithm has a huge success in the inversion of discontinuous functions and a greater stability, but also was concluded that the Iseger's algorithm takes too long when the time and the number of reversal points are increased.

INTRODUCCIÓN

Desde el trabajo seminal de van Everdingen y Hurst [33], 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 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 [36] que es una de las ventajas que se muestra en este proyecto.

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 nuestra industria de software está aún incipiente, por tanto es una muy buena oportunidad lograr incursionar en este campo, en donde como ingenieros aprenderíamos de una temática inexplorada por nosotros y mostrarnos 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.

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. Esto

se consigue construyendo un modelo matemático, el cual genera una respuesta de salida similar al del sistema real.

La parte fundamental de este trabajo de grado 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.

1. FUNDAMENTOS SOBRE PRUEBAS DE PRESIÓN

1.1. CONCEPTOS BÁSICOS

Las pruebas de presión pueden entenderse por aplicación de la tercera ley de Newton, como se ilustra en la figura 1 [13].

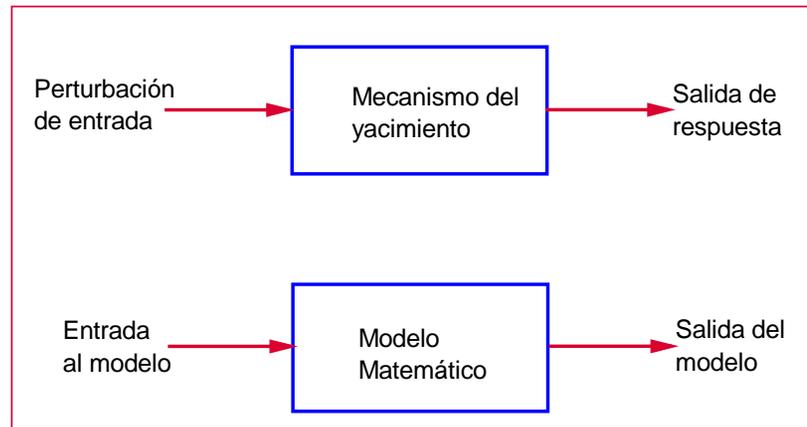


Fig. 1. Esquema de la representación matemática de una prueba de presión [13]

Básicamente los objetivos del análisis de las pruebas de presión son [13]:

- Evaluación del Yacimiento: Entrega, propiedades, tamaño, permeabilidad por espesor (útil para espaciamiento y estimulación), presión inicial (energía y pronóstico), límites (tamaño y determinación de existencia de un acuífero).
- Administración del yacimiento.
- Descripción del yacimiento.

Las pruebas DST y restauración de presión se usan principalmente en producción primaria y exploración. Las pruebas múltiples se usan más a menudo durante proyectos de recuperación secundaria y las pruebas multicapa y de permeabilidad vertical se usan en pozos productores/inyectores. Las pruebas de declinación, de restauración de interferencia y de pulso se usan en todas las fases de producción. Las pruebas multitasa, de inyección, de interferencia y pulso se usan en las etapas primaria y secundaria.

El análisis de pruebas de presión tiene una variedad de aplicaciones durante la vida de un yacimiento. Las pruebas DST y de restauración de presión de pozos únicos se usan principalmente durante producción primaria y exploración, mientras que las pruebas múltiples se usan más a menudo durante proyectos de recuperación secundaria. Las pruebas multicapa y de permeabilidad vertical también se corren en pozos productores/inyectores. Pruebas de declinación, de restauración, de interferencia y de pulso se utilizan en todas las fases de producción. Las pruebas multitasas, de inyección, de interferencia y de pulso se

usan en las etapas primaria y secundaria. La tabla 1 resume los parámetros que pueden obtenerse del análisis de pruebas de presión. Los ingenieros de petróleo deberían tener en cuenta el estado del arte de la interpretación de pruebas de presión, herramientas de adquisición de datos, métodos de interpretación y otros factores que afectan la calidad de los resultados obtenidos del análisis de pruebas de presión.

Una vez los datos han sido obtenidos y revisados, el análisis de presiones comprende dos pasos: (1) El modelo del yacimiento e identificación de los diferentes regímenes de flujo encontrados durante la prueba, y (2) la estimación de parámetros. Entre ellos tenemos: gráficos log-log de presión y derivada de presión vs tiempo de transiente (herramienta de diagnóstico), gráfico semilog de presión vs tiempo, gráfico Cartesiano de los mismos parámetros, etc. La tabla 2 proporciona diferentes gráficos y regímenes de flujo que normalmente se encuentran en cada prueba y las Figs. 2 a 3 ilustran diferentes condiciones de yacimiento y características de flujo encontrados en una prueba de presión.

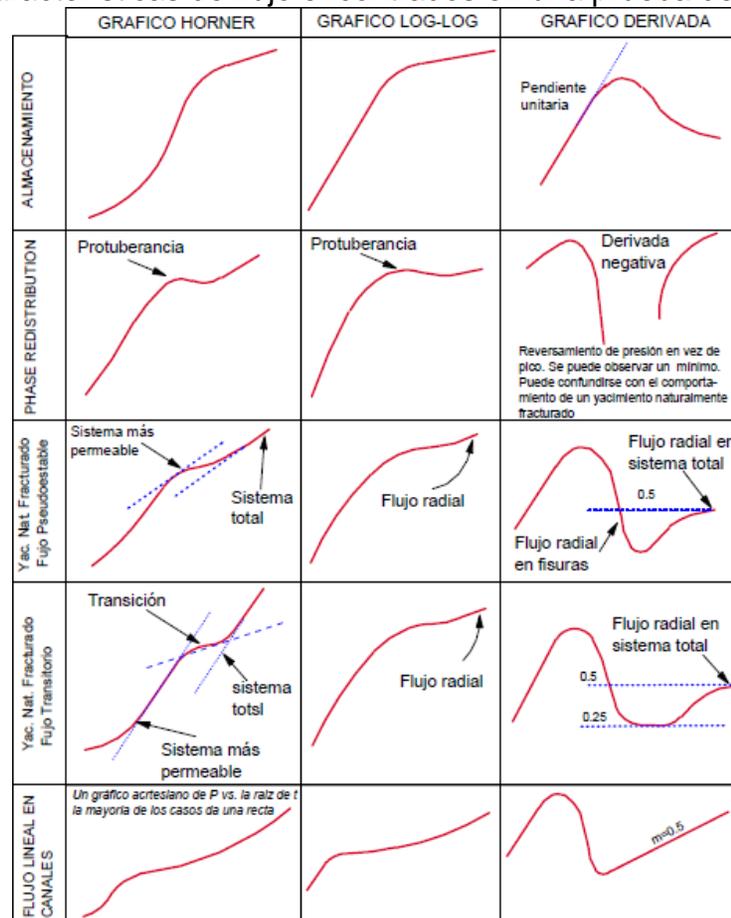


Fig. 2.a. Cartas de Identificación de yacimientos [13]

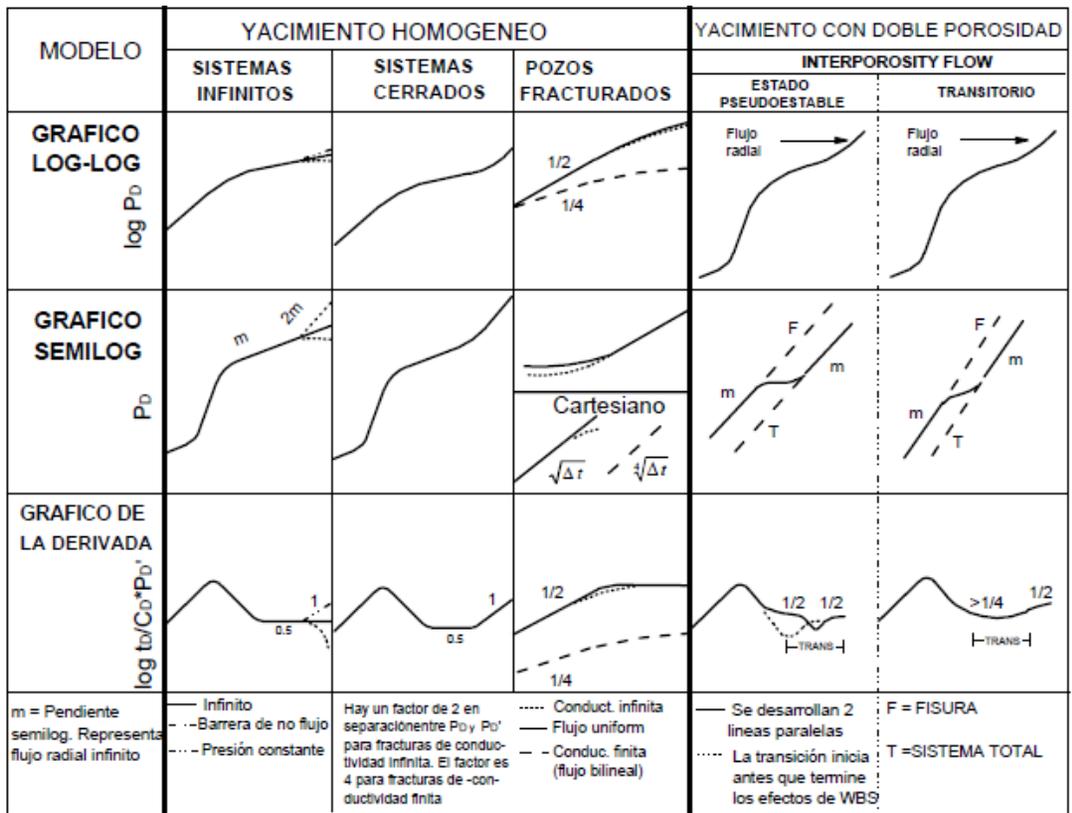


Fig. 2.b. Resumen de reacciones de modelos de pozos – yacimientos [13]

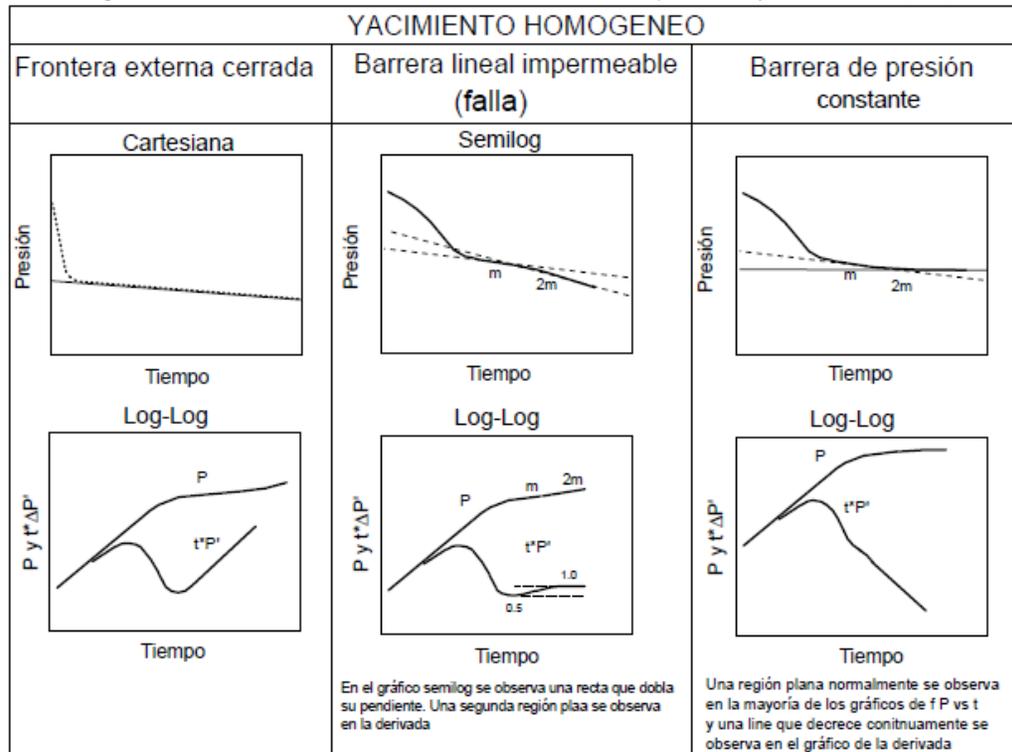


Fig. 3. Resumen de reacciones de modelos de pozos – yacimientos [13]

Tipo de Prueba	Parámetro Obtenido
DST	Comportamiento del yacimiento Permeabilidad Daño Longitud de fractura Presión del yacimiento Límites del yacimiento Fronteras
Prueba de formación múltiple repetida	Perfil de Presión
Prueba de declinación de presión	Comportamiento del yacimiento Permeabilidad Daño Longitud de fractura Límites del yacimiento Fronteras
Prueba de restauración de presión	Comportamiento del yacimiento Permeabilidad Daño Longitud de fractura Presión del yacimiento Fronteras
Prueba de paso de rata	Presión de rotura de formación Permeabilidad Daño
Prueba Falloff	Movilidad en varios bancos Daño Presión del yacimiento Longitud de fractura Ubicación del frente Fronteras
Prueba de pulso e interferencia	Comunicación entre pozos Comportamiento del tipo de yacimiento Porosidad Permeabilidad interpozos Permeabilidad vertical
Pruebas de yacimientos con capas	Propiedades de capas individuales Permeabilidad horizontal Permeabilidad vertical Daño Presión de capa promedio Fronteras externas

Tabla 1. Parámetros obtenidos de pruebas de pozo [13]

Régimen de flujo	Gráficas				
	Cartesiana	$\sqrt{\Delta t}$	$\sqrt[3]{\Delta t}$	Log-log	Semilog
Almacenamiento	Línea recta Pendiente $\rightarrow C$ Intercepto $\rightarrow \Delta t_c$ ΔP_c			Pendiente unitaria en Δp y p' Δp y p' coincide	s Positivo s Negativo
Flujo Lineal		Línea recta Pendiente $\rightarrow x_f$ Intercepto \rightarrow Daño de fractura		Pendiente = 1/2 en ΔP y P' si $s=0$ Pendiente = 1/2 en Δp y P' si $s=0$ a medio nivel de ΔP Pendiente = 1/2 después de almacenamiento indica un canal del yacimiento	
Flujo Bilineal			Línea recta Pendiente $\rightarrow C_{fd}$	Pendiente = 1/4 P' a 1/4 de nivel de ΔP	
Primer IARF (alta- k capas, fracturas)	Disminución de pendiente			P' horizontal a $P'_{D'}=1/2$	Línea recta Pendiente $\rightarrow kh$ $\Delta P_{D'} \rightarrow s$
Transición	Más disminución de pendiente			$\Delta P = \lambda e^{-\alpha t}$ $P'_{D'}=1/4$ (transición) $P'_{D'} < 1/4$ (estado pseudoestable)	Línea recta Pendiente = 1/2 (transición) Pendiente = 0 (estado pseudoestable)
Segundo IARF	Pendiente similar al primer IARF			P' horizontal a $P'_{D'}=1/2$	Línea recta Pendiente $\rightarrow kh, P^*$ $\Delta P_{D'} \rightarrow s$
Frontera sencilla de no flujo				P' horizontal a $P'_{D'}=1$	Línea recta Pendiente = 2m Intersección con IARF \rightarrow distancia a frontera
Fronteras externas de no flujo (solo declinación)	Línea recta pendiente $\rightarrow \phi h$ $P_{int} \rightarrow C_A$			Pendiente unitaria para ΔP y P' ΔP y P' coincide	Incremento de pendiente

IARF = flujo radial de acción infinita

Tabla 2. Gráficas y regímenes de flujo encontrados en pruebas de pozo [13]

En general, el análisis de presiones es una herramienta excelente para describir y definir el modelo de un yacimiento cuando se maneja un campo hidrocarburífero. Los regímenes de flujo son una función directa de las características del sistema pozo/yacimiento, i.e., una fractura sencilla que intercepta el pozo puede identificarse mediante la detección de un flujo lineal. Sin embargo, siempre que exista flujo lineal, no necesariamente implica la presencia de una fractura.

La interpretación de pruebas de presión es el método primario para determinar permeabilidad, factor de daño, presión de yacimiento, longitud y conductividad de fractura y heterogeneidad del yacimiento. Además, es el único método más rápido y más barato para estimar variables dependientes del tiempo como el factor de daño y la permeabilidad en yacimientos sensibles al esfuerzo.

El período de comportamiento infinito ocurre después del fin del almacenamiento y antes de la influencia de los límites del yacimiento. Puesto que los límites no afectan los datos durante este período, el comportamiento de presión es idéntico al comportamiento de un yacimiento infinito. El flujo radial puede reconocerse por una estabilización aparente del valor de la derivada.

El análisis de presiones puede utilizarse para determinar permeabilidad, daño, presión promedio, longitud media de una fractura hidráulica, dirección de una fractura, conductividad de la fractura, entre otros. Obtenidos los datos siguen dos pasos (1) Definir el modelo del yacimiento e identificación de los regímenes de flujo y (2) Estimación de parámetros.

1.2. GENERALIDADES SOBRE LAS PRUEBAS DE PRESIÓN

Declinación de presión (ver. Fig. 4). Se le conoce como prueba de flujo. Luego de que el pozo ha sido cerrado por un tiempo suficientemente largo para alcanzar estabilización, el pozo se coloca en producción, a caudal constante, mientras se registra la presión de fondo contra el tiempo. Su principal desventaja es que es difícil mantener el caudal constante.

Restauración de presión (ver. Fig. 4). Se le conoce como prueba de cierre. En esta prueba el pozo se cierra mientras se registra la presión estática del fondo del pozo en función del tiempo. Esta prueba se cataloga como una prueba multirata con dos caudales (cero y otro diferente de cero) y permite obtener la presión promedio del yacimiento. Su principal desventaja es económica ya que el cierre ocasiona pérdida de producción.

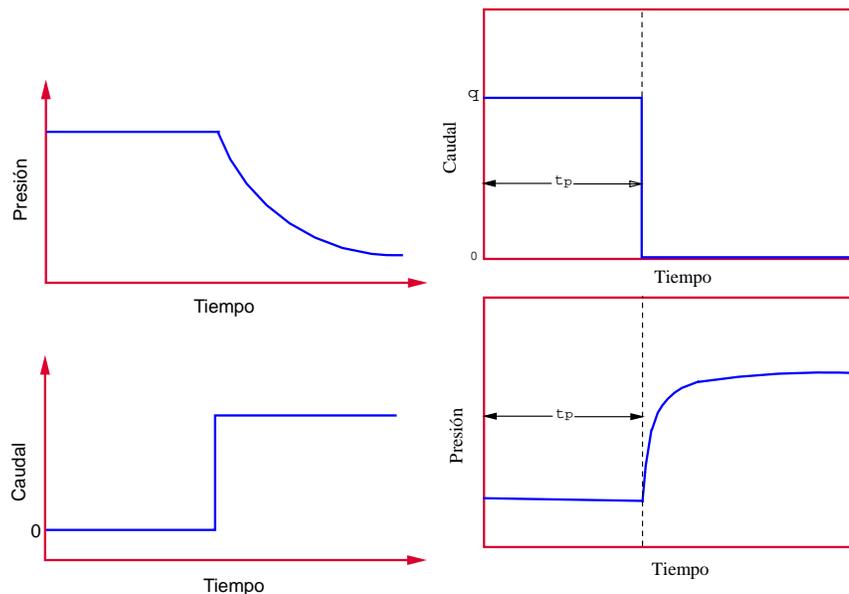


Fig. 4. Representación esquemática de pruebas de restauración (abajo) y declinación o caída de presión (arriba) [13]

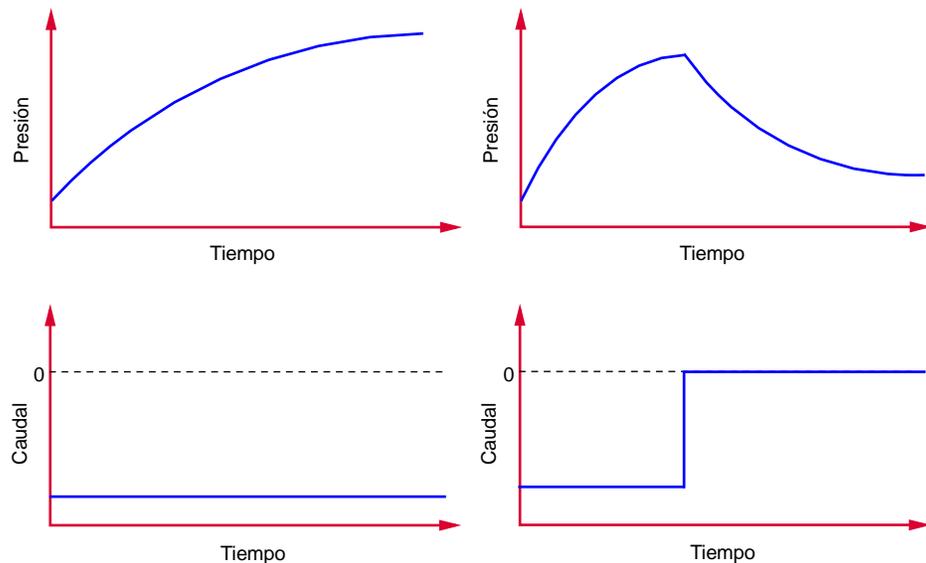


Fig. 5. Prueba de inyección (izquierda) y prueba de abatimiento (derecha) [13]

Inyección. Ver (Fig. 5). Es una prueba similar a la prueba de declinación de presión, pero en lugar de producir fluidos se inyectan fluidos, normalmente agua.

Abatimiento. (Ver Fig. 5). Considera una declinación de presión inmediatamente después de la inyección. Idéntico a una prueba de restauración.

1.2.1 Otras pruebas:

Interferencia y/o Múltiples. Involucran más de un pozo y su propósito es definir conectividad y hallar permeabilidades direccionales.

DST. Esta prueba se usa durante o inmediatamente después de la perforación del pozo y consiste de pruebas de cierre o flujo, cortas y seguidas. Su propósito es establecer el potencial del pozo, aunque el factor de daño estimado no es muy representativo porque puede ocurrir una limpieza del mismo pozo durante la primera etapa productiva del mismo.

El uso masivo de computadoras personales, ha facilitado la aplicación del método de la derivada de presión y la simulación con curvas tipo para el análisis de las pruebas de presión transitoria en la industria petrolera. El concepto de la derivada de presión con respecto al tiempo en el análisis de pruebas de presión lo introdujeron Bourdet et. al. en el año 1983. Esta técnica consiste en graficar la velocidad de cambio de la presión (la derivada de la presión) con respecto a una función de tiempo versus el tiempo transcurrido, en una escala doble logarítmica. El gráfico de la derivada ofrece la ventaja de proporcionar una línea de pendiente cero (estabilización horizontal) para los regímenes de flujo cilíndrico.

La derivada de presión es sumamente sensible a las variaciones de la presión provocada por los cambios ocurridos en los regímenes de flujo que se producen en el yacimiento; aun cuando se producen a los tiempos tardíos. Debido a ello, la buena capacidad del instrumento utilizado para adquirir los datos de presión transitoria, adquiere suma importancia. Si el sensor o medidor introduce cambios debidos a las características del instrumento y no provocados por variaciones reales de la presión, se obtendrá una interpretación incorrecta y, en consecuencia se puede llegar a cometer graves errores en el cálculo de los parámetros del pozo y del yacimiento.

Las condiciones del fondo del pozo constituyen un gran desafío para la operación de un sensor de presión: puede verse afectado por golpes producidos durante las carreras de entrada y salida del pozo; por las elevadas presiones y temperaturas que pueden sufrir cambios muy bruscos debido a los procesos de producción, inyección o estimulación; por la presencia de sustancias químicas hostiles como ácido sulfhídrico (H_2S), Dióxido de Carbono (CO_2) o ácidos de estimulación; y por las vibraciones provocadas por el flujo mismo de los fluidos, o los golpes causados por los cañones de perforación bajados con la tubería de producción.

Cuando se utiliza una sonda para adquirir datos durante las pruebas de presión transitoria, las características fundamentales del instrumento son la deriva, el ruido, la resolución y la respuesta del instrumento frente a una variación de temperatura. En efecto la sonda debe estar diseñada para poder mantener el comportamiento metrológico frente a estas condiciones difíciles. A continuación se hace una breve descripción de estas características:

- Precisión: se refiere a la variación máxima esperada del valor medido con respecto al valor real.
- Repetitividad: la variación máxima esperada cuando se realiza varias veces la misma medición.
- Resolución: el menor cambio del valor real que la sonda es capaz de detectar. Respuesta a la presión transitoria durante una variación de temperatura: la tendencia de la sonda a variar la medida de presión cuando se produce algún cambio en la temperatura ambiente del sensor. Los cambios en la temperatura son de particular importancia cuando se realizan pruebas en pozos de gas.
- Histéresis: si la presión real aplicada sobre la sonda aumenta gradualmente hasta alcanzar la presión máxima admitida, y luego se vuelve a reducir a cero, la histéresis es la diferencia máxima entre la medida en el ciclo creciente y la medida en el ciclo decreciente para un valor real cualquiera de la presión.
- Derivada: el cambio de la presión medida por la sonda mediante un periodo de tiempo dado cuando se aplica una presión constante de referencia.

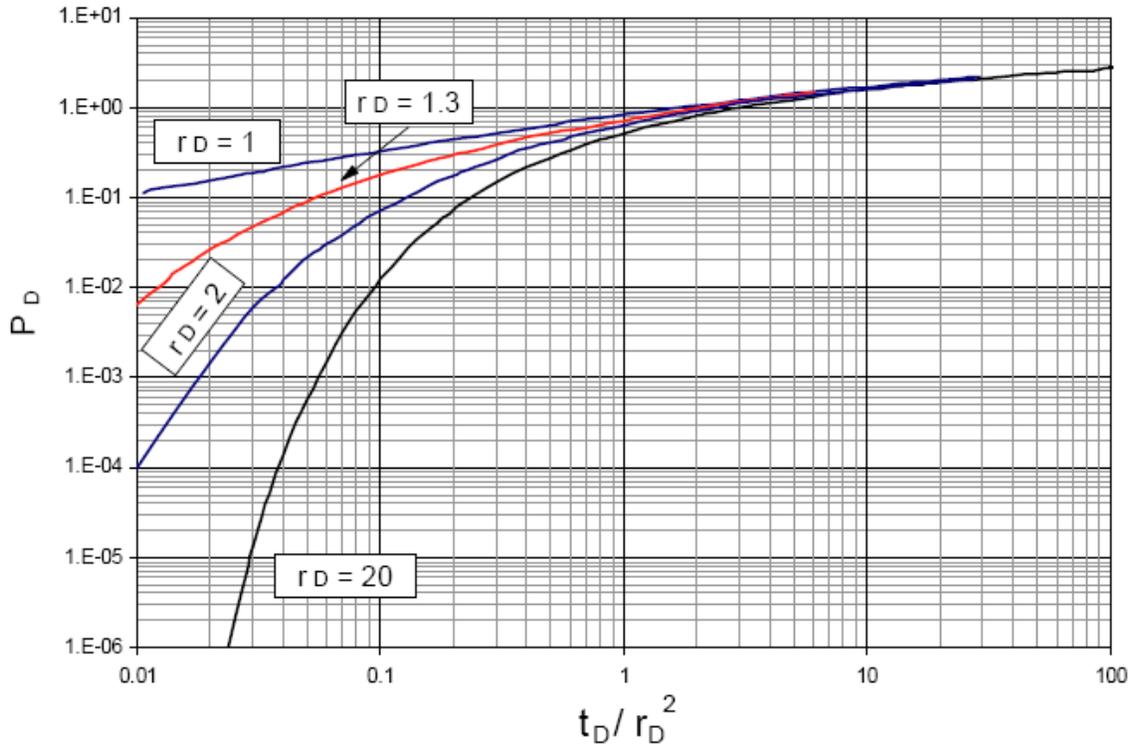


Fig. 7. Presión Adimensional para un pozo sin almacenamiento y daño en un yacimiento infinito [13]

1.4. MODELOS DE YACIMIENTOS PETROLEROS UTILIZADOS

1.4.1 Pozo en yacimiento infinito con almacenamiento (C) y daño (S)

Los yacimientos son cuerpos originados durante largos procesos geológicos donde actúan diversos ambientes depositacionales tanto en tiempo como en espacio. Como resultado de subsecuentes reorganizaciones físicas y químicas, como compactación, solución, dolomitización y cementación, las características del yacimiento son variables. El modelo matemático que gobierna el comportamiento de la presión es:

$$\bar{P}_D = \frac{K_0(\sqrt{s}) + S\sqrt{s}K_1(\sqrt{s})}{s \left\{ \sqrt{s}K_1(\sqrt{s}) + sC_D \left[K_0(\sqrt{s}) + S\sqrt{s}K_1(\sqrt{s}) \right] \right\}} \quad (1)$$

Donde \bar{P}_D es la presión adimensional en el espacio Laplaciano, s es el parámetro Laplace, S es el factor de daño, C_D es el almacenamiento del pozo adimensional y K_0 y K_1 son funciones de Bessel de orden 0 y 1 respectivamente.

Las características geológicas de las rocas que conforman los yacimientos son extremadamente variables y presentan variaciones laterales y verticales. Estas variaciones pueden ocurrir a grandes distancias, o incluso a distancias muy cercanas.

Si el yacimiento es homogéneo, la medición de propiedades en cualquier ubicación describe completamente la distribución de dicha propiedad en todo el yacimiento. La descripción de yacimientos homogéneos es bastante simple.

1.4.2 Pozo en yacimiento infinito sin almacenamiento (C) y daño (S)

Los yacimientos infinitos sin almacenamiento y daño, presentan un comportamiento descrito por las siguientes ecuaciones:

$$P_D = -\frac{1}{2} E_i(-x) \quad (2)$$

siendo $E_i(-x)$ la función exponencial definida como

$$E_i(-x) = \int_x^\infty \frac{e^{-x}}{x} \frac{dx}{du} \quad (3)$$

que al ser reemplazada en (2) resulta

$$P_D = -\frac{1}{2} \int_x^\infty \frac{e^{-x}}{x} \frac{dx}{du} \quad (4)$$

Donde:

$$x = \frac{948\phi\mu c_i r^2}{k * t} \quad (5)$$

Al desarrollar la integral (4), y reemplazando previamente el valor de x presentado en (5), se obtiene el valor de la presión adimensional en cualquier tiempo t . Aplicando la regla de Leibnitz y tomando derivada resulta finalmente:

$$t_D * P_D' = \frac{1}{2} e^{-x} \quad (6)$$

1.4.3 Pozo en yacimiento infinito naturalmente fracturado con almacenamiento (C) y daño (S)

Los yacimientos naturalmente fracturados son aquellos yacimientos de hidrocarburos cuya producción está influenciada por la presencia de una red de

fracturas. Para modelar el comportamiento de la porosidad de estos yacimientos se plantea que “una roca porosa con un sistema de fracturas altamente desarrollado puede ser representado como la superposición de dos medios porosos de diferentes tamaños”, en otras palabras el sistema está conformado por la porosidad ínter granular y la porosidad formada por las fracturas. En el caso de los carbonatos, puede presentarse un sistema poroso con vacuolas o huecos, a esto se le llama porosidad vacular.

Las fracturas, son aquellas discontinuidades planas en el material que compone la roca, y pueden ser microscópicas o macroscópicas. Las fracturas aparecen cuando los esfuerzos de confinamiento a los que ha estado sometida la roca, son superiores a los esfuerzos de ruptura.

La condición de almacenamiento del pozo es [25]:

$$C_D \frac{\partial P_{wFD}}{\partial t_D} - \left(\frac{\partial P_{FD}}{\partial r_D} \right)_{r_D} = 1 \quad (7)$$

$$C_D = \frac{C}{2\pi h(\phi_f c_f + \phi_m c_m) r_w^2} \quad (8)$$

Y la condición del efecto de daño es:

$$P_{wFD} = \left(p_{FD} - S \frac{\partial P_{FD}}{\partial r_D} \right)_{r_D=1} \quad (9)$$

Donde S es el factor daño.

$$f(s) = \frac{\varpi(1-\varpi)s + \lambda}{(1-\varpi)s + \lambda} \quad (10)$$

$$\bar{P}_D(s) = \frac{s\sqrt{sf(s)} + S\sqrt{sf(s)}K_1(\sqrt{sf(s)})}{s\sqrt{sf(s)}K_1(\sqrt{sf(s)}) + C_D s K_0(\sqrt{sf(s)}) + S\sqrt{sf(s)}K_1(\sqrt{sf(s)})} \quad (11)$$

Se puede afirmar que los yacimientos fracturados fueron inicialmente yacimientos matriciales, y que de alguna manera sus propiedades físicas fueron cambiadas, deformadas o alteradas en el proceso de diagénesis física o durante el proceso de depositación. Por ende solo se toma como yacimientos fracturados a aquellos cuyas fracturas gobiernan a los patrones de productividad del yacimiento.

2. LA TRANSFORMADA DE LAPLACE

2.1. ORIGEN

La transformada de Laplace debe su nombre al matemático francés Pierre-Simon Laplace, que la presentó dentro de su teoría de la probabilidad. Antes de él, en 1744, Leonhard Euler había investigado un conjunto de integrales de la forma [27]:

$$z = \int X(x)e^{ax} dx \quad (12)$$

$$z = \int X(x)e^A dx \quad (13)$$

Presentó estas integrales como soluciones de ecuaciones diferenciales, pero no profundizó en ellas y luego abandonó la investigación. Joseph Louis Lagrange, admirador de Euler, investigó estas integrales y las ligó a la teoría de la probabilidad, mostrando sus resultados de la forma:

$$\int X(x)e^{-ax} a^x dx \quad (14)$$

Esta expresión es considerada por algunos historiadores como auténticas transformadas de Laplace. Este tipo de integrales llamaron la atención del señor Laplace que en 1782 y siguiendo la idea de Euler, trató de emplear estas integrales para solucionar ecuaciones diferenciales. En el año de 1785 avanzó en su empeño, pero cambió el enfoque del problema, esta vez no usó las integrales como soluciones sino que las aplicó a las ecuaciones dando lugar a las transformadas de Laplace tal y como se conocen hoy en día. La integral usada fue de la forma [27]:

$$\int x^s \phi(s) dx \quad (15)$$

Esta integral es análoga a la transformada de Mellin, con la que se transformó una ecuación diferencial a una ecuación algebraica la cual se solucionó. Planteó también algunas de las propiedades más importantes de la transformada y de alguna forma reconoció que el método de Joseph Fourier para resolver por medio de series de Fourier la ecuación de difusión podría relacionarse con su transformada integral para un espacio finito con soluciones periódicas [27].

A pesar de este logro, las transformadas de Laplace cayeron en el olvido muy pronto, ya que no fueron presentadas como ahora, aplicadas a la física y a la ingeniería, sino que fueron presentadas en el campo de la probabilidad como objetos matemáticos meramente teóricos.

En la segunda mitad del XIX, un ingeniero inglés llamado Oliver Heaviside trataba de resolver ecuaciones diferenciales relacionadas con la teoría de vibraciones, en su empeño descubrió que los operadores diferenciales podían ser tratados analíticamente como variables algebraicas; de este modo se empezó la aplicación moderna de las transformadas de Laplace y toda su teoría subyacente.

De acuerdo con el cálculo operacional, si se tiene la ecuación diferencial de la forma:

$$(D - a)y = f(t) \quad (16)$$

donde D es el operador diferencial, es decir, $D = d / dx$, entonces la solución general a dicha ecuación es de la forma:

$$y = e^{at} \int e^{-at} f(t) dt + c_1 e^{at} \quad (17)$$

Heaviside observó que si se trataba al operador D como una variable algebraica, era posible alcanzar igualmente la solución de toda ecuación pareja a la de arriba.

La utilidad demostrada de los resultados de Heaviside para resolver ecuaciones de física e ingeniería hicieron que pronto se extendieran, aun así algunos matemáticos puristas rechazaron argumentando que los resultados no podían surgir de tal forma ya que consideran el trabajo de Heaviside como formal y poco riguroso. De todos modos, el éxito del método hizo que los ingenieros y físicos adoptaran este método y que cierto número de matemáticos trataran de justificar el método de manera rigurosa.

Después de varias décadas de intentos se llegó a la conclusión que la transformada descubierta por Laplace hacía un siglo, ofrecía tanto fundamento teórico al método de Heaviside, así como otra alternativa más sistemática a tales métodos.

La transformada de Laplace se convirtió en una herramienta común de la teoría de vibraciones y de la teoría de circuitos a principios del siglo XX, antes, no había sido aplicada en estos campos con éxito. La transformada es adecuada para resolver sistemas de ecuaciones diferenciales lineales con condiciones iniciales en el origen. Entre sus ventajas más importantes está que la integración y derivación se convierten en multiplicación y división, por lo tanto las ecuaciones diferenciales se transforman en ecuaciones polinómicas que son mucho más fáciles de resolver [27].

2.2. DEFINICIÓN DE LA TRANSFORMADA

La transformada de Laplace es una operación lineal que asigna una $f(s)$ funcional, a una función $f(t)$ definida para todo $t > 0$, por la regla

$$\mathcal{L}\{f(t)\} = \bar{f}(s) = \int_{0+}^{\infty} e^{-st} f(t) dt \quad (18)$$

Un conjunto de condiciones de la función $f(t)$ y el parámetro s de la transformada de Laplace, acompañan esta definición para garantizar la convergencia de la integral y así mismo la existencia de la transformada de Laplace de la función $f(t)$ [8]. Como una transformación integral, la transformada de Laplace reduce una ecuación diferencial parcial en n variables independientes a una en $n-1$, la cual se espera que tenga una solución más simple o conocida. En física e ingeniería, la transformada de Laplace es usualmente aplicada a las variables en el tiempo para reconstruir el problema en el dominio Laplace (conocido como el dominio de la frecuencia en física) donde la dependencia del tiempo no aparece explícitamente pero es preservada en el parámetro s de la transformada de Laplace de manera paramétrica. La simplificación de la operación de convolución en productos algebraicos de funciones, es otra ventaja de la transformada de Laplace, por lo cual tiene importantes aplicaciones en la solución de ecuaciones integrales. Esta propiedad es útil al enfrentarse a problemas de producción tipo variable en flujo de fluidos transiente en medios porosos.

2.3. LA TRANSFORMADA INVERSA

Toda transformada de Laplace debe poseer una única transformada inversa; es decir que si $f(s)$ es una transformada de Laplace entonces debe haber una única función $f(t)$ como su transformada inversa. Para ciertas clases de funciones, un conjunto de reglas garantiza la existencia de la transformada inversa de Laplace [8]. La fórmula de inversión en el plano complejo es:

$$f(t) = \frac{1}{2\pi i} \int_{\gamma-i\infty}^{\gamma+i\infty} e^{st} f(s) ds \quad (19)$$

Esta fórmula es bien conocida pero a pesar de ello, la fórmula de inversión en el plano real no está disponible [15], esta se podría mostrar de la forma:

$$\mathcal{L}^{-1}\{\bar{f}(s)\} = f(t) \quad (20)$$

La transformada inversa $f(t)$ de la función $f(s)$, puede ser hallada evaluando la integral compleja de la ecuación (19); esto se puede si la función $f(s)$ es conocida

en la mitad compleja del plano de convergencia. En algunos problemas, la transformada de Laplace solo se conoce para valores reales y positivos de s , muchos de estos problemas son de interés práctico. En estos casos, debido a que no hay una fórmula de inversión disponible en la forma de la ecuación (20), el problema mal planteado de la ecuación integral de Volterra de la primera clase dada en la ecuación (18) necesita ser evaluada para encontrar la función original $f(t)$.

Hay tres casos generales encontrados en la inversión de la transformada de Laplace definidos por Kryzhniy [16]:

- 1) La solución se obtiene en forma analítica en el dominio Laplace: La transformada inversa puede ser hallada analíticamente o con la ayuda de tablas. Si la solución es muy complicada se puede usar la inversión numérica.
- 2) La transformada de Laplace de la solución se puede calcular en la mitad del plano complejo de convergencia: En este caso, la transformada inversa se puede obtener evaluando la integral compleja de la transformada inversa. La inversión numérica basada en el cálculo de la integral de Bromwich (cuando es posible) arroja buenos resultados.
- 3) La transformada de Laplace se puede calcular o medir solo en el eje real y positivo: En este caso, no hay fórmula de inversión exacta. Todos los métodos de inversión numérica presentan dificultades porque el problema de inversión está muy mal planteado.

La transformada inversa de Laplace puede ser hallada de diferentes formas. Por ejemplo, se puede preparar una tabla de transformadas en la cual se pongan las transformadas de muchas funciones y se recurra a esta tabla para encontrar la transformada inversa. Se puede usar la tabla de transformadas en conjunto con la ecuación y otros parámetros conocidos de la transformada. Otra técnica se basa en la integración en el plano complejo. Sin embargo, en la mayoría de problemas relacionados a las pruebas de pozos, esta última técnica puede llevar a expresiones que son muy difíciles de evaluar. Por esta razón, la tendencia actual es encontrar la transformada inversa numéricamente y presentar los resultados en la forma de una curva tipo [36].

Muchos métodos han sido creados y empleados, proporcionando diversos resultados, algunos con desempeño más favorable que otros. En este trabajo de tesis se compara el Algoritmo de Stehfest, el cual es el más usado desde hace mucho tiempo, con el Algoritmo de Iseger que es bastante reciente pero que argumenta tener algunas ventajas frente a Stehfest como el manejo de singularidades y discontinuidades en ciertas funciones.

2.4. USO DE LA TRANSFORMADA

La transformada de Laplace ha sido una herramienta estándar para la solución de problemas de transientes de flujo en medio porosos.

La transformada de Laplace ofrece muchas ventajas en la solución de problemas de análisis de transientes de presión. Usualmente estas aplicaciones llevan a una solución que necesita ser invertida numéricamente al dominio del tiempo real. El algoritmo de Stehfest es la herramienta más común usada en la ingeniería de petróleos, otros algoritmos han sido usados pero sin mucho éxito o popularidad. Un algoritmo reciente, el algoritmo de Iseger, quita la restricción de continuidad y da oportunidades para muchas aplicaciones prácticas [36].

En problemas de transientes se han encontrado un gran grupo de problemas de análisis exponencial, como lo señala Kryzhniy [15], que involucran transformadas de Laplace con valores reales, son casos que tienen problemas de inversión porque no está disponible una fórmula de inversión exacta para las funciones de valores reales.

Los problemas de transientes fluidos con datos diferenciables por partes generalmente surgen debido a las aproximaciones de los rangos y presiones tabuladas por partes, interrupción de la producción por cierres y secuencias tipo paso como en aplicaciones mini-DST. En algunos de estos casos, la función a ser invertida puede ser continua pero no es continua al derivar (por ejemplo aproximaciones por partes de datos muestreados). En otros, la función puede ser continua por partes y derivable por partes (por ejemplo una secuencia de periodos de acumulación y retiro). La transformada de Laplace de funciones continuas por partes incluye contribuciones exponenciales en los límites de sus secciones, lo que causa problemas en la inversión numérica. Si la función es discontinua en los límites de su sección, entonces la mayoría de los algoritmos de inversión requieren tratamiento especial o fallan del todo. Estas funciones derivables por partes o las funciones discontinuas aparecen comúnmente en aplicaciones de deconvolución donde los problemas numéricos inherentes debido a la naturaleza mal planteada de la deconvolución [17] se agravan aún más debido a los errores en la inversión numérica de Laplace.

En la ingeniería de petróleos se usa comúnmente el algoritmo de inversión numérica propuesto por Stehfest [28], pero este algoritmo solo es aplicable a funciones continuas lo que limita su uso para problemas que involucren cambios tipo paso y cierres. Otros algoritmos han sido aplicados en estos casos, pero no ganaron tanta popularidad debido a su complejidad e inflexibilidad en aplicaciones estándar. Algunos algoritmos de estos fueron los propuestos por Bellman [6], Crump [11] y Talbot [31]. Debido a los problemas persistentes y al aumento de interés en los algoritmos de deconvolución se ha hecho una continua búsqueda de

nuevos algoritmos de inversión de transformadas de Laplace, como el algoritmo de Gaver-Wynn-Rho presentado por Abate y Valkó [3] y Valkó y Abate [32].

Recientemente, un algoritmo presentado por Iseger [35] dice remover la restricción de continuidad y brinda oportunidades para muchas aplicaciones prácticas. Aquí se prueba el algoritmo de Iseger para varias condiciones comunes que requieren el uso de funciones derivables por partes y funciones discontinuas, casos en los que se presentan problemas de inversión con los algoritmos de inversión de la transformada de Laplace comúnmente usados. De igual manera se comparan los resultados del algoritmo de Iseger con los del algoritmo de Stehfest.

Gracias a la llegada de los computadores muchas tareas se han facilitado, y debido a esto ha crecido el interés en encontrar métodos numéricos para hallar de manera precisa las transformadas inversas de Laplace. Los métodos numéricos de inversión de Laplace según Cohen [9] se pueden clasificar así:

1. Métodos de series de expansión.
2. Métodos de Cuadratura (incluyen los métodos basados en series de Fourier).
3. Métodos de aproximación racional o racionales de aproximación.
4. El método de Talbot.
5. Métodos basados en la fórmula de inversión Post – Widder.
6. Métodos de regularización.

Las categorías 2, 4 y 5 abarcan la mayoría de los algoritmos de inversión usados en la ingeniería de petróleos. Por ejemplo, el algoritmo de Stehfest [28] está basado en la fórmula de inversión Post–Widder. Un ejemplo de un algoritmo de inversión que usa la regla de cuadratura Gauss-Legendre es el propuesto por Bellman [6]. El algoritmo propuesto por Crump [11] es un método de cuadratura usando series de Fourier. El nuevo algoritmo de Iseger [35] es también un algoritmo basado en series de Fourier.

Los algoritmos ya mencionados tienen características comunes de las categorías a las que pertenecen. El algoritmo de Stehfest [28] muestra algunas características de suavizado como lo indica Onur y Reynolds [20], esto se debe a que este algoritmo se basa en análisis de datos estocásticos, que se origina de la interpretación probabilística de la transformada de Laplace de Gaver [21]. Aunque esta propiedad es útil para tener unas representaciones de la función invertida más suaves como en cálculos derivativos [20], también causa algo de pérdida de precisión y calidad si la función que se invierte tiene alguna discontinuidad o cambios tipo paso. Por otro lado, los algoritmos basados en las series de Fourier, como los son el de Crump [11] y el de Iseger [35], brindan una mejor precisión y preservan la naturaleza y calidad de la función a la hora de enfrentarse con funciones paso y funciones con discontinuidades y singularidades, pero también pueden llevar a resultados de inversión más oscilatorios [36].

2.5. ALGORITMO DE STEHFEST

El algoritmo presentado por Stehfest [28] ha ganado amplia aceptación por investigadores en el campo de las pruebas de pozos.

El algoritmo de Stehfest está basado en las siguientes fórmulas:

$$V_i = (-1)^{n/2+i} \sum_{k=(i+1)/2}^{\min(i, n/2)} \frac{k^{n/2} (2k)!}{(n/2 - k)! k! (k-1)! (i-k)! (2k-i)!} \quad (21)$$

$$f(t) = \frac{\ln 2}{t} \sum_{i=1}^n V_i P\left(\frac{\ln 2}{t} i\right) \quad (22)$$

Donde P es la presión y t es el tiempo. El número n , en estas expresiones debe ser optimizado por prueba y error. Incrementando n se aumenta la precisión de los resultados hasta cierto punto, y luego la precisión disminuye debido a errores de redondeo, ya que la longitud de la palabra en el computador es finita. Se debe tener en cuenta que $f(t) = L^{-1}P(z)$, y z es reemplazado por $i \ln 2/t$, donde t es el tiempo en el cual se requiere la transformada inversa. También se tiene en cuenta que para un n dado el algoritmo de Stehfest requiere el cálculo de V_i solo una vez.

El programa de Stehfest está escrito en Visual Basic 6.0. y hace parte de un proyecto de tesis anterior; está escrito para hallar la transformada inversa de dos modelos de pozos dados. Para una $f(z)$ dada para la cual no se sabe la transformada inversa, n puede ser optimizada recurriendo a una tabla de transformadas y escogiendo una función que esté cerca de la función a la mano. También, si n no es seleccionada adecuadamente, la gráfica de la transformada inversa tenderá a oscilar, mientras que un valor escogido apropiadamente producirá una buena transformada inversa.

2.6. ALGORITMO DE ISEGER

2.6.1. Introducción

El énfasis en el cálculo probabilístico aumenta el valor de los modelos estocásticos en cola, confiabilidad y problemas de inventario. Incluir algoritmos para el cálculo de distribuciones de interés se está convirtiendo en algo estándar para el modelado y análisis. Varias herramientas han sido desarrolladas para este propósito. Una herramienta muy poderosa es la inversión numérica de Laplace. Las distribuciones probabilísticas pueden ser a menudo caracterizadas en términos de transformadas de Laplace. Muchos resultados en cola y confiabilidad, entre otros, están dados en forma de transformadas y se prestan para cálculos

prácticos ya que hay disponibles métodos rápidos y precisos para la inversión numérica de Laplace. La inversión numérica de Laplace es mucho más fácil de usar de lo que a menudo parece. Aquí se presenta un nuevo y efectivo algoritmo para la inversión numérica de Laplace. El nuevo algoritmo supera los métodos existentes, en especial cuando la función que se va a invertir tiene discontinuidades o singularidades, como es frecuente en el caso de aplicaciones.

El algoritmo puede calcular los valores de la función deseada $f(k\Delta)$, $k = 0, 1, \dots, M-1$ para un rango mucho más grande de transformadas de Laplace que aquellas que se pueden invertir con los métodos conocidos en la literatura. Puede invertir transformadas de Laplace de funciones con discontinuidades y singularidades, incluso si no se conoce previamente la localización de éstas discontinuidades y singularidades, y también funciones sin suavizado local y sin límites. Solo se necesitan valores numéricos de la transformada de Laplace, los cálculos gastan un tiempo de $M \log(M)$, y los resultados son casi de precisión de máquina. Con el nuevo método, se pueden calcular valores de la función $f(k\Delta)$, $k = 0, 1, \dots, M-1$, de una en un tiempo $M \log(M)$.

Hay muchos algoritmos de inversión de Laplace conocidos. Cuatro métodos ampliamente usados son (1) el método de series de Fourier, el cual está basado en la fórmula de sumatoria de Poisson [4,5,7,12,22,26], (2) el algoritmo de Gaver-Stehfest, que está basado en combinaciones de las funcionales de Gaver [14,28], (3) el método Weeks, que está basado en transformaciones bilineales y expansiones de Laguerre [1,2,34], y (4) el método Talbot, que se basa en la deformación del contorno de la integral de inversión de Bromwich [19,31].

El nuevo algoritmo está basado en la ya bien conocida fórmula de sumatoria de Poisson y por lo tanto puede ser visto como un supuesto método de series de Fourier, desarrollado en la década de 1960 por Dubner y Abate [12]. La fórmula de sumatoria de Poisson relaciona una suma infinita de los valores de la transformada de Laplace con la transformada-z (series de Fourier) de los valores de la función $f(k\Delta)$, $k = 0, 1, \dots$. Desafortunadamente, la suma infinita de los valores de la transformada de Laplace, en general, convergen muy lentamente. En su artículo inicial, Abate y Whitt [4] usaron una técnica de aceleración llamada sumatoria de Euler para acelerar la velocidad de convergencia de la suma infinita de los valores de la transformada de Laplace. Recientemente, Sakurai [26] extendió la sumatoria de Euler para que sea eficaz para una clase más amplia de funciones. Una de las desventajas de todas las variantes de los métodos de Fourier es que a menos que se haya especificado información acerca de la localización de las singularidades, las técnicas de aceleración no son muy eficaces y la convergencia es lenta.

Aquí se presenta una regla de cuadratura Gaussiana para la suma infinita de los valores de la transformada de Laplace. La regla de cuadratura Gaussiana, aproxima con precisión la suma infinita a una suma finita. Entonces se calculan los

valores de la función $f(k\Delta)$, $k = 0, 1, \dots, M-1$, de manera eficiente con el conocido algoritmo de la transformada rápida de Fourier (sus siglas en inglés FFT, Fast Fourier Transform) [10]. Para funciones suavizadas, los resultados son casi de precisión de máquina. Con una simple modificación, se pueden manejar discontinuidades conocidas en los puntos $k\Delta$, $k = 0, 1, \dots$, de tal manera que el tiempo de ejecución de el algoritmo sigue siendo insensible a el número de discontinuidades y se obtienen resultados cercanos a precisión de máquina.

2.6.2. Esquema del método

Sea f una función integrable de valores complejos de Lebesgue, con $e^{-\alpha t} f(t) \in L^1[0, \infty)$, para todo $\alpha > c$. La transformada de Laplace de f está definida por la integral

$$\hat{f}(s) := \int_0^{\infty} e^{-st} f(t) dt, \quad \text{Re}(s) > c. \quad (23)$$

La fórmula de sumatoria de Poisson relaciona una suma infinita de valores de transformada de Laplace con la transformada Z (series de Fourier amortiguadas) de los valores de la función $f(k\Delta)$, $k = 0, 1, \dots$.

TEOREMA 1 (PSF): Suponga que $f \in L^1[0, \infty)$ y f es de variación acotada. Entonces para todo $v \in [0, 1)$,

$$\sum_{k=-\infty}^{\infty} \hat{f}(a + 2\pi i(k + v)) = \sum_{k=0}^{\infty} e^{-ak} e^{-i2\pi kv} f(k); \quad (24)$$

aquí, $f(t)$ se debe entender como $(f(t^+) + f(t^-))/2$, el llamado factor de amortiguamiento a es un número real dado, y para denotar el número imaginario $\sqrt{-1}$ se usa i .

Una prueba de este clásico resultado puede ser encontrado, por ejemplo, en Abate y Whitt [4] o Mallat [18]. El lado derecho de (24) es la de la serie de Fourier (amortiguada) de los valores de la función $\{f(k); k=0, 1, \dots\}$. Aquí se presenta una regla de cuadratura de Gauss para el lado izquierdo de (24); es decir, se aproxima esta suma infinita a una suma finita,

$$\sum_{k=1}^n \beta_k \hat{f}(a + i\lambda_k + 2\pi i v) \quad (25)$$

con $\{\beta_k\}$ números positivos dados y los $\{\lambda_k\}$ números reales dados. En el Anexo A se pueden encontrar estos números para varios valores de n ; así mismo se hablan de las condiciones que se tienen en cuenta para escoger el valor de n . En este proyecto de tesis se evalúan dos funciones correspondientes a dos modelos de

pozo cuyas características llevaron a escoger $n=16$ [35,36]. Se pueden calcular los valores de la función $\{f(k); k = 0, 1, \dots, M-1\}$ por

$$\frac{e^{ak}}{M_2} \sum_{j=0}^{M_2-1} \cos\left(\frac{2\pi jk}{M_2}\right) \sum_{i=1}^n \beta_i \hat{f}\left(a + i\lambda_i + \frac{2\pi ij}{M_2}\right) \quad (26)$$

aquí, M_2 es una potencia de 2 dada. Con el bien conocido algoritmo de la Transformada Rápida de Fourier [10] se pueden calcular estas sumas en un tiempo $M_2 \log(M_2)$. Se pueden calcular los valores de las funciones $f(k\Delta)$, $k = 0, 1, \dots$, aplicando (26) a la transformada de Laplace escalada

$$\hat{f}_\Delta(s) = \frac{1}{\Delta} \hat{f}\left(\frac{s}{\Delta}\right) \quad (27)$$

que es la transformada de Laplace de la función $f_\Delta(t) = f(\Delta t)$.

2.6.3. La Regla de Cuadratura de Gauss

Se asocia un producto punto con el lado izquierdo de (24), por ejemplo $\langle \dots \rangle_{Q_v}$.

DEFINICIÓN 1: Sea el producto punto $\langle \dots \rangle_{Q_v}$ dado por

$$\langle f, g \rangle_{Q_v} = \sum_k \frac{1}{|2\pi(k+v)|^2} f\left(\frac{1}{2\pi i(k+v)}\right) g^*\left(\frac{1}{2\pi i(k+v)}\right) \quad (28)$$

Teniendo $\mathbf{1}(s) = 1$ y $\Psi \hat{f}(s) = s^{-1} \hat{f}(s^{-1})$, se puede escribir el lado izquierdo de (24) como $\langle \Psi \hat{f}, \Psi \mathbf{1} \rangle_{Q_v}$. La idea es aproximar este producto punto a la regla de cuadratura de Gauss. Como es costumbre, se asocia el producto punto $\langle \dots \rangle_{Q_v}$ con la norma $\|\cdot\|_{Q_v}$, que indujo el espacio de secuencias $L^2(Q_v)$. Se dice que una función f pertenece a $L^2(Q_v)$ si y solo si la secuencia $\{f(1/2\pi i(k+v))\}$ pertenece a $L^2(Q_v)$. Sean los polinomios $\{q_n^v; n \in N_0\}$ dados por

$$q_n^v(s) := p_n(s) - (-1)^n e^{-i2\pi v} p_n(-s), \quad (29)$$

donde

$$p_n(s) = \sqrt{2n+1} \sum_{k=0}^n \frac{(k+n)! (-s)^k}{(n-k)! k!} \quad (30)$$

El siguiente resultado se mantiene.

TEOREMA 2: La serie $\{q_j^v; j = 0, 1, \dots\}$ es una serie ortogonal completa de polinomios en $L^2(Q_v)$.

Ahora se puede obtener fácilmente la regla de cuadratura de Gauss deseada (consultar la definición C.1 en el Anexo C). Se denota esta regla de cuadratura con $\langle \cdot, \cdot \rangle_{Q_v^n}$.

DEFINICIÓN 2: El producto punto $\langle \cdot, \cdot \rangle_{Q_v^n}$ está dado por

$$\langle f, g \rangle_{Q_v^n} = \sum_{k=1}^n \alpha_k^v \hat{f}(\mu_k^v) \hat{g}^*(\mu_k^v) \quad (31)$$

$\{\mu_k^v\}$ son los ceros del polinomio q_n^v . Los llamados números Christopel $\{\alpha_k^v\}$ son números positivos dados por

$$\alpha_k^v = \frac{1}{\sum_{j=1}^n |q_j^v(\mu_k^v)|^2} \quad (32)$$

Se sabe que las raíces de los polinomios ortogonales son todas diferentes y se encuentran ubicadas en el soporte del producto interno [30]; por lo tanto, las raíces son todas distintas y se encuentran en el eje imaginario. Teniendo $\mathbf{1}(s) = 1$ y $\Psi \hat{f}(s) = s^{-1} \hat{f}(s^{-1})$, se aproxima $\langle \Psi \hat{f}, \Psi \mathbf{1} \rangle_{Q_v}$ con esta regla de cuadratura de Gauss y así obtener la fórmula de la cuadratura

$$\sum_{k=1}^n \frac{\alpha_k^v}{|\mu_k^v|^2} \hat{f}\left(\frac{1}{\mu_k^v}\right) \approx \sum_{k=-\infty}^{\infty} \hat{f}(2\pi i(k+v)) = \sum_{k=0}^{\infty} \exp(-2\pi i k v) f(k) \quad (33)$$

donde la última identidad se deduce de la Fórmula de Suma de Poisson PSF (Poisson Summation Formula) (24). Considerando que solo la parte real se obtiene de la fórmula de cuadratura

$$\sum_{k=1}^n \frac{\alpha_k^v}{|\mu_k^v|^2} \operatorname{Re}\left(\hat{f}\left(\frac{1}{\mu_k^v}\right)\right) \approx \sum_{k=0}^{\infty} \cos(2\pi k v) f(k) \quad (34)$$

aquí se toman α_k^v como reales. Del mismo modo que en el algoritmo de Abate y Whitt [4], se usa un factor de amortiguamiento para funciones con soporte contenido en $[0, \infty)$; es decir, se toma la transformada de Laplace $\hat{f}(a+s)$, que es la transformada de Laplace de $e^{-at}f(t)$. El resultado es la regla de cuadratura

$$F_a(v) \approx \sum_{k=0}^{\infty} e^{-ak} \cos(2\pi kv) f(k) \quad (35)$$

donde

$$F_a(v) = \sum_{k=1}^n \frac{\alpha_k^v}{|\mu_k^v|^2} \operatorname{Re} \left(\hat{f} \left(a + \frac{1}{\mu_k^v} \right) \right) \quad (36)$$

Quedan por calcular los valores de la función $f(k)$, $k=0,1,\dots,M-1$. Esto se puede realizar por la fórmula de inversión de la serie de Fourier discreta

$$f(k) \approx \frac{e^{ak}}{M_2} \sum_{j=0}^{M_2-1} \cos \left(\frac{2\pi \left(j + \frac{1}{2} \right) k}{M_2} \right) F_a \left(\frac{j + \frac{1}{2}}{M_2} \right) \quad (37)$$

donde M_2 es una potencia de 2 dada. Con el conocido algoritmo de la transformada rápida de Fourier [10], se pueden calcular las sumas (37) en un tiempo de $M_2 \log(M_2)$.

Nota 1: La regla de cuadratura también es válida para una función f con soporte $(-\infty, \infty)$. La fórmula (35) quedaría como

$$F_a(v) \approx \sum_{k=-\infty}^{\infty} \cos(2\pi kv) e^{-ak} f(k\Delta) \quad (38)$$

Nota 2: Para funciones suaves en $(-\infty, \infty)$ o $[0, \infty)$, la fórmula de cuadratura es extremadamente precisa. Para funciones suaves, el error de aproximación de (35) es aproximadamente

$$\frac{1}{2} (-1)^n \frac{(n-1)! n!}{(2n-1)!} \sum_{j=1}^{\infty} j e^{-i2\pi jv} \frac{f^{(2n+1)}(j-1+\alpha)}{(2n+1)!} \quad (39)$$

con α como algún número en $(0,2)$. Esto implica que si se calculan los valores de la función $f(k\Delta)$, $k=0,1,\dots$, al invertir la transformada de Laplace

$$\hat{f}_{\Delta}(s) = \frac{1}{\Delta} \hat{f}\left(\frac{s}{\Delta}\right) \quad (40)$$

con (37), el error de aproximación es del orden $O(\Delta^{2n+1})$. La fórmula (39) muestra también que si

$$\frac{f^{(2n+1)}(j-1+\alpha)}{(2n+1)!} \quad (41)$$

está acotada, entonces la fórmula de cuadratura converge más rápido que cualquier potencia de n .

Nota 3: La convergencia de la regla de cuadratura es insensible a la discontinuidad en $t=0$.

Nota 4: La cuadratura de Gauss es un método estándar para la evaluación numérica de integrales [29]. Piessens trató de aplicar la cuadratura de Gauss directamente a la integral de inversión de Bromwich [23,24]. En [24] se informa para la primera transformada de Laplace de la Tabla 3 un error de aproximación promedio de 1×10^{-4} para una regla de cuadratura de 17 puntos (para los valores $t = 0, 1, \dots, 12$). Este método da un error de aproximación promedio de 1×10^{-15} (consultar la Tabla 3).

No.	Function	Laplace Transform
1	$J_0(x)$	$(s^2 + 1)^{-1/2}$
2	$\exp(-x/2)$	$(s + \frac{1}{2})^{-1}$
3	$\exp(-0.2x)\sin(x)$	$((s + 0.2)^2 + 1)^{-1}$
4	1	s^{-1}
5	x	s^{-2}
6	$x \exp(-x)$	$(s + 1)^{-2}$
7	$\sin(x)$	$(s^2 + 1)^{-1}$
8	$x \cos(x)$	$(s^2 - 1)(s^2 + 1)^{-2}$

No.	$\Delta = 1/16$	$\Delta = 1$	$\Delta = 10$
1	1e-15	1e-15	5e-13
2	1e-15	1e-15	3e-16
3	2e-16	1e-15	3e-12
4	1e-15	1e-15	1e-15
5	3e-15	5e-15	6e-15
6	2e-16	3e-16	2e-16
7	1e-15	1e-15	5e-12
8	4e-16	6e-15	2e-12

Tabla 3. Resultados para las funciones analíticas de prueba [35]

El error absoluto promedio $\sum_{k=0}^{M-1} |e(k\Delta)| / (1 + e(k\Delta))$ se presenta para la inversión de ocho funciones analíticas en los puntos $\{0, 1/16, \dots, 31/16\}$, $\{0, 1, \dots, 31\}$, y $\{0, 10, \dots, 310\}$. De esta tabla, se puede concluir que para funciones analíticas, los resultados son casi de precisión de máquina.

2.6.4. Resumen y Análisis del método

Como se indicó en la introducción, el renovado interés en la inversión numérica de funciones singulares y discontinuas ha sido provocado por un nuevo algoritmo presentado por Iseger [35]. El algoritmo de Iseger elimina la restricción de continuidad que presentan algunos de los algoritmos más comunes de inversión, tales como el algoritmo de Stehfest [28], y ofrece oportunidades para muchas aplicaciones prácticas. El algoritmo propuesto por Iseger [35] es un método de series de Fourier, que está basado en la fórmula de sumatoria de Poisson. En este algoritmo, 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. Iseger [35] muestra que los resultados del algoritmo son casi de precisión de máquina.

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, incluso si los puntos de discontinuidad y singularidad no son conocidos de antemano. La implementación de este algoritmo puede ser un poco más complicada que la implementación del algoritmo de Stehfest [28], pero es comparable a la implementación de otros algoritmos comunes. Aquí se presentan algunas características y parámetros del algoritmo de Iseger que se han observado durante las aplicaciones de problemas de flujo de fluidos en medios porosos.

Un parámetro crítico usado en el algoritmo de Iseger es Δ (delta) el cual se calcula a partir de la siguiente relación:

$$\Delta = \frac{T}{M} \quad (42)$$

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. Δ es un parámetro crítico porque relaciona dos parámetros claves en el algoritmo de Iseger que son el tiempo T y el número de muestras M. Este parámetro relaciona de manera proporcional estas dos variables (t y M) y traza la naturaleza lineal del algoritmo; a su vez, indica el incremento de tiempo que habrá entre un punto muestreado y el siguiente. Es importante recalcar que el error de aproximación también depende de Δ . Por esta razón, después de varias pruebas realizadas en [36], se encontró que deben cumplirse dos condiciones para obtener resultados precisos. La primera condición que debe cumplirse es que $(1/\Delta) \in N$ y además se debe cumplir que,

$$M \geq 10\Delta M, \text{ lo que equivale a decir que } M \geq 10T. \quad (43)$$

Para demostrar cómo afectan estas condiciones, se consideró la inversión de una función de paso unitario con un período de 20 horas, definido de manera arbitraria para efectos de prueba, como se muestra en la figura 8. El mínimo número de puntos de inversión requeridos para este caso es $M \geq 200$, esto se obtiene aplicando la ecuación (43). Si se usan 220 puntos entonces la condición de la ecuación (43) se satisface y $(1/\Delta) = 11 \in N$. La figura 8 muestra que se obtienen inversiones precisas de la función paso con 220 puntos de datos. Si se usan, por ejemplo, 103 puntos de datos, la condición en la ecuación (43) no se cumple y si se usan 223 puntos de datos, se cumple la condición de la ecuación (43) pero $(1/\Delta) \notin N$. Para ambos casos, las inversiones no son suficientemente precisas. También se observó que si las características de la función cambian en intervalos cortos (por ejemplo, variaciones de tipo y presión debido a reducciones cortas y

períodos de acumulación en un test mini-DST), se requiere tener un M grande para obtener inversiones precisas.

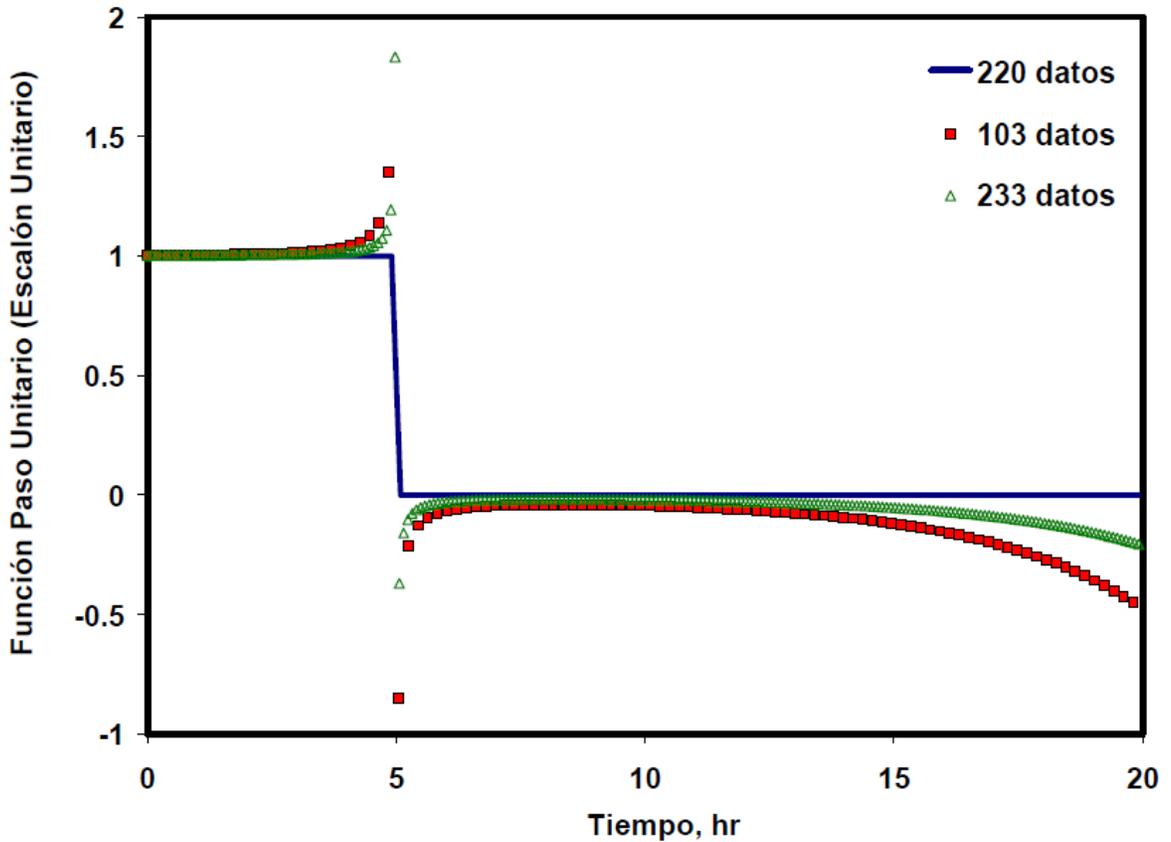


Fig. 8. Inversión numérica de una función escalón unitario por el algoritmo de Iseger. Efecto del número de puntos de inversión [36]

El algoritmo de Iseger usa $M_2 = nrp \times M$ puntos de sobremuestreo para calcular las inversiones en M puntos de datos. La elección del parámetro M_2 es una compensación entre el tiempo de ejecución y la precisión del algoritmo. El tiempo de ejecución real es $M_2 \log(M_2)$; por lo tanto, desde esta perspectiva, se desea escoger M_2 tan pequeño como sea posible. Sin embargo, para obtener un resultado numéricamente estable, se debe escoger M_2 tan grande como sea posible.

El parámetro nrp del algoritmo de Iseger es dado por el autor. A pesar de que no especifica el significado de las letras, sí menciona que este es el factor de sobremuestreo, lo que incrementa la precisión de los resultados. Iseger [35] recomienda usar $M_2=8M$, es decir recomienda usar $nrp = 8$ (dado por el autor) 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 [35]. Sin embargo, 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 [36]. La figura 9 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.

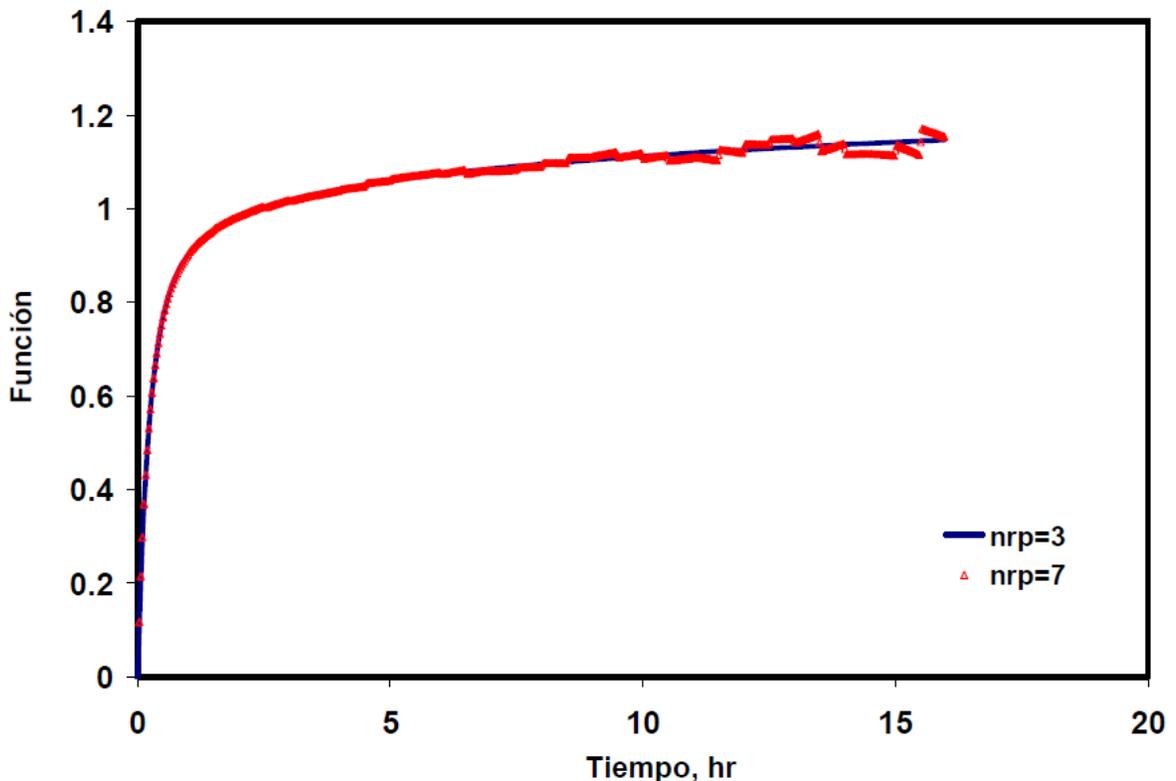


Fig. 9. Efecto de n_{rp} en la inversión numérica de una función por el algoritmo de Iseger [36]

2.6.4.1. Ejemplos de verificación

A continuación se presentan ejemplos para verificar el éxito del algoritmo de Iseger en la inversión de funciones que son derivables por partes o discontinuas. Se compararon los resultados con los del algoritmo de Stehfest para trazar las diferencias respecto a los algoritmos estándar.

La figura 10 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 10 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).

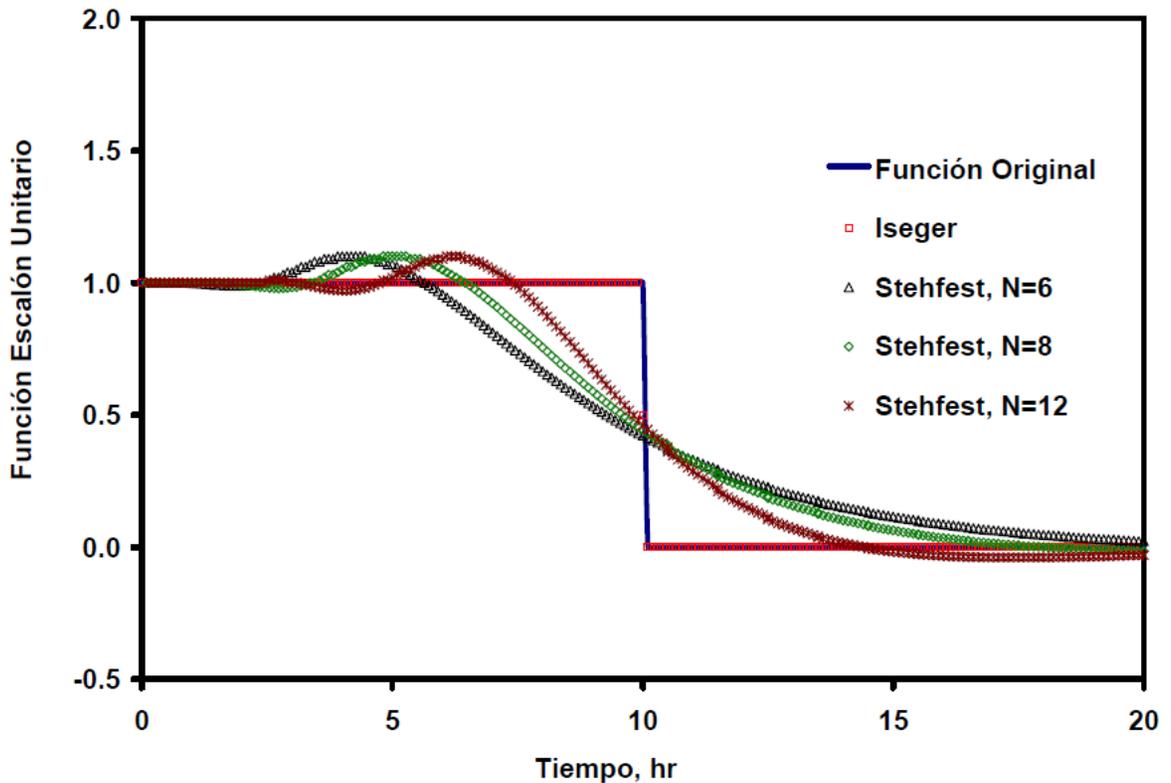


Fig. 10. Inversión numérica de una función escalón unitario; comparación de los algoritmos de Iseger y Stehfest [36]

En la figura 11 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 11, se usó $M = 512$ y $n_{rp} = 3$ 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.

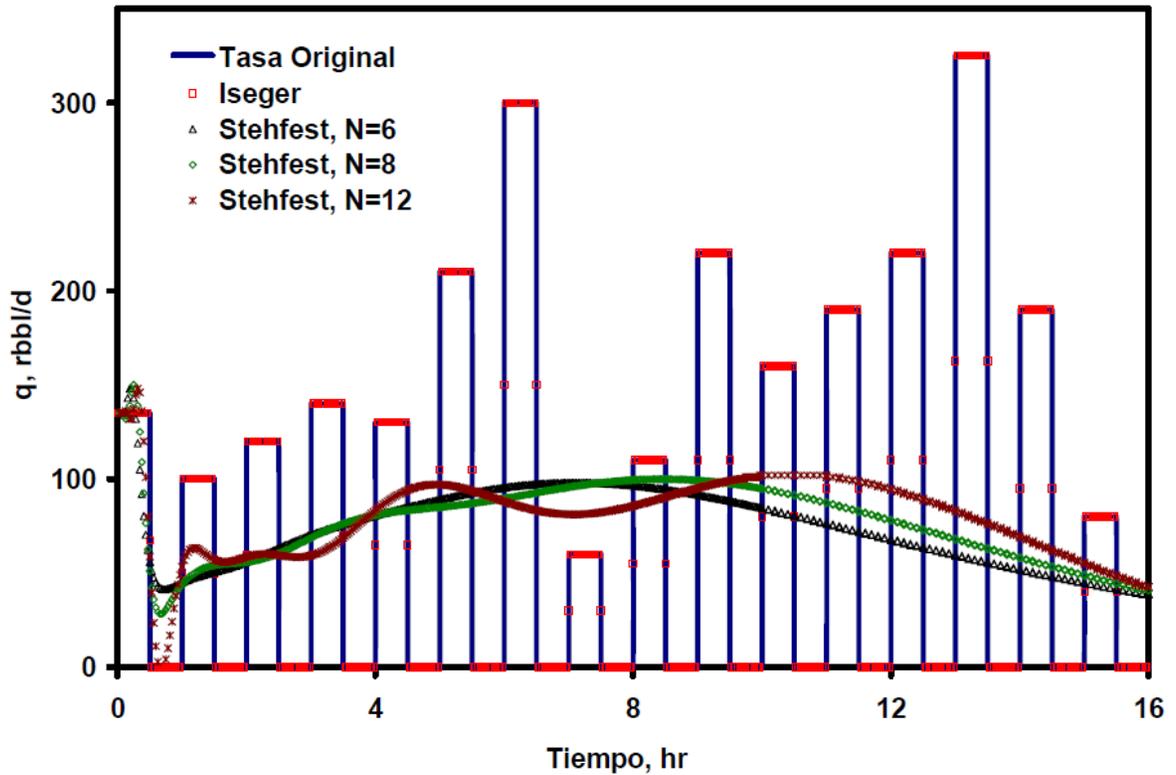


Fig. 11. Inversión numérica de una función con múltiples cambios tipo escalón; comparación de los algoritmos de Iseger y Stehfest [36]

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 12. Para generar la función de este ejemplo se usó la función de la figura 12 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 Roumboutsos – Stewart (el algoritmo Onur – Reynolds arrojó los mismos resultados) y se invirtió usando los algoritmos de Iseger y Stehfest. Como se muestra en la figura 12, 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.

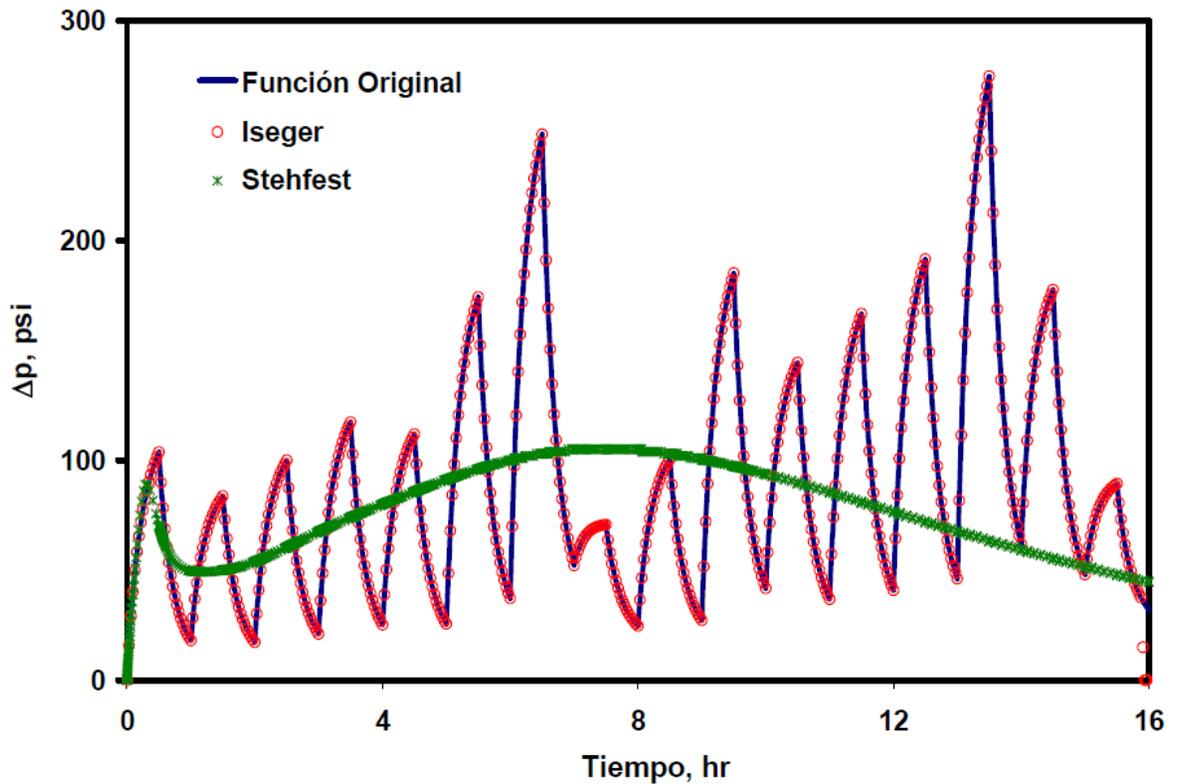


Fig. 12. Inversión numérica de una función derivable por partes; comparación de los algoritmos de Iseger y Stehfest [36]

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.

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 [35].

3. DESARROLLO DEL SOFTWARE

El software de este proyecto se realizó utilizando Visual Basic .Net 2008 y su objetivo es realizar una comparación entre el algoritmo de Stehfest y el de Iseger, que sirven para obtener la transformada inversa de Laplace de dos modelos de yacimiento dados. La inversa de Laplace proporciona soluciones en problemas de análisis de transientes de presión en pozos. En el software se realizó una interfaz gráfica para realizar la comparación de los resultados de ambos métodos.

El algoritmo de Stehfest ya estaba implementado pero se pasó de Visual Basic 6.0 a Visual Basic .Net 2008. Debido a que el algoritmo de Iseger trabaja de manera lineal el tiempo y no lo expande como estaba en el algoritmo de Stehfest, se hizo un ajuste para que este último trabajara del mismo modo y ambos tuvieran aproximadamente el mismo número de puntos de evaluación de la función, tal que se pudiera establecer una comparación. Ahora el tiempo aumenta de manera constante de acuerdo a la cantidad de puntos de muestreo. A la variable de aumento entre punto y punto se le llamó *delta* (Δ), y corresponde a la razón entre el tiempo y el número de puntos, de manera análoga al algoritmo de Iseger.

Para la parte de Iseger fue necesario buscar el algoritmo, el cual se encontró en [35]; allí se explica el funcionamiento del algoritmo, el esquema de funcionamiento así como algunos ejemplos de la implementación de varios algoritmos con diferentes funciones. Se usó el algoritmo más robusto, el que tiene un uso más amplio.

El autor presenta variaciones del algoritmo, empezando por uno básico y le va haciendo modificaciones para que sea de más amplia utilización en diversos tipos de funciones. El primero que presenta, es un algoritmo que funciona muy bien, con una precisión adecuada invirtiendo funciones suaves. Luego realiza ciertas modificaciones que se detallan en [35] para que el algoritmo obtenga los mismos resultados precisos cuando son funciones sin suavizado. Posteriormente, el autor incluye una modificación para que además de las funciones suaves y sin suavizado, el algoritmo pueda manejar con éxito aquellas funciones suavizadas por partes y proporcione resultados con una precisión casi de máquina para estas funciones con discontinuidades en los puntos $k\Delta$, $k \in Z$ [35]. Más adelante, presenta el que Iseger denomina como el algoritmo robusto para la inversión de Laplace. La modificación es efectiva para casi todas las clases de discontinuidades, singularidades y sin suavizado local, obteniendo resultados cercanos a la precisión de máquina en puntos de continuidad para casi todas las funciones, incluso si no se saben de antemano las discontinuidades y singularidades [35]. Este último fue el utilizado para el desarrollo de este proyecto de tesis, por lo tanto, se puede afirmar que es el más robusto porque puede manejar singularidades y discontinuidades sin problema alguno [35]. Es más robusto, en cuanto a que tiene un uso más amplio.

El algoritmo de Iseger se desarrolla en el campo complejo para lo cual se utilizó un complemento llamado “Extreme Optimization Numerical Libraries for .NET 3.6”, el cual es de licencia gratuita de prueba por 60 días renovables y que contiene las librerías necesarias para el manejo de números complejos y operaciones entre ellos. De este modo se pudo declarar el conjunto de variables que eran de naturaleza compleja, incluyendo vectores y matrices, así como desarrollar las operaciones.

Algunos parámetros del algoritmo de Iseger se modificaron con respecto a los valores dados en el documento original del autor, esto se hizo para lograr mayor estabilidad en los resultados obtenidos al evaluar modelos de yacimientos. Uno de los parámetros es n_{rp} 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 $n_{rp} = 8$; luego se estableció, mediante pruebas, que usando $n_{rp} = 3$ con la transformada de Fourier discreta se obtienen resultados más estables para estas aplicaciones [36].

Otro de los parámetros que se ajustó a ciertas condiciones es Δ , el cual es un parámetro crítico usado en el algoritmo de Iseger que relaciona el número de puntos para los cuales se calcula la transformada inversa de Laplace con el período para el cual las inversiones se calculan. Para este parámetro se estableció que $(1/\Delta) \in N$ y que la condición $M \geq 10\Delta M$, que es igual a decir que $M \geq 10T$, debe cumplirse, por lo que se agregó un bucle que garantice que esta condición se cumpla.

Se incluyó en el código unas líneas que aseguran que se cumpla la anterior condición, si los datos introducidos no satisfacen la condición, estas líneas se encargan de calibrar los datos hasta que se cumplan.

```
Do
delta = Total_Hour / M
variable1 = 1.0 / delta
variable2 = 1.0 / delta
  If (variable1 = variable2 And M > (10 * delta * M)) Then
    M = M
    Exit Do
  Else
    M = M + 1
    delta = Total_Hour / M
  End If
Loop
```

Como se mencionó anteriormente, otra modificación fue hecha para el tiempo T en el algoritmo de Stehfest, donde se ajustó el incremento del tiempo para que fuera de manera lineal, tal como pasa en el algoritmo de Iseger y así poder establecer una comparación punto a punto.

Los anteriores parámetros se definieron en la rutina Public Function Process, la cual contiene el código principal del algoritmo de Iseger y es a donde se hace el retorno de los resultados de tiempo, presión y derivada de presión. Esta rutina se encuentra dentro de la clase Iseger.vb.

Los modelos de yacimiento utilizados manejan funciones de Bessel. Para el algoritmo de Stehfest ya se encontraban listas estas funciones, pero no se pudieron utilizar para Iseger debido a que el algoritmo de Iseger maneja números complejos y por lo tanto las funciones de Bessel deben ser las complejas. Se debieron buscar y agregar las subrutinas de las funciones de Bessel en el plano complejo; estas se encontraron en lenguaje FORTRAN, se tradujeron y adecuaron a su uso en este proyecto. Las funciones de Bessel complejas se programaron en la clase ComplexBessel.vb que contiene las rutinas de las diferentes funciones de Bessel y herramientas necesarias para su correcto funcionamiento.

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 en el documento [35]. Cuando el valor de n , es otro de los anteriormente mencionados, las constantes también cambian su valor. Los valores de beta y lambda para diferentes valores de n se encuentran en el Anexo A. En estas constantes se puede ver el uso de la librería para el manejo de números complejos previamente mencionada.

Estas constantes fueron incluidas en la clase Iseger.vb y son las siguientes:

```
a(1) = 2.0
a(2) = 2.0
a(3) = 2.0000000000000008
a(4) = 2.0000000000000008
a(5) = 2.00000030233693
a(6) = 2.00000030233693
a(7) = 2.00163683400961
a(8) = 2.00163683400961
a(9) = 2.19160665410378
a(10) = 2.19160665410378
a(11) = 4.01375304677447
a(12) = 4.01375304677447
a(13) = 109.907452904076
a(14) = 109.907452904076
a(15) = 11.8855502586988
a(16) = 11.8855502586988
l(1) = New DoubleComplex(0.0, -3.14159265358979)
l(2) = New DoubleComplex(0.0, 3.14159265358979)
l(3) = New DoubleComplex(0.0, -9.42477796076937)
l(4) = New DoubleComplex(0.0, 9.42477796076937)
l(5) = New DoubleComplex(0.0, -15.7079633498486)
l(6) = New DoubleComplex(0.0, 15.7079633498486)
l(7) = New DoubleComplex(0.0, -21.9918840702852)
l(8) = New DoubleComplex(0.0, 21.9918840702852)
```

```
l(9) = New DoubleComplex(0.0, -28.4288098692614)
l(10) = New DoubleComplex(0.0, 28.4288098692614)
l(11) = New DoubleComplex(0.0, -37.4385643171158)
l(12) = New DoubleComplex(0.0, 37.4385643171158)
l(13) = New DoubleComplex(0.0, -173.674723843715)
l(14) = New DoubleComplex(0.0, 173.674723843715)
l(15) = New DoubleComplex(0.0, -59.3141454252504)
l(16) = New DoubleComplex(0.0, 59.3141454252504)
```

Se incluyeron varias funciones que juntas permiten que se pueda introducir una función cualquiera que se quiera invertir del dominio Laplace al dominio del tiempo. Estas funciones están programadas y distribuidas en varias clases y se inician mediante la función `Funcion.Procesar`, es decir la rutina `Procesar` de la clase `Funcion.vb`. Las funciones correspondientes a los yacimientos de interés para este proyecto están escritas dentro del código pero son procesadas de igual manera.

Dentro de la clase `Iseber.vb` se incluye la subrutina `IDFT(fs(), NH)` que sirve para calcular la transformada de Fourier discretizada, procedimiento que hace parte del algoritmo de `Iseger`. Dentro de esta misma clase se incluyó la subrutina para generar el archivo de texto plano con los resultados del algoritmo de `Iseger`.

Para hallar la derivada de la presión se utilizó una clase llamada `Derivada.vb` ubicada en la carpeta utilidades junto a la clase de las funciones de Bessel complejas. La clase de la derivada tiene dos partes, una función y una subrutina, la subrutina es la de hallar los coeficientes necesarios y se llama "coeficientes" y la función se llama "derivar" que es como tal la que se llama desde el programa.

En la clase `Funcion.vb` se encuentra la función `Procesar`, por medio de la cual se desglosa la ecuación del modelo en dominio Laplace a invertir al dominio del tiempo y también se encuentran las funciones de Bessel en el dominio Real que son las que se utilizan para el algoritmo de `Stehfest`. Las de Bessel complejas no se incluyeron en esta clase debido a su extensión en el código y para diferenciarlas más fácilmente en el momento de su uso.

Por último, la clase `Stehfest.vb` se modificó lo estrictamente necesario ya que funcionaba perfectamente en Visual 6.0. Esta clase consta de todo lo necesario para poner en marcha el algoritmo de `Stehfest`, incluyendo la subrutina de la generación del archivo de texto plano con los resultados. De aquí solo se extrajo la parte de la derivada para ponerla como función compartida para que también se pudiera utilizar por el algoritmo de `Iseger`.

4. ANÁLISIS DE RESULTADOS

Se realizaron varias pruebas al software desarrollado. Se probó que tuviera los resultados adecuados en ambos modelos de pozo. El principal inconveniente se presentó con la implementación del código del algoritmo de Iseger, por la complejidad de los procedimientos matemáticos empleados en el algoritmo y por el uso de las funciones de Bessel en el plano complejo con las cuales no se contaba al principio. Se probó el algoritmo de Iseger con diferentes funciones ingresadas en dominio Laplace. Para estas pruebas, el tiempo se definió arbitrariamente como $Total_Hour=20$ y por el parámetro delta se definió el número de muestras como $M=220$. Aquí está la función escalón unitario (1/s):

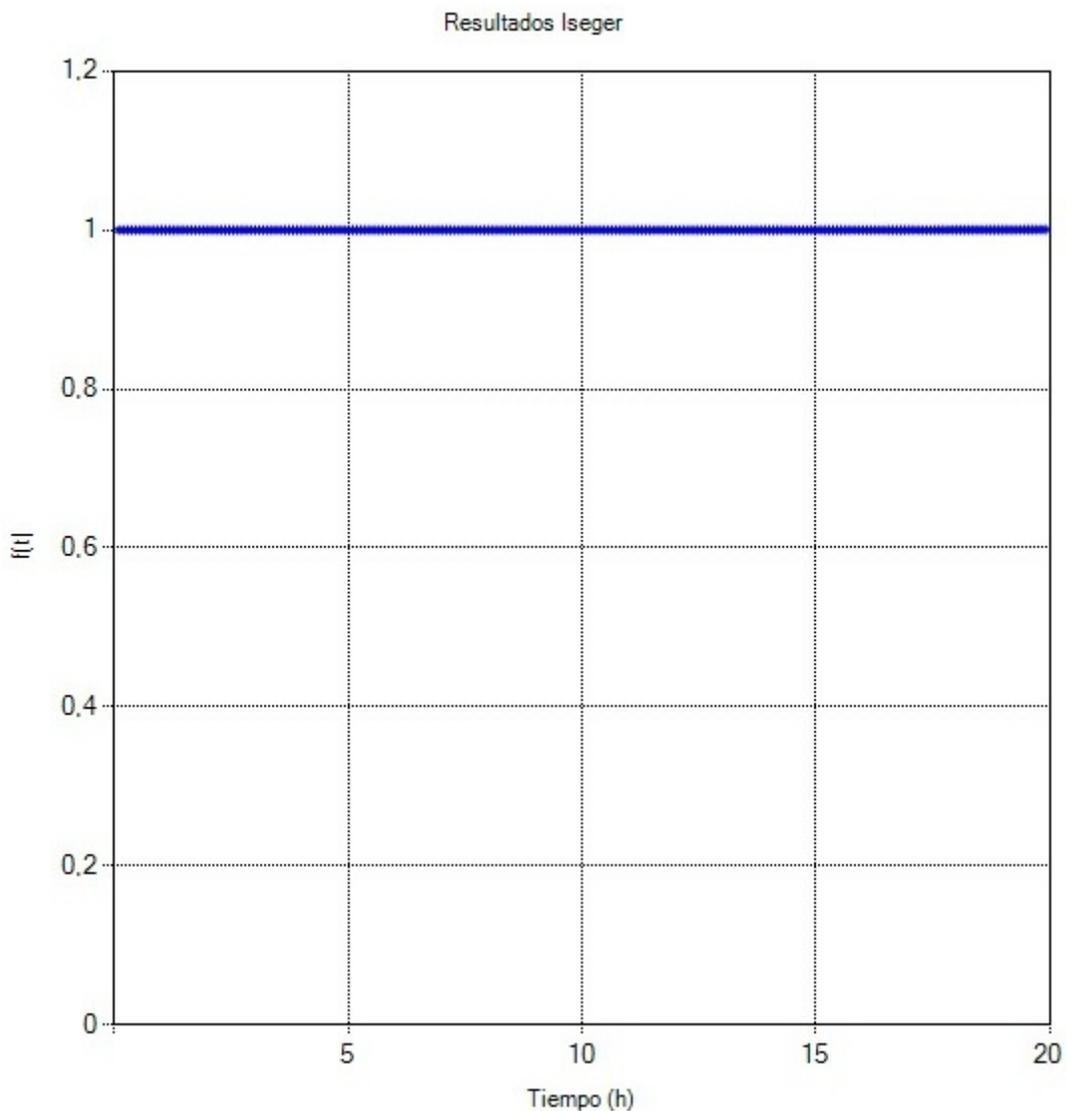


Fig. 13. Inversión de función escalón unitario por algoritmo Iseger

Aquí la prueba con una función rampa ($1/s^2$), en el programa se introduce de la manera ($1/s^2$):

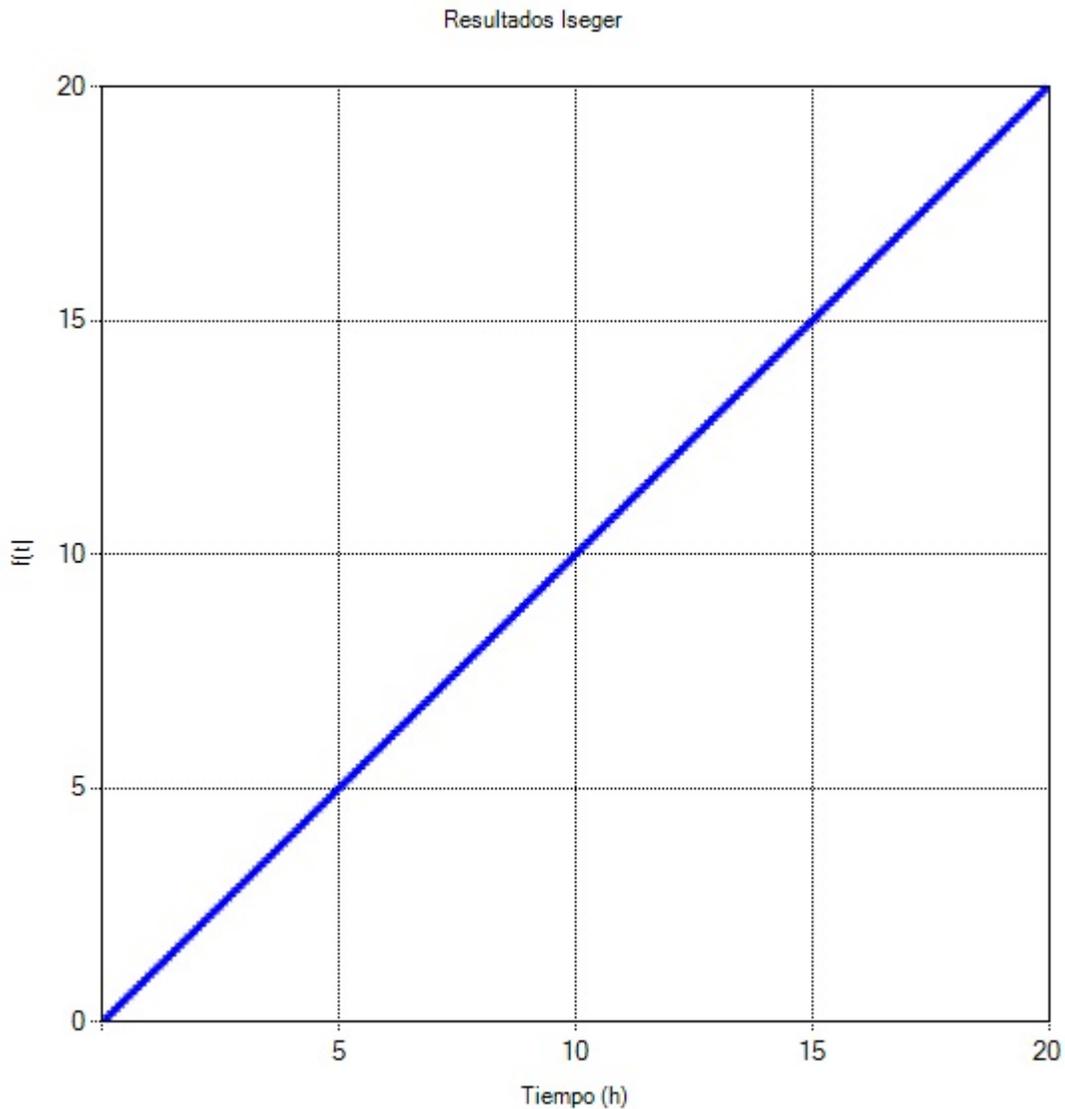


Fig. 14. Inversión de función rampa por algoritmo de Iseger

Ahora se muestran los resultados para la inversión de la función escalón unitario con retraso (e^{-10s}/s), el retraso se ubicó en 10. En el programa se introduce la función de la manera (CEXP(-10.0*v)/s):

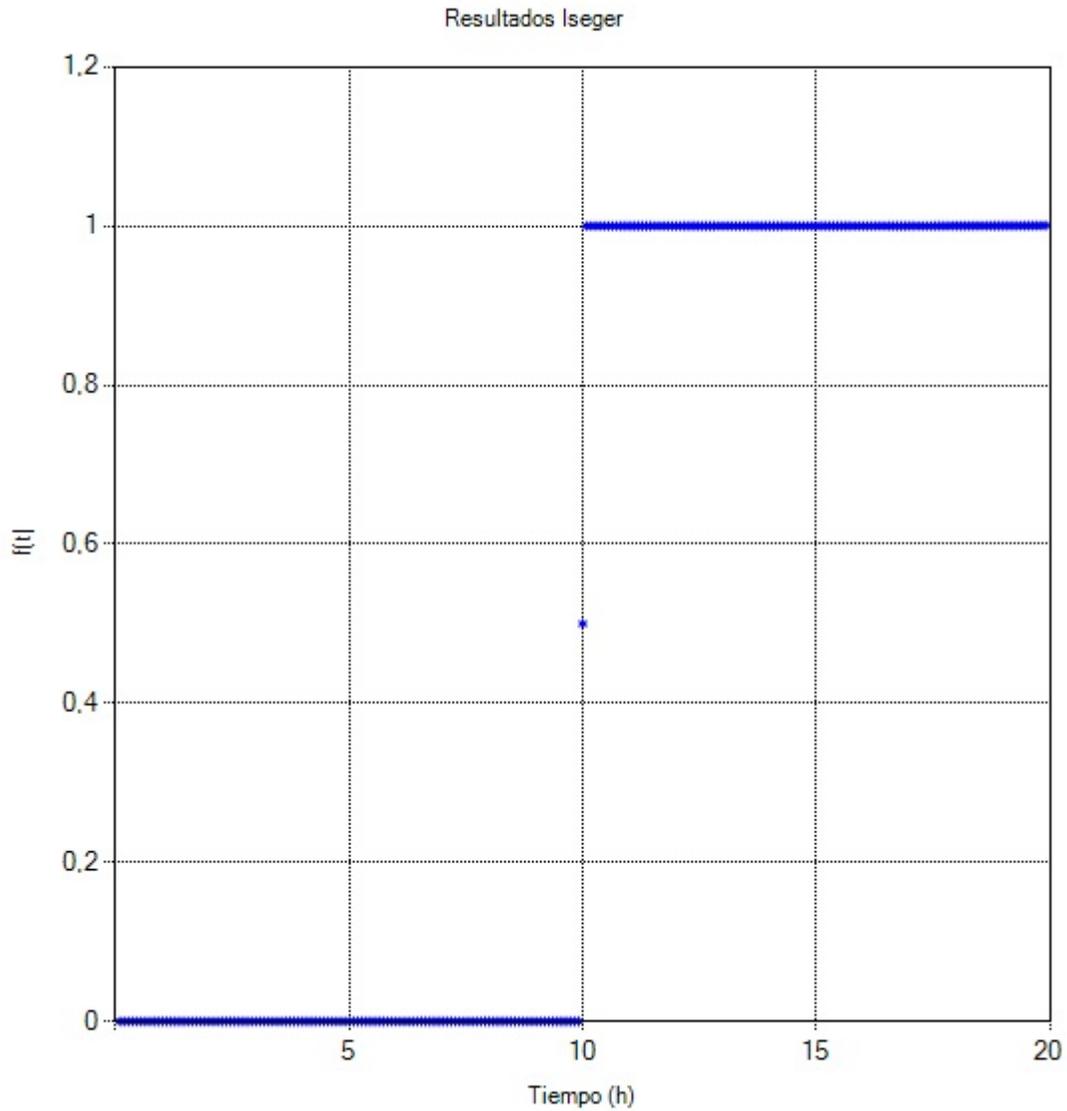


Fig. 15. Inversión de función escalón unitario con retraso por algoritmo de Iseger

La siguiente imagen corresponde a la inversión de la función escalón múltiple, la ecuación utilizada es $(1/s - e^{-5s}/s + e^{-10s}/s - e^{-15s}/s)$ y se introduce en el programa de la manera $(1.0/s-CEXP(-5.0*v)/s)+(CEXP(-10.0*v)/s-CEXP(-15.0*v)/s)$:

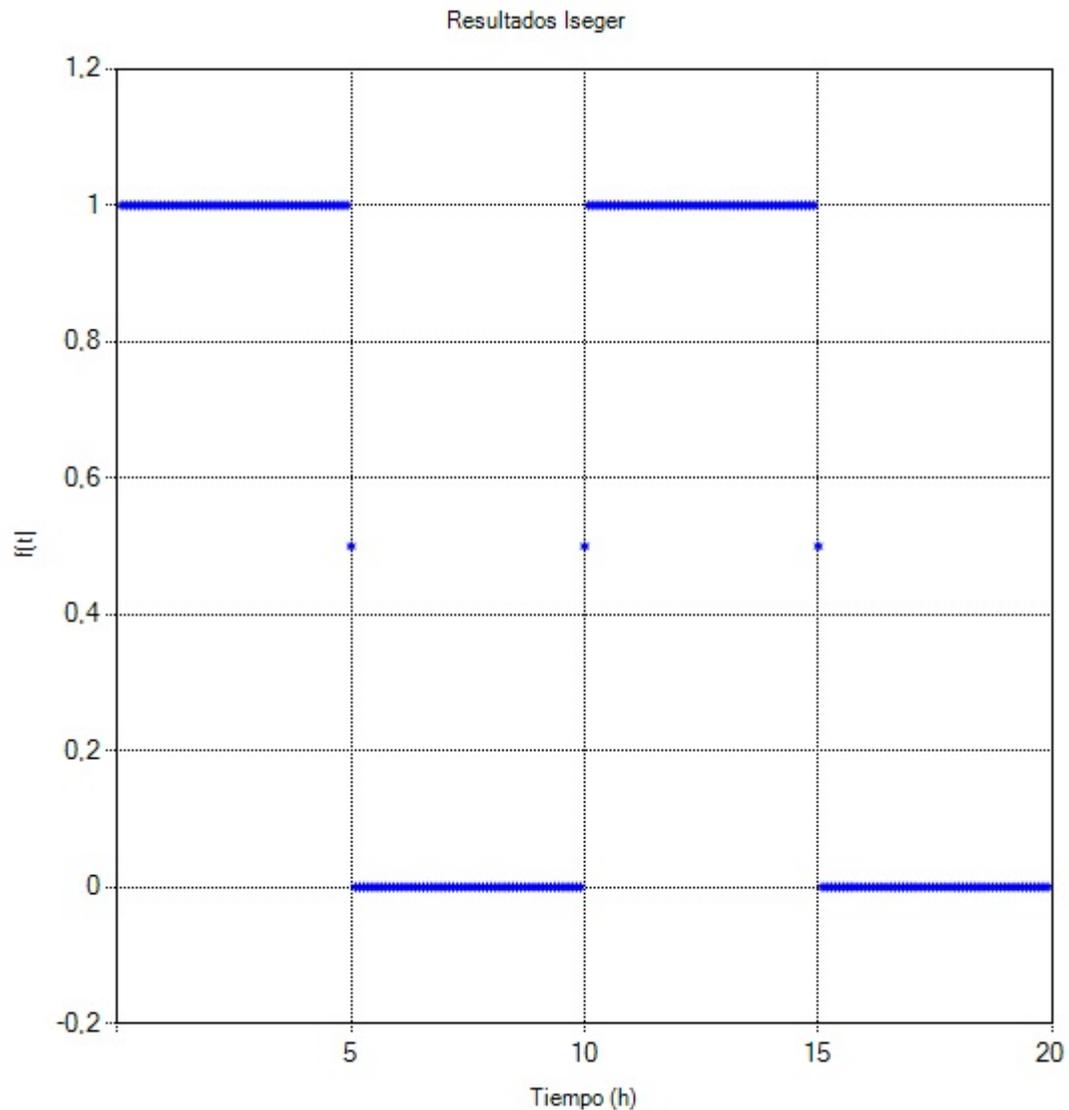


Fig. 16. Inversión de la función escalón múltiple por algoritmo de Iseger

Es evidente el éxito del algoritmo de Iseger en la inversión de funciones que son discontinuas, la naturaleza de la función no se pierde en la inversión y conserva las características de manera fiel. De aquí se puede concluir que el algoritmo de Iseger si maneja con éxito funciones discontinuas.

Una última función utilizada fue la función seno y la ecuación de la función que va a ser utilizada en el dominio Laplace es $(1/(s^2+1))$. En el programa se introduce de la manera $(1/((s^2)+1))$:

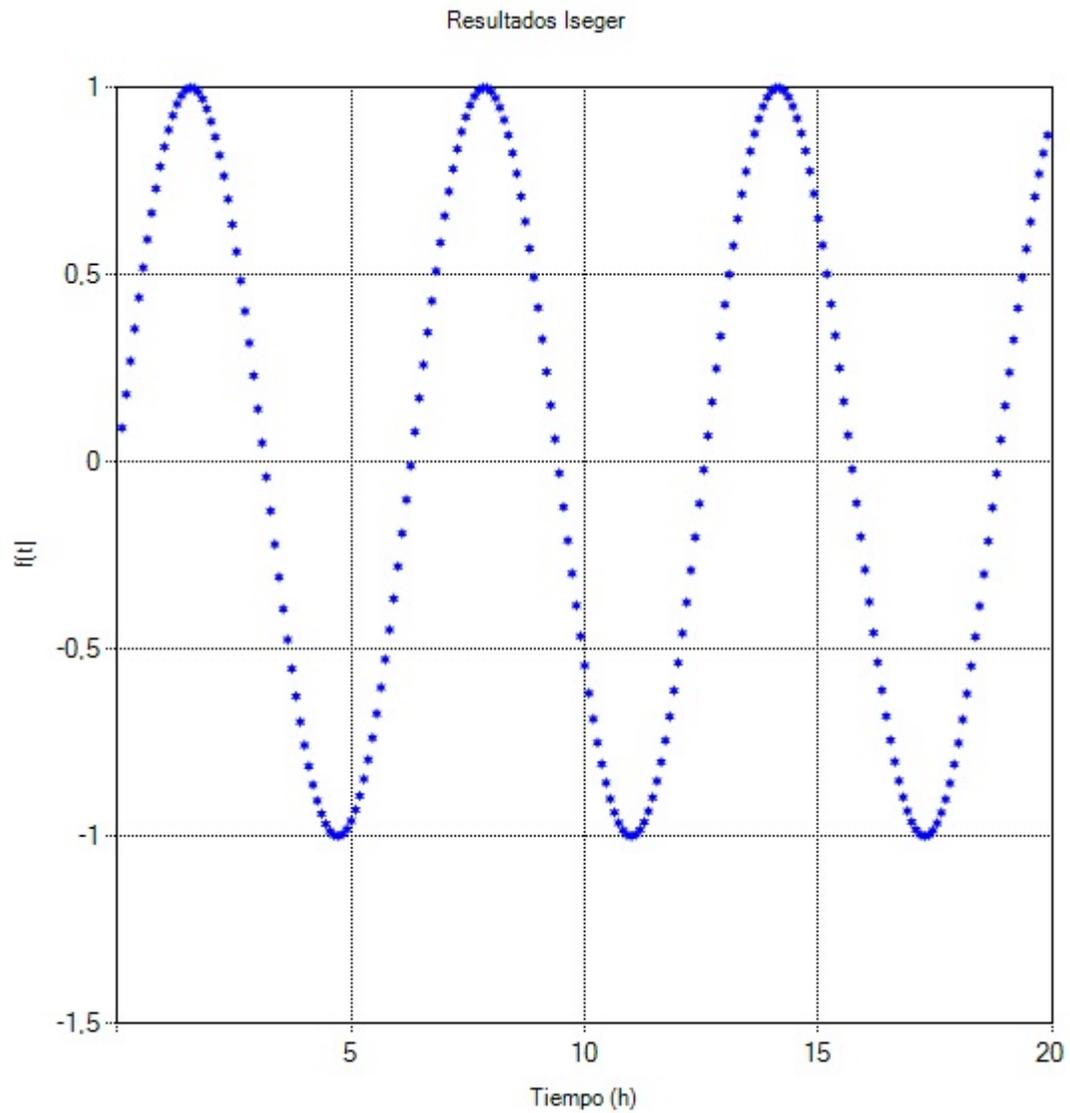


Fig. 17. Inversión de la función seno por algoritmo de Iseger

Así mismo el programa es capaz de invertir cualquier función del dominio Laplace al dominio del tiempo, el funcionamiento del algoritmo de Iseger fue satisfactorio y cumple las expectativas.

A continuación se evalúan los resultados para los dos modelos de yacimientos, el homogéneo infinito y el naturalmente fracturado

4.1. ANÁLISIS DEL YACIMIENTO HOMOGÉNEO INFINITO.

Aquí se muestra los resultados gráficos en escala logarítmica para la inversión de este modelo de pozo:

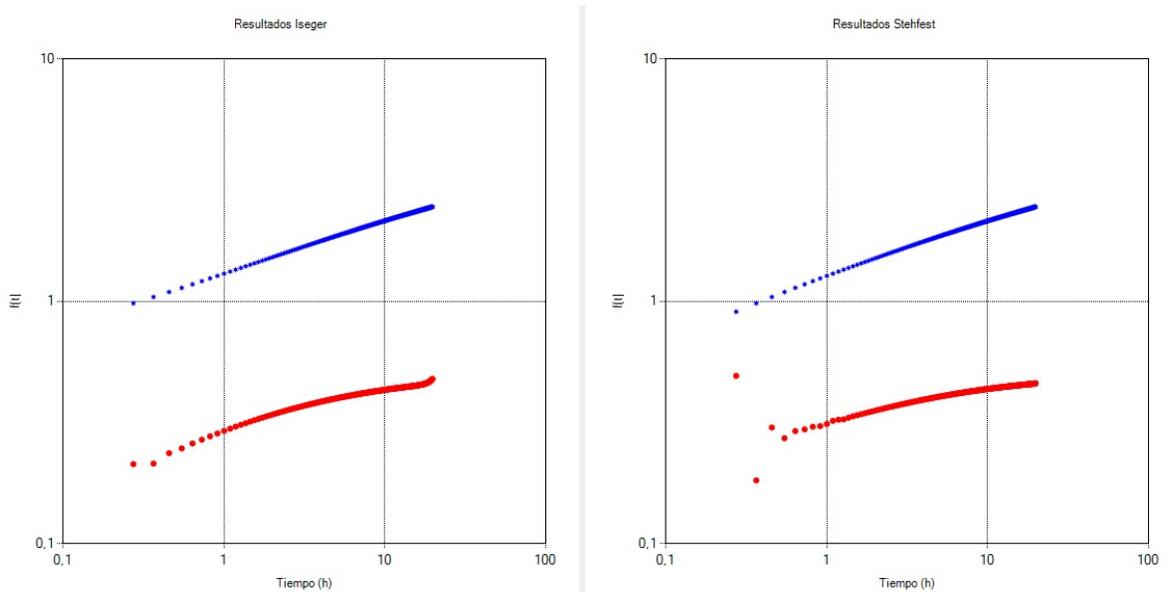


Fig. 18. Resultados inversión del modelo homogéneo infinito

Aquí se ve que 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 y al final; en Stehfest los puntos son más dispersos al inicio mientras que en Iseger los puntos están más cercanos; la derivada de presión de Stehfest parece estabilizar más rápido mientras que Iseger parece tener una ligera inclinación más pronunciada y al final tiene un incremento leve.

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.

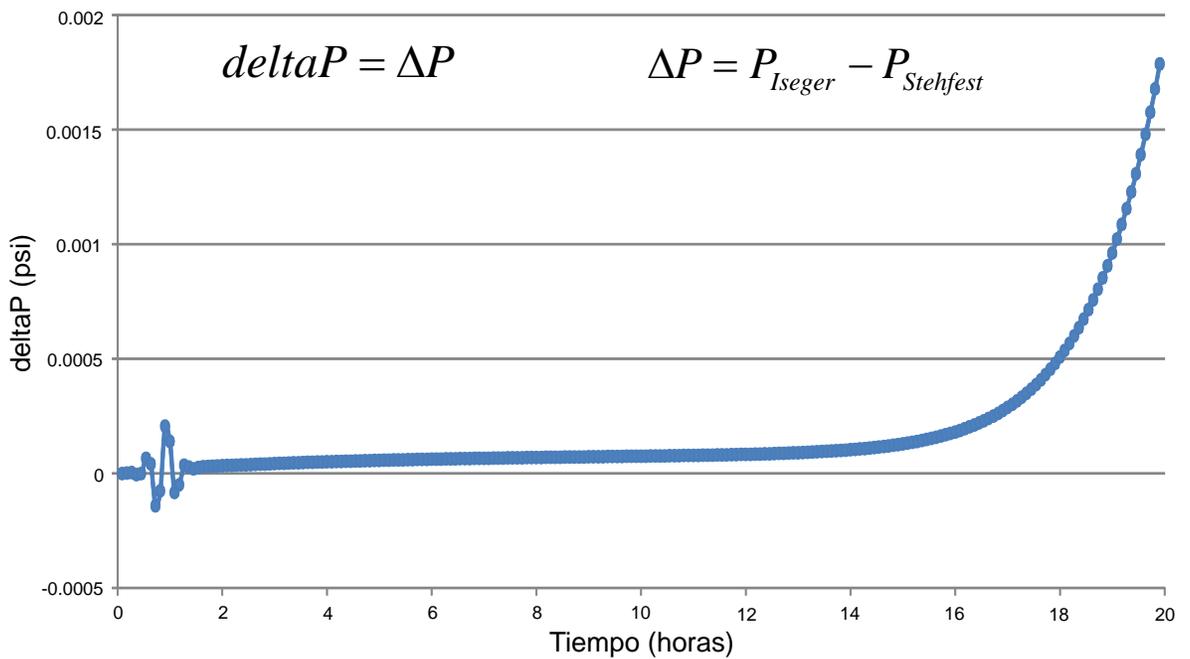


Fig. 19. Gráfica de ΔP contra el tiempo para el yacimiento homogéneo infinito

A continuación se presenta una gráfica del error expresado en porcentaje:

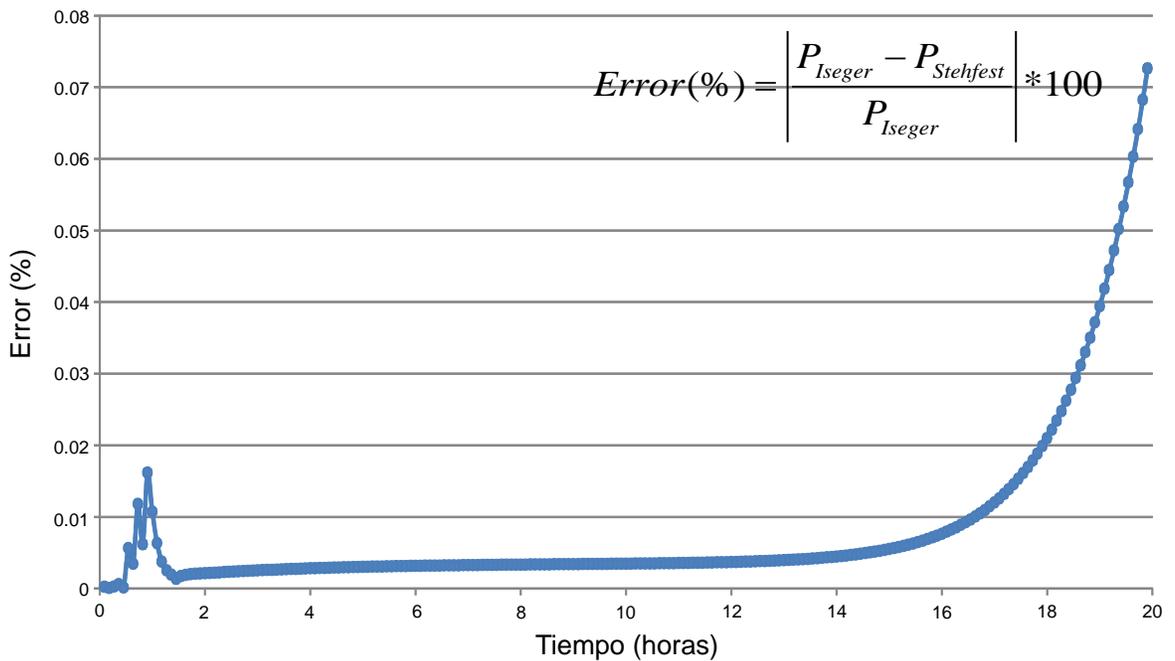


Fig. 20. Gráfica del error (%) contra el tiempo para el yacimiento homogéneo infinito

Tal y como muestran las 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 ínfima 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.

4.2. 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 Fig. 36 las gráficas de los resultados obtenidos para este yacimiento:

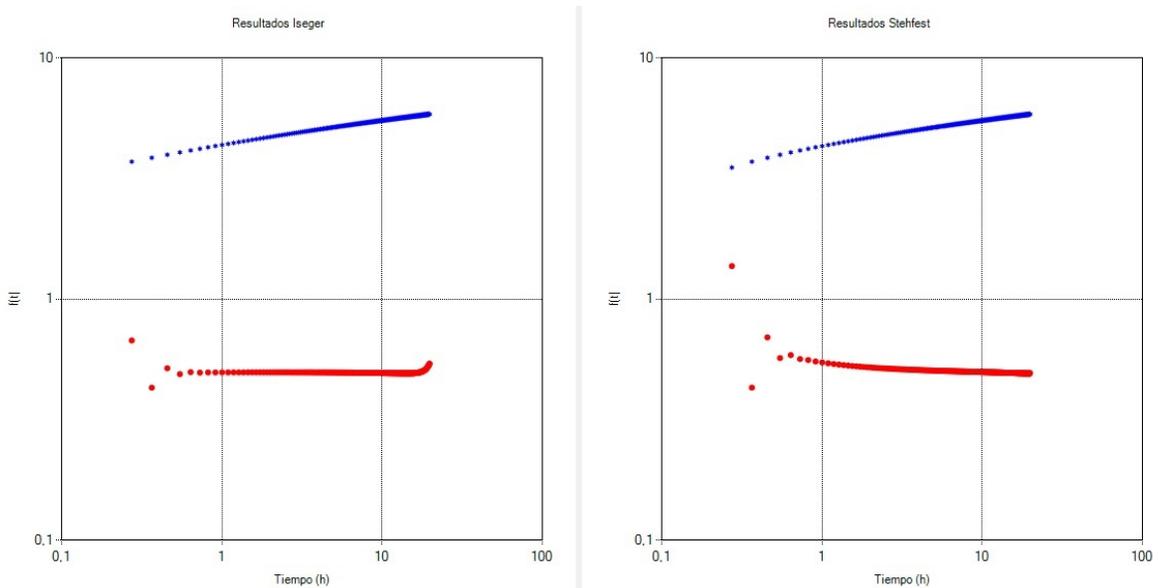


Fig. 21. Resultados inversión del modelo naturalmente fracturado

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 cuanto a la derivada se nota la estabilidad de la inversión por Iseger en el principio siendo los puntos bastante continuos, por otro lado, en la inversión por Stehfest hay una notable dispersión en los primeros puntos, todo esto se presenta antes de la hora 1. 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.

Al final de la gráfica de la derivada (roja), se notan diferencias en ambos algoritmos. Mientras en Iseger hay una inclinación hacia arriba al final de la gráfica, que indica un aumento mayor en la presión, en Stehfest parece quedar el rastro de una leve pendiente negativa. Lo de Iseger puede deberse a que la presión está en aumento de manera constante y tal vez al final el cambio es mayor.

Al igual que en el modelo de yacimiento anterior, 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.

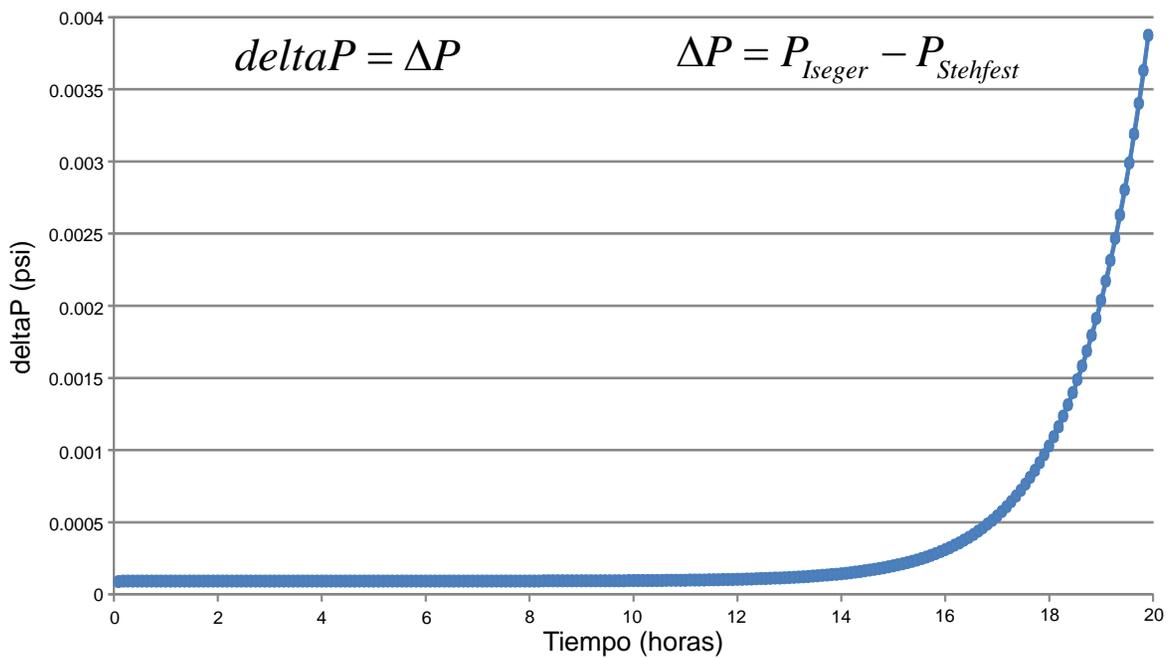


Fig. 22. Gráfica de ΔP contra el tiempo para el yacimiento naturalmente fracturado

Ahora la gráfica del error:

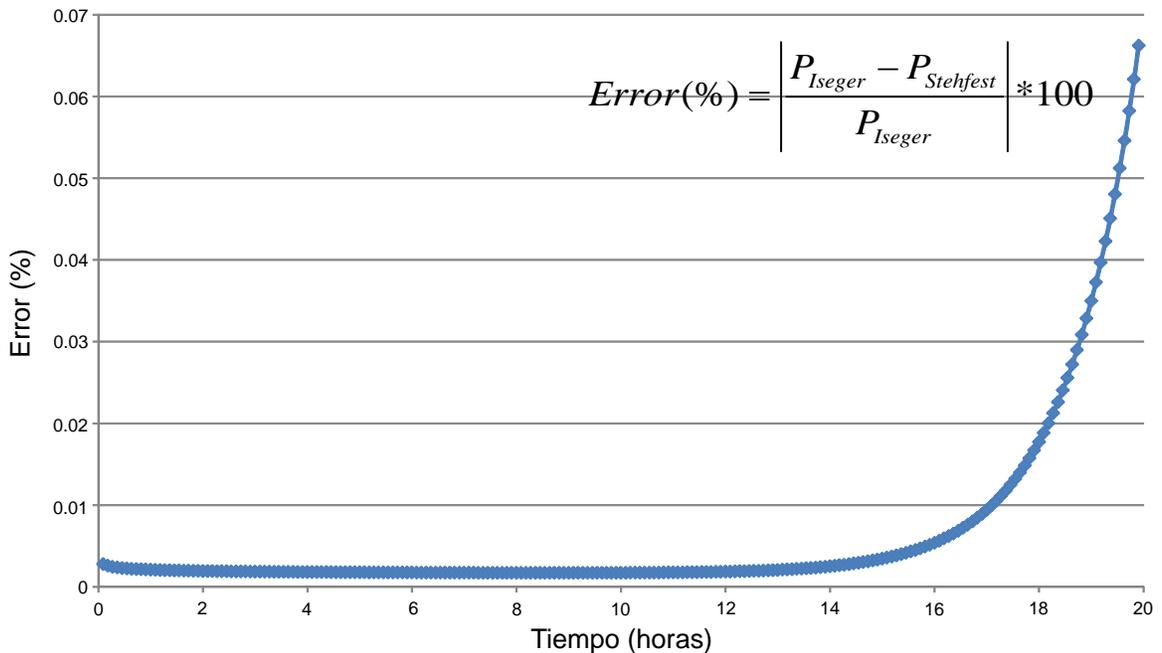


Fig. 23. Gráfica del error (%) contra el tiempo para el yacimiento naturalmente fracturado

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.

5. 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 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 , para el cual se recomienda que se cumpla la condición $(1/\Delta) \in \mathcal{N}$. Un error de este algoritmo es que al aumentar el número de puntos de muestreo (y por consiguiente el tiempo) para aumentar precisión, 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 $n_{rp} = 3$.

6. RECOMENDACIONES

Se recomienda para futuros proyectos que incluyan el actual, que se incluyan más opciones para modificar parámetros sin necesidad de acceder al código, como por ejemplo el tiempo, el número de puntos de inversión que se quieran usar y que a la vez el programa le indique al usuario si los parámetros escogidos son los adecuados para llevar a cabo de manera exitosa la inversión y si no es así, que el programa pueda sugerir al usuario unos parámetros que se acomoden de mejor manera a cada caso.

En este proyecto se realizaron las gráficas con las herramientas que cuenta .NET Framework 3.5, pero se podría utilizar algún software complementario que sea más completo como es el caso del Component One; con este se pueden realizar las gráficas y las opciones de personalización son más completas, lo que permite una mejor apreciación de los parámetros y los puntos evaluados para las funciones.

Para el manejo de las variables complejas, se utilizó un software complementario llamado Extreme Optimization Numerical Libraries for .NET 3.6, el cual es de licencia de prueba gratuita por 60 días y es renovable, pero esto se puede evitar utilizando el Visual Studio 2010 que usa el .NET Framework 4.0, el cual cuenta con las librerías necesarias para el manejo de números complejos así como para poder realizar las operaciones pertinentes en este plano.

Se espera versiones posteriores al software que contenga más modelos de yacimientos y otras funciones específicas que el anexo de estos otros modelos implique, lo cual contribuiría a la generación de un análisis más amplio para los diversos yacimientos petroleros existentes. También se pueden aumentar la cantidad de variables que pueda reconocer el software mediante el analizador de ecuaciones incluido. Otra cosa muy importante sería poder utilizar este analizador de ecuaciones de manera adecuada con el algoritmo de Stehfest para poder comparar cualquier función sin necesidad de ser programada como un modelo de yacimiento predeterminado.

El uso de software como el Visual Studio, que facilita la creación de aplicaciones para este tipo de proyectos, debe ser apoyado ya que es muy importante para fomentar la investigación e implementación de proyectos. Sus múltiples herramientas y entornos permiten desarrollar aplicaciones muy útiles para el ámbito educativo, laboral y con fines investigativos, así como proyectos a mayor escala que pudieran servir para comercializar y/o concursar en iniciativas de otras índoles.

NOMENCLATURA

B	Factor de volumen de formación del aceite, rb/STB
C	Factor de almacenamiento, bbl/psi
C_D	Factor de almacenamiento adimensional
C_f	Confer, comparte, consulte
c_t	Compresibilidad total, 1/psi
h	Espesor de formación, ft
M	Puntos de inversión
P	Presión, psi
P'	Derivada de presión, psi/hr
P_D	Presión adimensional
P_D'	Derivada de presión adimensional
P_i	Presión inicial, psi
P_{wf}	Presión de fondo fluyente (pruebas de declinación de presión), psi
P_{wD}	Solución Laplace para la presión de la cara del pozo
P_{ws}	Presión de fondo estática (pruebas de restauración de presión), psi
q	Caudal de petróleo, BPD
r	Radio, ft (distancia entre el pozo activo y el de observación)
r_D	Radio adimensional
r_{eD}	Radio adimensional de los límites del yacimiento, $r_D = r_{eD} = r_e/r_w$
r_w	Radio del pozo, ft
S	Factor de daño (skin)
s	parámetro Laplace
t	Tiempo, hrs, tiempo de prueba
t_D	Tiempo adimensional
$t_D^*P_D'$	Derivada de presión semilog adimensional
$t^*\Delta P$	Derivada de presión semilog, psi
w, w_f	Ancho de fractura, in
z	Variable Laplaciana

SIMBOLOS GRIEGOS

ω	Coeficiente adimensional de almacenaje de un yacimiento naturalmente fracturado, $(\phi c_t)_f / [(\phi c_t)_m + (\phi c_t)_f]$
Δ	Cambio, caída
ΔP	Diferencia de presión, psi
$\Delta P'$	Cambio de la rata de presión con el tiempo (derivada de presión), psi
Δt	Tiempo de cierre total, hrs, intervalo de tiempo
ϕ	Porosidad, fracción
λ	Parámetro de flujo interporoso
ρ	Densidad, lbm/ft ³
μ	Viscosidad del aceite, cp
γ	Constante exponencial de Euler, 1.871

η	Factor de difusividad hidráulica
\mathcal{L}	Transformada de Laplace
\mathcal{L}^{-1}	Transformada inversa de Laplace

BIBLIOGRAFÍA

LIBROS

1. Abate, J., Choudhury, G.L., & Whitt, W. (1996). On the Laguerre method for numerically inverting Laplace transforms. *INFORMS Journal on Computing* 8: 413–427.
2. Abate, J., Choudhury, G.L., & Whitt, W. (1998). Numerical inversion of multidimensional Laplace transforms by the Laguerre method. *Performance Evaluation* 31: 216–243.
3. Abate, J. and Valkó, P. P. 2004. Multi-Precision Laplace Transform Inversion. *Computers Math. Applic.* 48: 629-636.
4. Abate, J. & Whitt, W. (1992). The Fourier-series method for inverting transforms of probability distributions. *Queueing Systems* 10: 5–88.
5. Abate, J. & Whitt, W. (1995). Numerical inversion of Laplace transforms of probability distributions. *ORSA Journal on Computing* 7: 36– 43.
6. 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.
7. Choudhury, G.L., Lucantoni, D.L., & Whitt, W. (1994). Multi-dimensional transform inversion with applications to the transient M/G/1 queue. *Annals of Applied Probability* 4: 719–740.
8. Churchill, R. V. (1972). *Operational Mathematics*. 3rd Ed. McGraw-Hill, Inc. New York.
9. Cohen, A. M. (2007). *Numerical Methods for Laplace Transform Inversion*. Springer Science+Business Media LLC. New York.
10. Cooley, J.W. & Tukey, J.W. (1965). An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computing* 19: 297–301.
11. Crump, K. S. 1976. Numerical Inversion of Laplace Transforms Using a Fourier Series Approximation. *J. ACM* 233: 89-96.
12. Dubner, H. & Abate, J. (1968). Numerical inversion of Laplace transforms by relating them to the finite Fourier cosine transform. *Journal of the ACM* 15: 115–123.

13. Escobar Macualo, Freddy (2009). *Análisis Moderno de Pruebas de Presión*. Editorial Universidad Surcolombiana. Segunda edición.
14. Gaver, D.P. (1966). Observing stochastic processes and approximate transform inversion. *Operations Research* 14: 444– 459.
15. Kryzhniy V.V. (2004). High-resolution exponential analysis via regularized numerical inversion of Laplace transforms. *J. Comp. Phys.*, Vol.199/2, 618- 630
16. Kryzhniy, V. V. (2006). Numerical Inversion of the Laplace Transform: Analysis via Regularized Analytic Continuation.
<http://laplacetransform.org/>.
17. Lamm, P. K. (2000). A Survey of Regularization Methods for First-Kind Volterra Equations. *Surveys on Solution Methods for Inverse Problems*. ed. D. Colton, H. W. Engl, A. Louis, J. R. McLaughlin, W. Rundell. Springer, Vienna and New York: 53-82.
18. Mallat, S. (2001). *A wavelet tour of signal processing*, 2nd ed. San Diego, CA: Academic Press.
19. Murli, A. & Rizzardi, M. (1990). Algorithm 682. Talbot's method for the Laplace inversion problem. *ACM Transactions on Mathematical Software* 16: 158–168.
20. Onur, M. and Reynolds, A. C. (1998). Numerical Laplace Transformation of Sampled Data for Well-Test Analysis. *SPERE* (June): 268-277.
21. Ozkan, E. and Raghavan, R. (1997). Some Strategies to Apply the Stehfest Algorithm for a Tabulated Set of Numbers. *SPE Journal*. 2 (Sept.): 363-372.
22. O'Kinneide, C.A. (1997). Euler summation for Fourier series and Laplace transform inversion. *Stochastic Models* 13: 315–337.
23. Piessens, R. (1971). Gaussian quadrature formulas for the numerical inversion of Laplace transform. *Journal of Engineering Mathematics* 5: 1–9.
24. Piessens, R. (1971). Some aspects of Gaussian quadrature formulae for the numerical inversion of the Laplace transform. *The Computer Journal* 14: 433–436.
25. Raghavan, Rajagopal (1993). *Well Test Analysis*. Prentice Hall Petroleum Engineering Series. New Jersey.

26. Sakurai, T. (2004). Numerical inversion of Laplace transform of functions with discontinuities. *Advances in Applied Probability* 20(2): 616– 642.
27. Spiegel, Murray R.: *Transformadas de Laplace* (1991) Mc Graw Hill / Interamericana de México, México D.F.
28. Stehfest, H. (1970). Numerical Inversion of Laplace Transforms. *Communications, ACM* 13 (1): 47–49.
29. Stoer, J. & Bulirsch, R. (1991). *Introduction to numerical analysis*, 2nd ed. Text in Applied Mathematics Vol. 12, Berlin: Springer-Verlag.
30. Szegő, G. (1975). *Orthogonal polynomials*, 4th ed. American Mathematical Society Colloquium Publication 23. Providence, RI: American Mathematical Society.
31. Talbot, A. (1979). The accurate inversion of Laplace transforms. *Journal of the Institute of Mathematics and Its Applications* 23: 97–120.
32. Valkó, P. P. and Abate, J. (2004). Comparison of Sequence Accelerators for the Gaver Method of Numerical Laplace Transforms Inversions. *International Journal for Numerical Methods in Engineering*. 60: 979-993.
33. van Everdingen, A. F. and Hurst, W. (1953). The Application of the Laplace Transformation to Flow Problems in Reservoirs, *Trans. AIME* 198: 171-176.
34. Weeks, W.T. (1966). Numerical inversion of Laplace transforms using Laguerre functions. *Journal of the ACM* 13: 419– 426.

DOCUMENTOS

35. Iseger, P. D. 2006. Numerical Transform Inversion Using Gaussian Quadrature. *Probability in the Engineering and Informational Sciences*, 20. 1-44.
36. N. Al-Ajmi, M. Ahmadi, E. Ozkan and H. Kazemi, 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.

PRINCIPALES ENLACES

<http://es.scribd.com/doc/56999152/Trabajo-de-Flujo-de-Fluidos-en-Medios-Porosos>

<http://es.wikipedia.org>

<http://www.lacomunidadpetrolera.com/showthread.php/411-Dolomitizaci%F3n.?p=881>

<http://www.azimadli.com/vibman-spanish/transiente1.htm>

singularidad, se puede calcular $f(t)$ de manera precisa invirtiendo la transformada de Laplace escalada.

$$f_w(t) = w(t)f(\Delta t),$$

con Δ lo suficientemente pequeña. El precio que se debe pagar por una Δ pequeña es que M se vuelve más grande ($M \approx t/\Delta$).

Cuando la función tiene muchos picos, y se quieren obtener resultados con precisión cercana a la de máquina, se recomienda escoger $n=48$.

Para este proyecto de tesis, en el cual se evalúan dos funciones correspondientes a dos modelos de pozo, se definió $n=16$ para que los resultados fueran precisos. [35, 36].

ANEXO B. LA TRANSFORMADA DE FOURIER

Se empieza con la introducción de la transformada de Fourier en el espacio $L^1(-\infty, \infty)$ de las funciones integrables de Lebesgue. Para una función $f \in L^1(-\infty, \infty)$ y puramente imaginaria s , la integral de Fourier

$$\hat{f}(s) := \int_{-\infty}^{\infty} e^{-st} f(t) dt$$

está bien definida. De hecho,

$$|\hat{f}(s)| \leq \int_{-\infty}^{\infty} |f(t)| dt = \|f\|_1$$

y \hat{f} es una función continua. Si el soporte de f está contenido en $[0, \infty)$, entonces se llama a \hat{f} la transformada de Laplace de f . Si $\hat{f} \in L^1(-i\infty, i\infty)$, el espacio de valores complejos de las funciones integrables de Lebesgue, cumple que $(1/2\pi i) \int_{-i\infty}^{i\infty} |\hat{f}(s)| ds < \infty$, entonces la integral inversa de Fourier está definida de manera correcta.

Teorema B.1. Si $f \in L^1(-\infty, \infty)$ y $\hat{f} \in L^1(-i\infty, i\infty)$, entonces

$$f(t) = \frac{1}{2\pi i} \int_{-i\infty}^{i\infty} e^{st} \hat{f}(s) ds$$

está bien definida y

$$f(t) \leq \frac{1}{2\pi i} \int_{-i\infty}^{i\infty} |\hat{f}(s)| ds = \|\hat{f}\|_1$$

Este teorema puede ser encontrado en cualquier texto acerca de las transformadas de Fourier; en este caso se tomó de Mallat [18]. Se deduce de inmediato que $\hat{f} \in L^1(-i\infty, i\infty)$ si y solo si f es una función continua.

ANEXO C. CUADRATURA GAUSSIANA

Definición C.1 (Cuadratura Gaussiana (Szegő [30])): Sea el producto escalar $\langle \cdot, \cdot \rangle$ dado por

$$\langle f, g \rangle = \int_I f(t) g^*(t) \mu(dt),$$

con μ como una medida positiva dada y con I como un subintervalo del eje real o un subintervalo del eje imaginario. Sea $\{q_k\}$ un conjunto ortogonal con respecto a este producto escalar. Sea el producto escalar $\langle \cdot, \cdot \rangle_n$ dado por

$$\langle f, g \rangle_n := \sum_{k=1}^n \alpha_k f(\mu_k) g^*(\mu_k),$$

con $\{\mu_k; k = 0, 1, \dots, n-1\}$ las raíces simples de q_n , y los números estrictamente positivos $\{\alpha_k; k=0, 1, \dots, n-1\}$, los apodados números Christopel, sean dados por

$$\alpha_k = \frac{1}{\sum_{j=1}^n |q_j(\mu_k)|^2} = \frac{A_n}{A_{n-1}} \frac{1}{\partial q_n(\mu_j)} \frac{1}{q_{n-1}^*(\mu_j)},$$

con A_n como el coeficiente más alto en q_n . Las raíces $\{\mu_k; k = 0, 1, \dots, n-1\} \in I$. El producto escalar $\langle \cdot, \cdot \rangle_n$ es llamado regla de cuadratura Gaussiana y es la única regla de cuadratura de orden n que satisface

$$\langle p, \mathbf{1} \rangle_n = \langle p, \mathbf{1} \rangle \text{ para todo } p \in \pi_{2n-1},$$

donde π_n es el espacio de polinomios con grado inferior o igual a n y $\mathbf{1}$ es la función constante $\mathbf{1}(t) = 1$.

ANEXO D. MANUAL DE USUARIO

Este es un software diseñado para comparar mediante una interfaz gráfica, dos algoritmos (Stehfest e Iseger) utilizados para realizar la transformada inversa de Laplace, herramienta necesaria para resolver problemas de análisis de transientes de presión en yacimientos. En esta herramienta de software se cuenta con la posibilidad de escoger entre dos modelos de pozos dados y también se puede introducir la ecuación en dominio Laplace que se desee invertir.

Al iniciar el programa el aspecto de la ventana principal es la que se muestra a continuación:

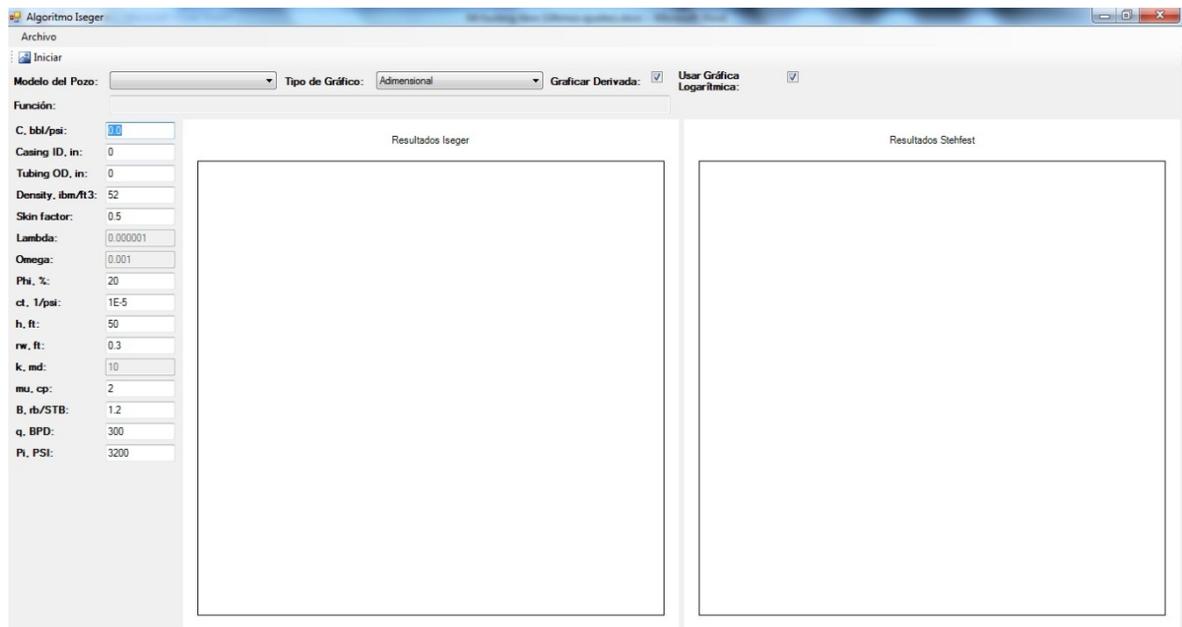


Fig. 24. Espacio de trabajo del programa

En esta ventana, que es el espacio de trabajo, se encuentra toda la interfaz del programa. Está el botón de iniciar, se encuentra la lista desplegable del Modelo del Pozo, el Tipo de Gráfico, aparecen las casillas para escoger si se grafica la derivada y si se quiere un gráfico a escala logarítmica. En el extremo izquierdo de la pantalla se encuentran las casillas con los parámetros, algunos se pueden modificar y otros no de acuerdo al modelo de pozo que se escoja. Aquí aparecen los dos campos de gráfica, el de la izquierda es para la gráfica de Iseger y el de la derecha para la de Stehfest.

Aquí una vista más detallada del campo de los gráficos.

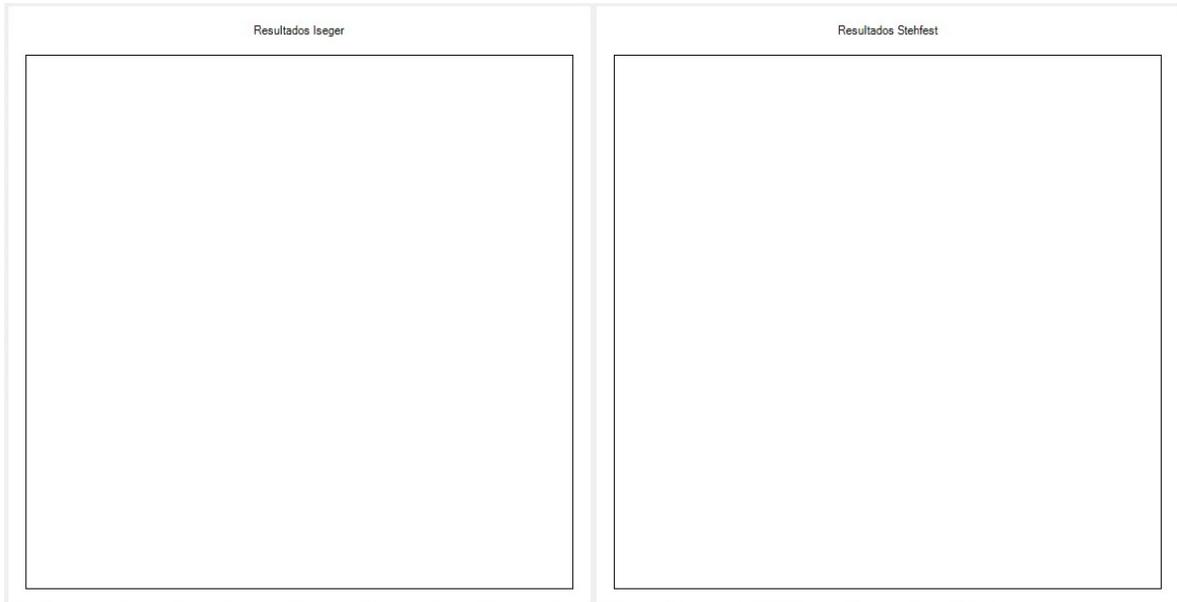


Fig. 25. Campos de gráficos de resultados

El botón iniciar es para empezar el proceso de inversión



Fig. 26. Botón Iniciar

Antes de dar click en este botón se debe escoger un modelo de pozo, de lo contrario, si no se ha escogido un modelo o no se ha introducido alguna ecuación aparecerá el siguiente mensaje:

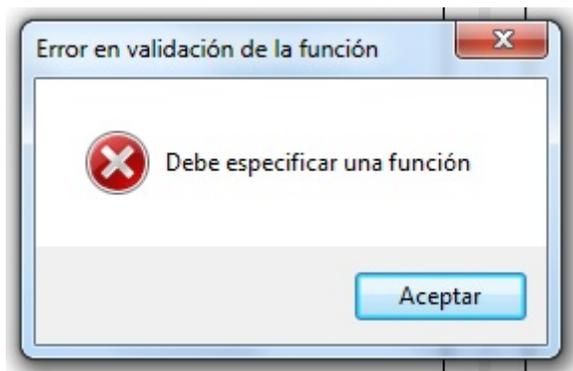


Fig. 27. Advertencia para escoger función

Por lo tanto, antes de dar click en el botón Iniciar, se debe escoger una función en el menú desplegable.

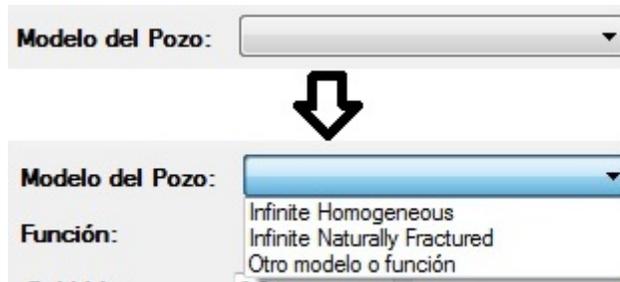


Fig. 28. Menú para escoger pozo o ecuación

Aquí aparecen tres opciones. La primera Infinite Homogeneous (Homogéneo Infinito) y la segunda Infinite Naturally Fractured (Naturalmente Fracturado) corresponden a los modelos programados, la tercera opción que dice “Otro” es para poder introducir una ecuación en dominio Laplace que se quiera invertir al dominio del tiempo. Debajo del menú desplegable para escoger el modelo de pozo se encuentra el campo donde va la función. Al escoger cualquiera de los dos modelos, este campo se llena con la ecuación correspondiente a cada modelo, si se escoge la opción “Otro”, el campo función se habilita para escribir allí la función deseada. A continuación se muestra lo que aparece para cada caso y para la opción “Otro” se introdujo por medio de teclado la función seno.

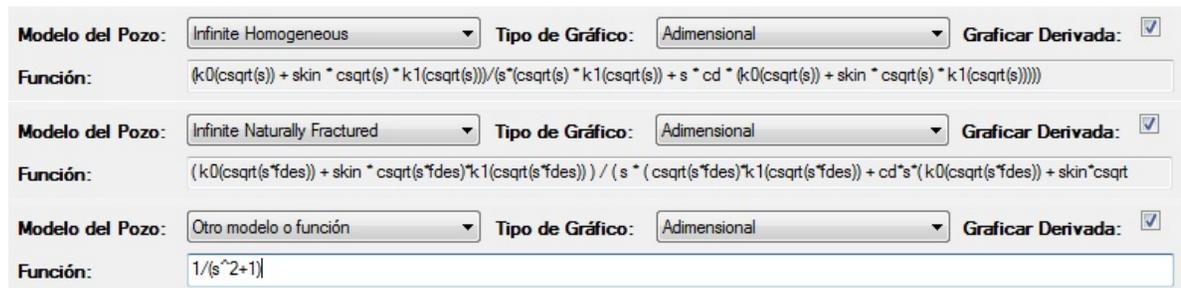


Fig. 29. Campo función según la opción escogida

Al lado de la opción modelo de pozo aparece la opción de Tipo de Gráfico, el cual es un menú desplegable donde aparece las opciones Dimensional y Adimensional.

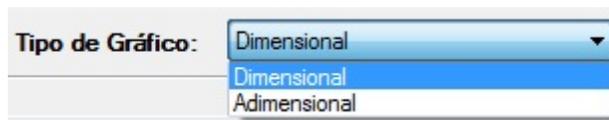


Fig. 30. Tipo de Gráfico

De acuerdo a la opción seleccionada, los datos para la gráfica serán los Dimensionales o Adimensionales.

Al lado derecho del menú desplegable para el tipo de gráfico se encuentran dos casillas para escoger si se quiere graficar la derivada y si se quiere el gráfico en escala logarítmica. Por defecto ambas están activadas.



Fig. 31. Graficar Derivada y/o Usar Gráfica Logarítmica

Los parámetros ajustables al extremo izquierdo de la pantalla varían de acuerdo al modelo que se escoja. Por defecto el parámetro k no se puede modificar y tiene el valor de 10. Para el modelo Homogéneo Infinito se desactiva la posibilidad de modificar los parámetros Lambda y Omega, pero para el modelo naturalmente fracturado estos parámetros si se pueden modificar.

C. bbl/psi:	0.0
Casing ID, in:	0
Tubing OD, in:	0
Density, ibm/ft ³ :	52
Skin factor:	0.5
Lambda:	0.000001
Omega:	0.001
Phi, %:	20
ct, 1/psi:	1E-5
h, ft:	50
rw, ft:	0.3
k, md:	10
mu, cp:	2
B, rb/STB:	1.2
q, BPD:	300
Pi, PSI:	3200

Fig. 32. Parámetros

Después de ajustar todos los valores necesarios se procede a empezar el proceso del programa dando click en el botón “Iniciar”. El programa mostrará la barra de progreso

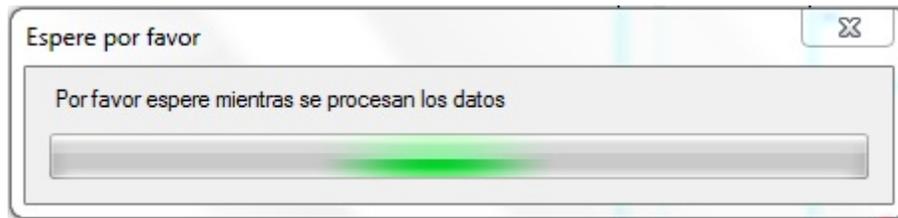


Fig. 33. Barra de progreso

Luego irán apareciendo las gráficas de los resultados así como la generación de dos archivos de texto plano con los datos de los resultados de cada algoritmo. Este es el caso en el que se escogió el modelo homogéneo infinito y Adimensional, graficando la derivada y escogiendo el tipo logarítmico.

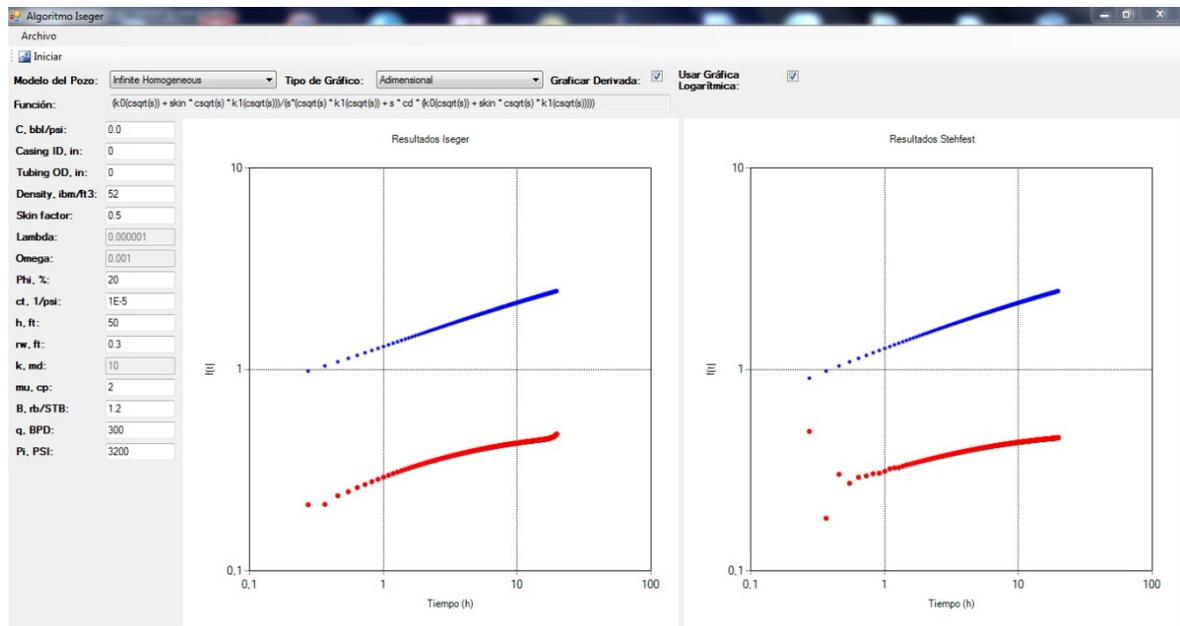


Fig. 34. Resultados en escala logarítmica para modelo de pozo homogéneo infinito

Ahora el mismo pozo, graficando la derivada pero con la gráfica lineal

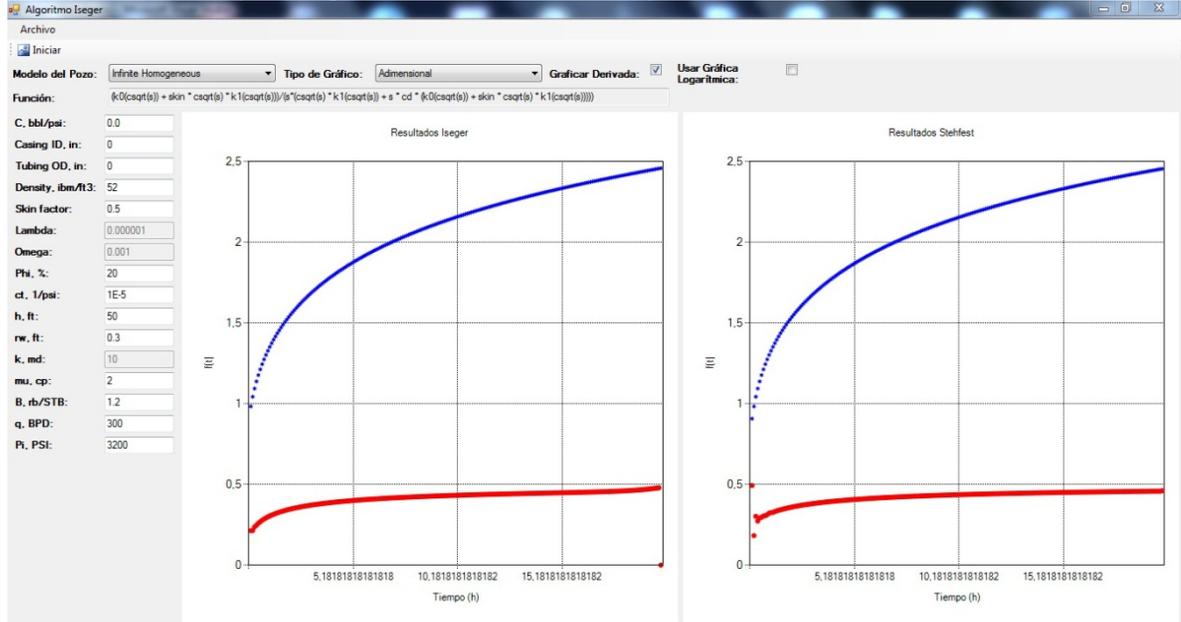


Fig. 35. Resultados en escala lineal para modelo de pozo homogéneo infinito

Es recomendable cerrar la ventana para volver a iniciar el proceso debido a que a veces se generan conflicto entre los datos. El proceso es el mismo para el otro modelo de pozo. Aquí se muestra un ejemplo de cuando se escoge la opción "Otro", se introduce la ecuación en dominio Laplace de la función seno; para esto se desmarca la opción de derivada, la de gráfica logarítmica y también se definió de manera arbitraria un tiempo de 20 horas para efectos de prueba.

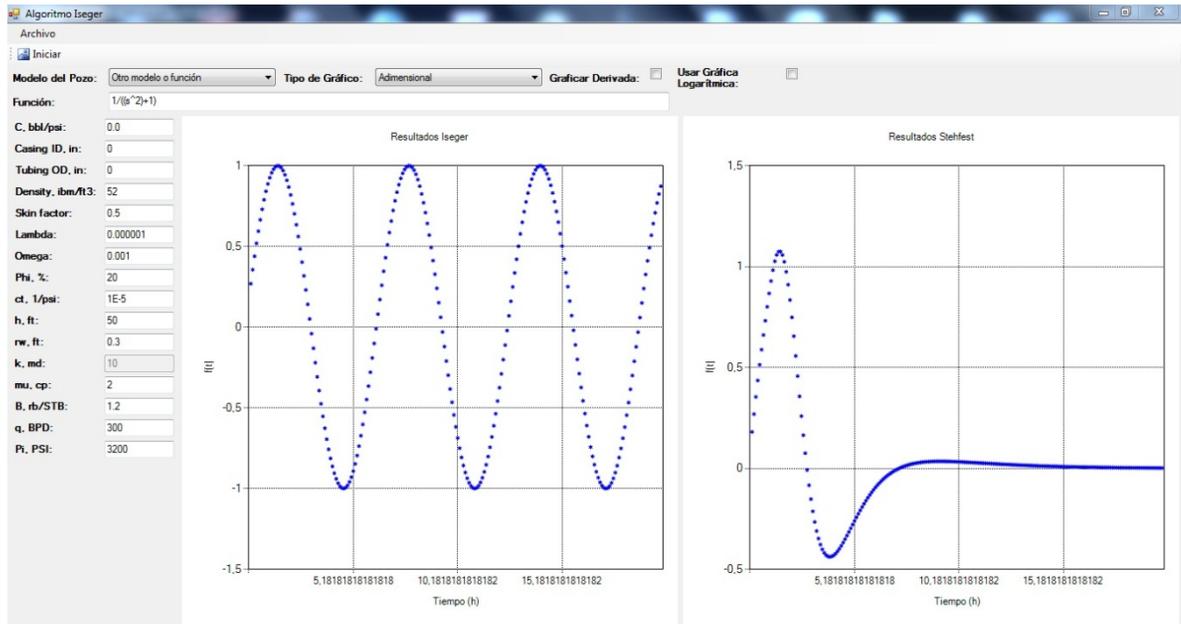


Fig. 36. Gráfica de inversión de la función seno

Otro ejemplo pero con una función tipo multi-paso.

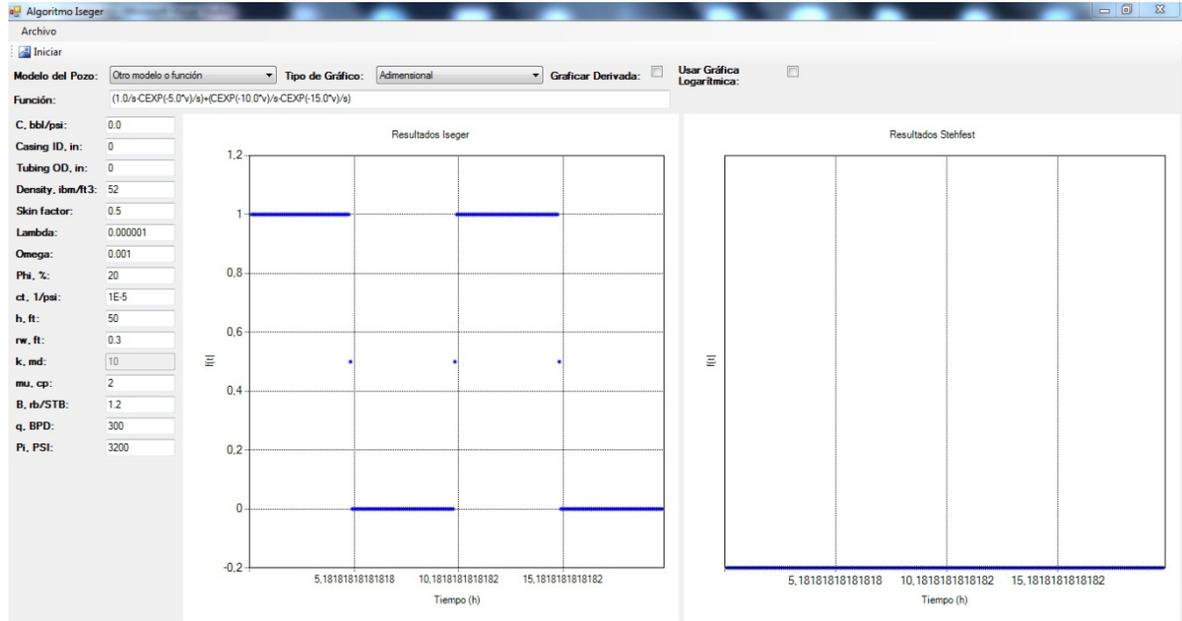


Fig. 37. Gráfica de inversión de función multipaso

Ahora una de la función rampa

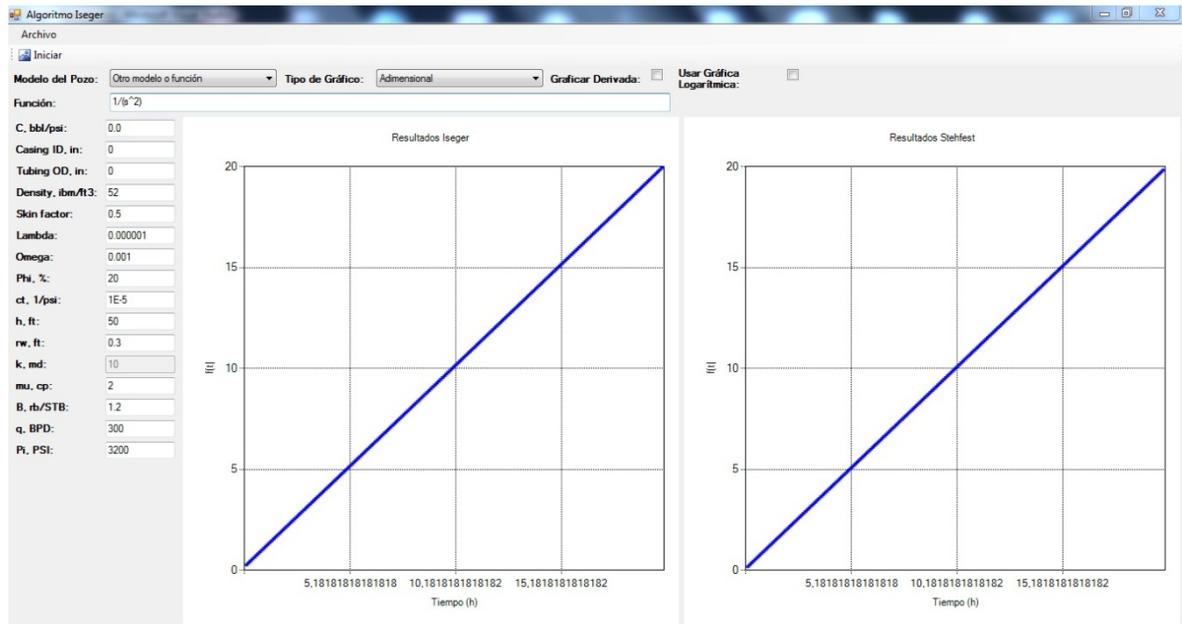


Fig. 38. Gráfica de inversión de función rampa

ANEXO E. DIAGRAMA DE FLUJO DE ALGORITMO DE ISEGER

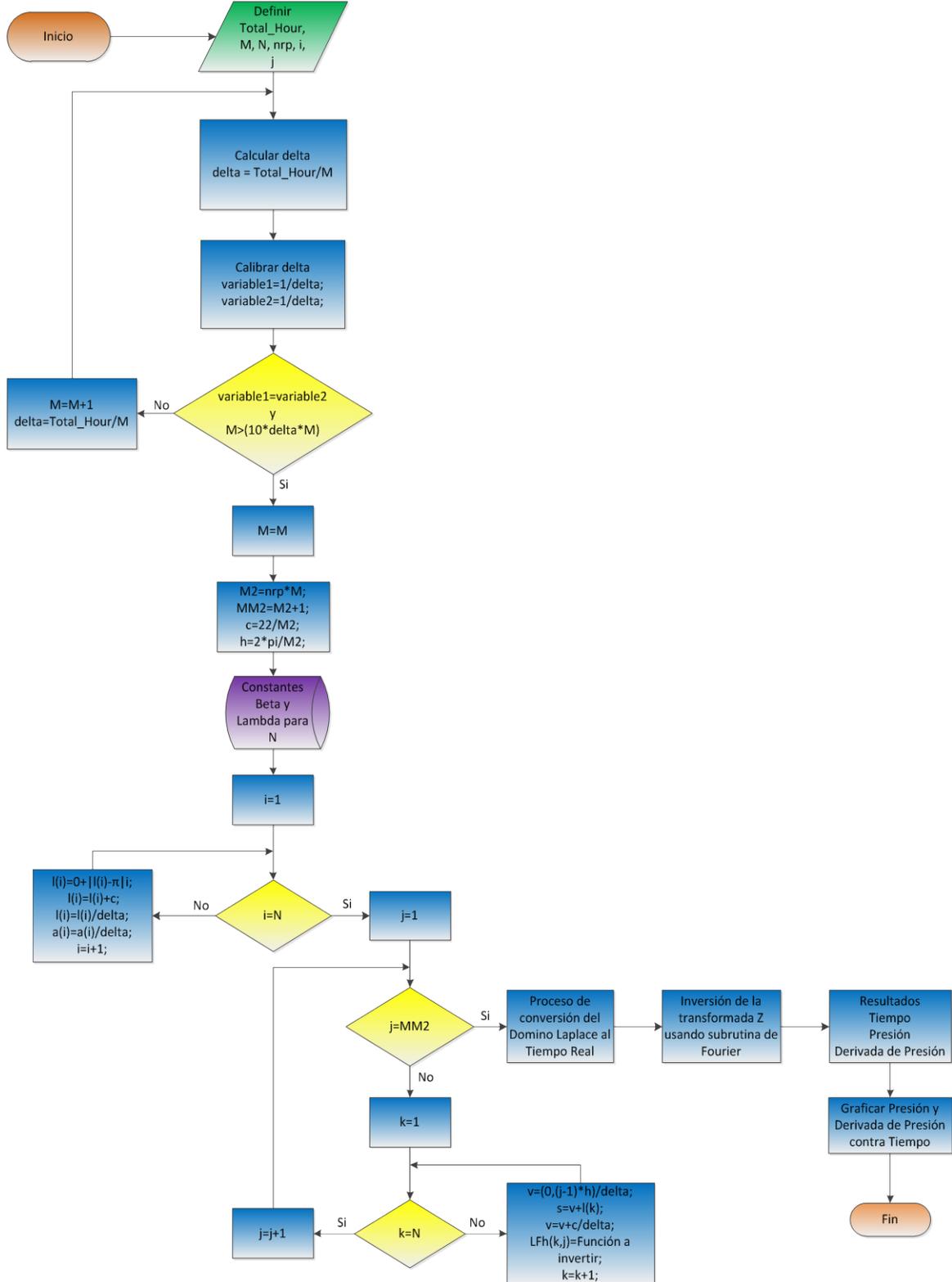


Fig. 39. Diagrama de flujo de algoritmo de Iseger

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 Leguízamo² 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.

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 via Pideciuesta, Bucaramanga, jose.cantillo@usco.edu.co

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. 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 ésto 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. Ésto 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 mas 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 $(1/\Delta) \in N$ 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 $M_2 = nrp \times M$ puntos de sobremuestreo para calcular las inversiones en M puntos de datos. El tiempo de ejecución real es $M_2 \log(M_2)$; por lo tanto, desde esta perspectiva, se desea escoger M_2 tan pequeño como sea posible. Sin embargo, para obtener un resultado numéricamente estable, se debe escoger M_2 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 esté es el factor de sobremuestreo, lo que incrementa la precisión de los resultados. Iseger (2006) recomienda usar $M_2 = 8M$, 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.

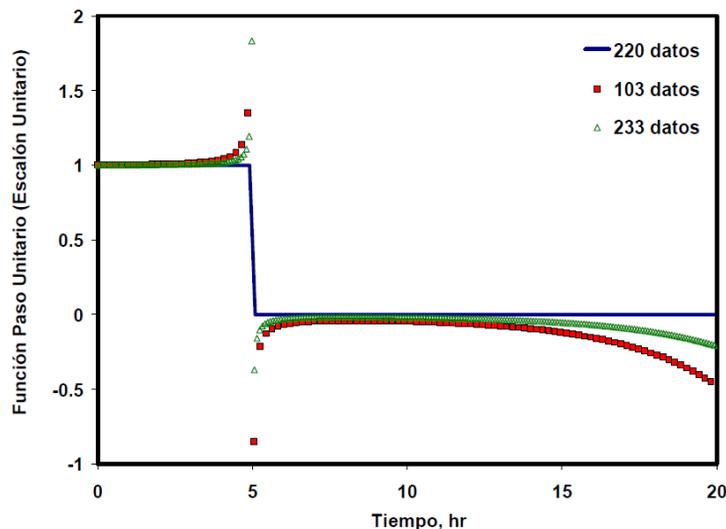


Fig. 1. Inversión numérica de una función escalón unitario por el algoritmo de Iseger. Efecto del número de puntos de inversión (Al-Ajmi *et al.*, 2008)

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.

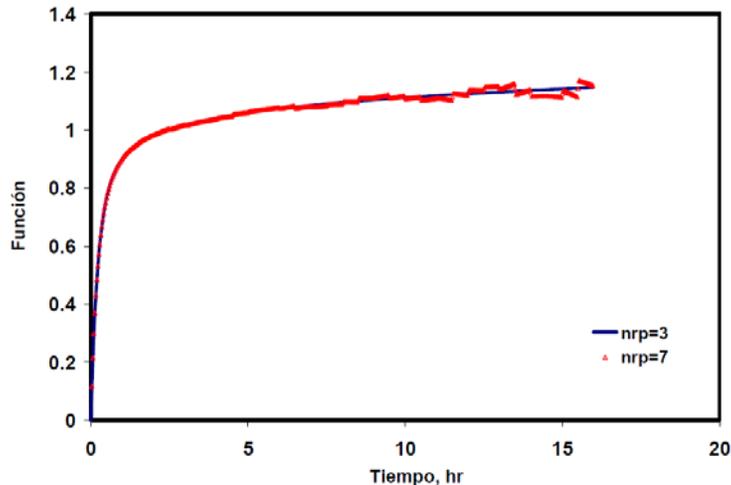


Fig. 2. Efecto de nrp en la inversión numérica de una función por el algoritmo de Iseger (Al-Ajmi *et al.*, 2008)

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 $nrp = 3$. 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).

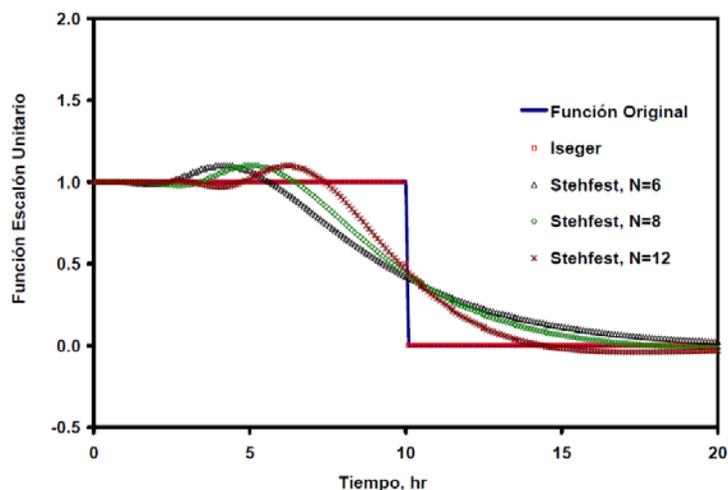


Fig. 3. Inversión numérica de una función escalón unitario; comparación de los algoritmos de Iseger y Stehfest (Al-Ajmi *et al.*, 2008)

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 $nrp = 3$ 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.

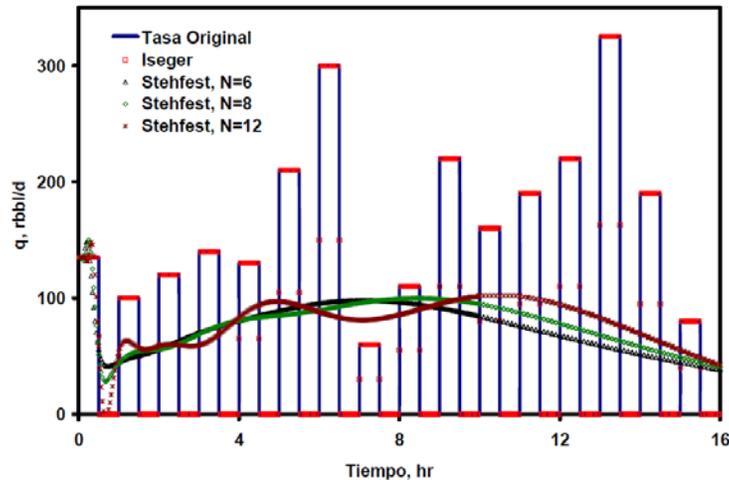


Fig. 4. Inversión numérica de una función con múltiples cambios tipo escalón; comparación de los algoritmos de Iseger y Stehfest (Al-Ajmi *et al.*, 2008)

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 Onur – 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.

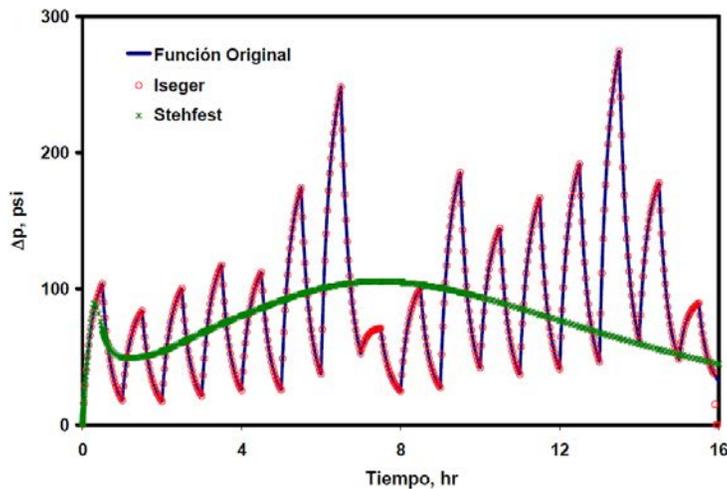


Fig. 5. Inversión numérica de una función derivable por partes; comparación de los algoritmos de Iseger y Stehfest (Al-Ajmi *et al.*, 2008)

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. Ésto 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 n_{rp} 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 $n_{rp} = 8$; luego se estableció, mediante pruebas, que usando $n_{rp} = 3$ 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.

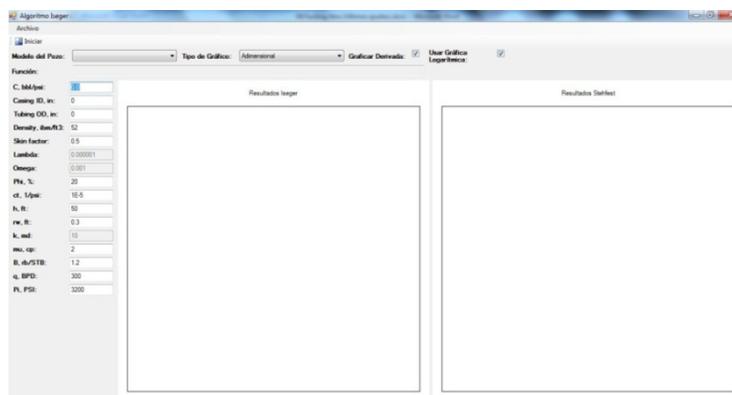


Fig. 6. Espacio de trabajo del programa

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.

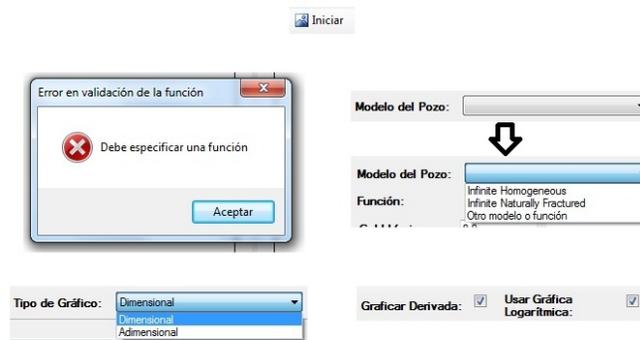


Fig. 7. Varias opciones y advertencias del software.

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.

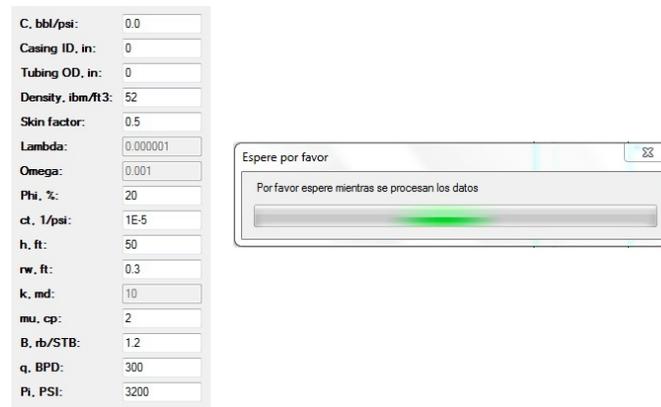


Fig. 8. Parámetros y barra de progreso

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.

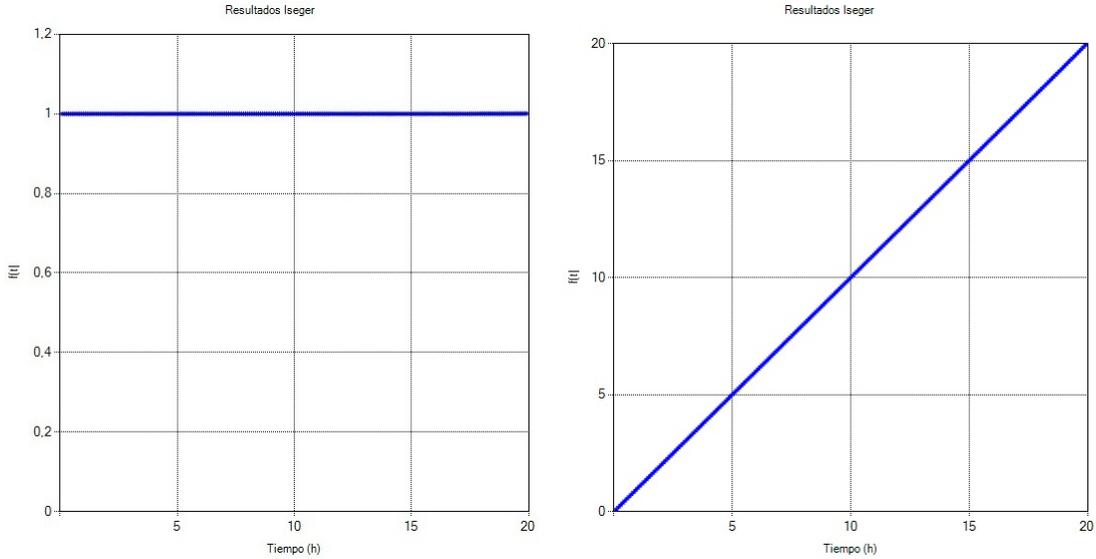


Fig. 9. Inversión de funciones escalón unitario y rampa por algoritmo Iseger

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.

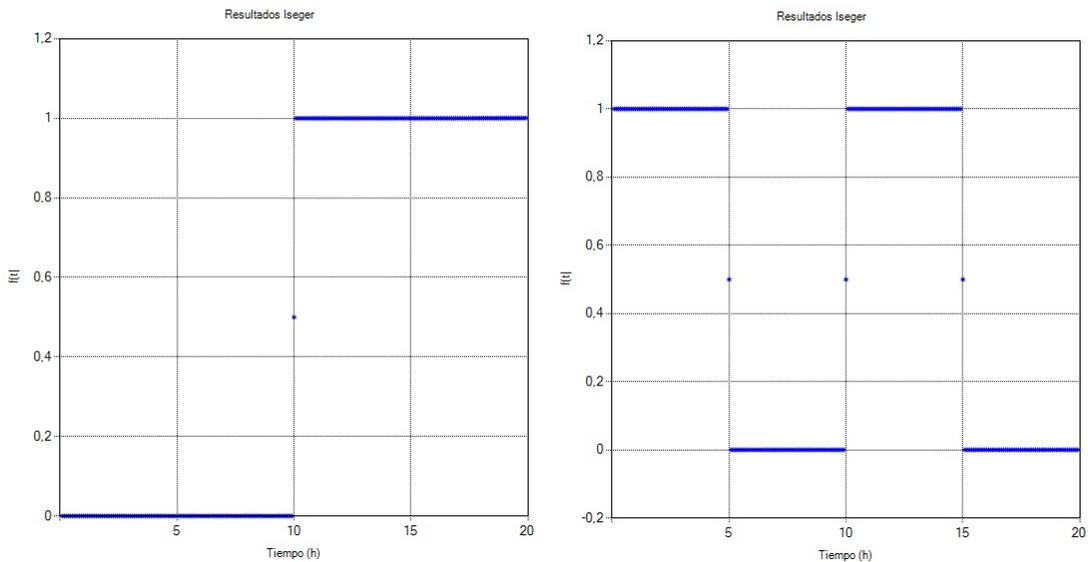


Fig. 10. Inversión de funciones escalón unitario con retraso y escalón múltiple por algoritmo de Iseger

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

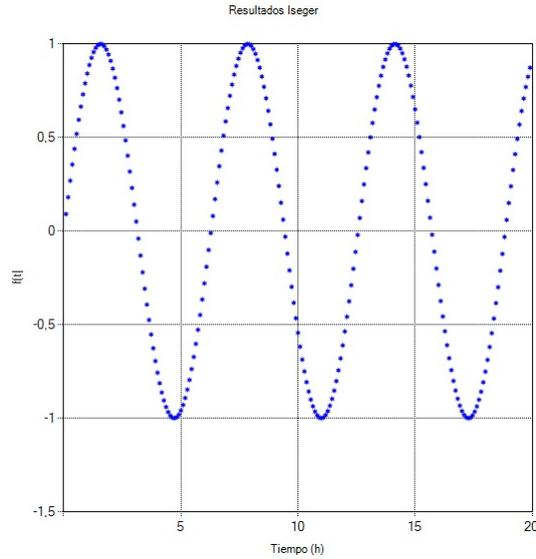


Fig. 11. Inversión de la función seno por algoritmo de Iseger

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:

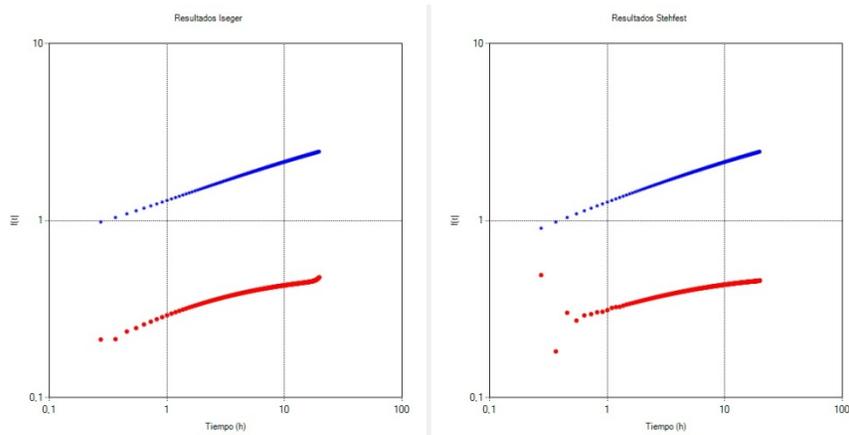


Fig. 12. Resultados inversión del modelo homogéneo infinito

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.

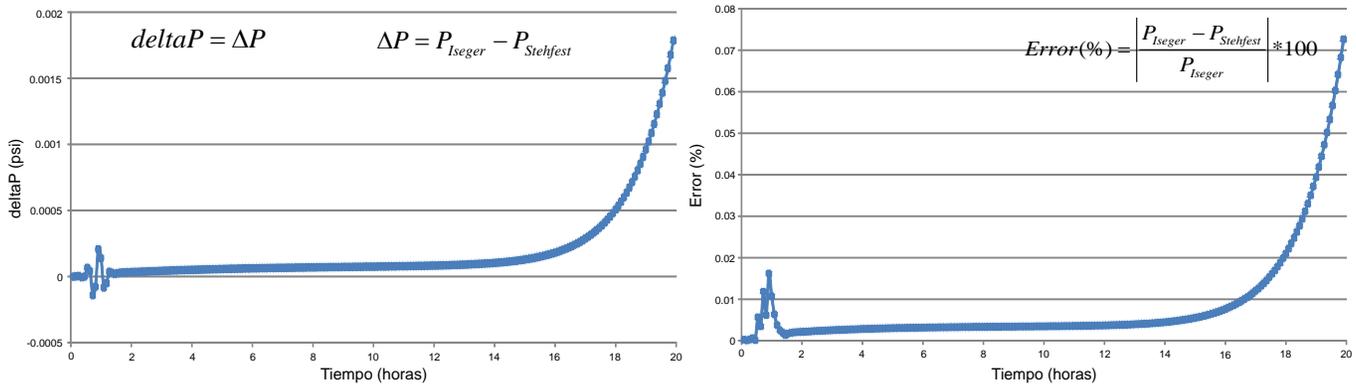


Fig. 13. Gráfica de ΔP y del error (%) contra el tiempo para el yacimiento homogéneo infinito

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 ínfima 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:

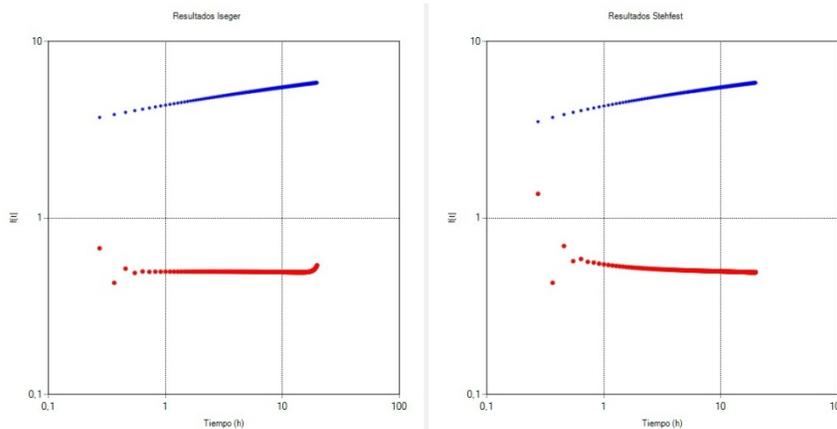


Fig. 14. Resultados inversión del modelo naturalmente fracturado

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.

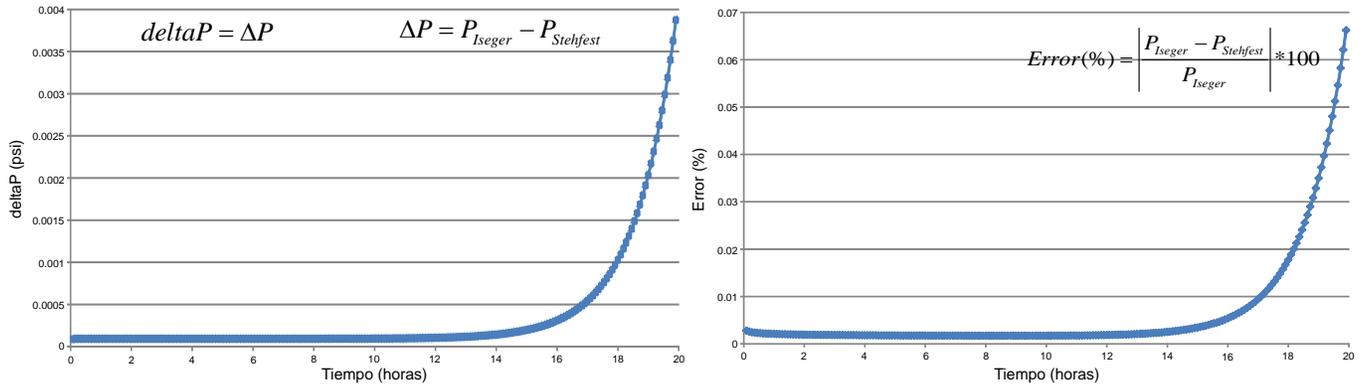


Fig. 15. Gráfica de ΔP y del error (%) contra el tiempo para el yacimiento naturalmente fracturado

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: 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.